

Major Project Report

On

DRIVER FATIGUE DETECTION SYSTEM

Submitted in partial fulfilment of the requirements for the award of the Degree of

Bachelor of Technology

In

COMPUTER SCIENCE AND ENGINEERING

By

ALLURI KAVYA SRI SAINI	16241A05I5
GALI ANUSHA	16241A05J4
SAI SOHILA MANTENA	16241A05K7
MANVITHA KRISHNA PATIBANDLA	16241A05K8

Under the Esteemed guidance of

G. Devi Priya

Assistant Professor

Department of CSE

GRIET, Hyderabad.



Department of Computer Science and Engineering

**GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND
TECHNOLOGY**

(Autonomous)

Bachupally, Kukatpally, Hyderabad- 500090

2019-2020



**GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND
TECHNOLOGY**

(Autonomous)

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the project work entitled “**Driver Fatigue Detection System**” is a bonafide work carried out by **Alluri Kavya Sri Saini (16241A05I5)**, **Gali Anusha (16241A05J4)**, **Sai Sohila Mantena (16241A05K7)**, **Manvitha Krishna Patibandla (16241A05K8)** in partial fulfilment of the requirements for the award of degree of **BACHELOR OF TECHNOLOGY** in Computer Science and Engineering from Gokaraju Rangaraju Institute Of Engineering & Technology(Autonomous), Bachupally, Kukatpally, Hyderabad, under our guidance and supervision.

INTERNAL GUIDE

G. Devi Priya

HEAD OF THE DEPARTMENT

Dr. K. Madhavi

EXTERNAL EXAMINER

DECLARATION

This is to certify that the work reported in this project entitled “**Driver Fatigue Detection System**” is a record of bonafide work done by us in the Department of Computer Science and Engineering, Gokaraju Rangaraju Institute of Engineering & Technology(Autonomous). The reports are based on the project work done entirely by us and have not been copied from any other source. The results encompassed in this project report have not been submitted to any other University or Institute for the award of any degree or diploma to the best of our knowledge and belief.

Alluri Kavya Sri Saini	16241A05I5
Gali Anusha	16241A05J4
Sai Sohila Mantena	16241A05K7
Manvitha Krishna Patibandla	16241A05K8

ACKNOWLEDGEMENT

We would like to express our sincere gratitude and indebtedness to our project guide **G. Devi Priya** for her valuable suggestions and interest throughout the course of this project.

We are also thankful to our principal **J.Praveen** and **Dr.K.Madhavi**, Professor and Head, Department of Computer Science and Engineering, GRIET Engineering College, Hyderabad for providing excellent infrastructure and a nice atmosphere for completing this project successfully as a part of our B.Tech Degree (CSE).

Finally, we would like to take this opportunity to also thank our families for their support throughout the work. We sincerely acknowledge and thank all those who directly or indirectly gave their support in completion of this work.

Alluri Kavya Sri Saini	16241A05I5
Gali Anusha	16241A05J4
Sai Sohila Mantena	16241A05K7
Manvitha Krishna Patibandla	16241A05K8

ABSTRACT

Many people working in different kinds of professions require undisturbed concentration for prolonged periods of time. It is therefore, vital for drivers to maintain an eye on the road at all times. By ensuring this, one can react to any events that occur suddenly. In the recent years, the reason for maximum number of vehicle accidents in the world is driver fatigue. Hence, it is important to develop systems that have the ability to reduce the number of car accidents that are caused due to fatigue by detecting any psychological or physical condition changes in the driver and notifying the driver about it. But the process of developing that kind of machines faces lots of difficulties that pertain to speedy and apt recognition of the symptoms for driver's fatigue.

Driver fatigue can be directly measured in a way by analyzing the current condition of the driver that is fatigue. Our project aims to develop a model for detection of fatigue. This driver fatigue detection system helps in identifying the fatigue and issuing a timely alarm to alert the driver. This real time system captures incessant pictures from a streaming video and measures what is the state of the eye based on an algorithm and notifies a warning message when necessary. Unlike the various existing approaches for identifying the fatigue of an individual, our approach proves to be completely non-intrusive by affecting the driver in any way and as a result providing us with the exact state of the driver very precisely.

In order to detect the fatigue, the PERCLOS(per closure value of eye) is considered. When the PERCLOS value outruns a given value, the driver is pointed out to be sleepy. To implement this system Haar-cascade, several OpenCv libraries are also used.

LIST OF FIGURES

Figure No.	Figure title	Page No.
Fig 1.1	Overview of other Existing systems	7
Fig 4.1	Class Diagram	28
Fig 4.2	Component Diagram	29
Fig 4.3	Deployment Diagram	30
Fig 4.4	Use Case Diagram	31
Fig 4.5	Sequence Diagram	32
Fig 4.6	Activity Diagram	33
Fig 5.1	Flowchart	34
Fig 5.2	Logic Flow	35
Fig 5.3	System Architecture	36
Fig 5.4	The Five HAAR like features	38
Fig 5.5	An example of HAAR feature based face detection	38
Fig 5.6	68 facial landmark	40
Fig 5.7	Open and close state of eye and frames of a video sequence	42
Fig 6.1	Console	48
Fig 6.2	Locating of eye and calculating the state	49
Fig 6.3	Fatigue Detection	50

CONTENTS	Page No.
Certificate	i
Declaration	ii
Acknowledgement	iii
Abstract	iv
List of Figures	v
 1. Introduction	 1
1.1 To specified topic	1
1.1.1 Definition of Fatigue	1
1.1.2 Fatigue Driving	2
1.2 Motivation	3
1.3 Existing System	4
1.4 Limitations of Existing system	5
1.5 Proposed System	6
1.6 Overview	7
 2. System Requirements	 9
2.1 Software Requirements	9
2.2 Hardware Requirements	9
 3. Technologies Used	 10
3.1 Tensor Flow	10
3.2 Machine Learning	11
3.2.1 Evolution of Machine Learning	12
3.2.2 Importance of Machine Learning	13
3.2.3 Applications of ML	13
3.3 OpenCV	15

3.3.1 Image processing using OpenCV	15
3.4 HAAR Feature based Classifier	18
3.5 Python	19
3.5.1 IDLE's interactive shell	19
3.5.2 IDLE's editor	19
3.5.3 Built-in Functions in Python	20
3.5.4 Basic Datatypes	21
3.5.5 Sequence Types	22
4. Understanding UML Diagrams	26
4.1 Introduction to UML	26
4.1.1 Structural Diagrams	27
4.1.2 Behavioral Diagrams	27
4.2 UML Diagrams	28
4.2.1 Class Diagram	28
4.2.2 Component Diagram	29
4.2.3 Deployment Diagram	30
4.2.4 Use Case Diagram	31
4.2.5 Sequence Diagram	32
4.2.6 Activity Diagram	33
5. System Design	34
5.1 Flow of the System	34
5.2 Logic Flow	35
5.3 System Architecture	36
5.4 Modules	37
5.4.1 Face Detection	37
5.4.2 Eye Detection	39
5.4.3 Recognition of Eye's state	41

5.4.4 Eye State Determination	43
5.4.5 Fatigue Detection	43
6. Implementation	44
6.1 Code	44
6.2 Output	48
7. Testing	51
7.1 Defining Testing	51
7.2 Testing Objectives	52
7.3 Testing Strategies	52
7.3.1 Unit Testing	52
7.3.2 Integration Testing	55
7.3.3 Validation Testing	55
7.3.4 System Testing	56
7.3.5 Output Testing	56
7.3.6 Acceptance Testing	56
7.4 Test Approach	57
7.5 Validation	58
8. Conclusion and Future Enhancement	61
8.1 Conclusion	61
8.2 Future Scope	62
9. Bibliography	63
10. References	64
10.1 Plagiarism Report	66

1. INTRODUCTION

1.1 TO SPECIFIED TOPIC

1.1.1 Definition of Fatigue

The word “fatigue” can be similar to somnolent eye that merely means as an inclination to sleep off or fall asleep. The various phases of sleep are being awake, in NREM and in REM. The NREM stage is categorized into the below three stages:

1st : Transitioning from being awake to being sleepy or being fatigued

2nd : Light weight sleep

3rd : Heavy weight sleep

For researching the driver’s temporary state of fatigue, scholars have worked on the first stage vastly, that being the fatigue part. The various accidents and crashes that take place because of driver fatigue state have variety of characteristics:

- They occur late at midnight timing being (00:00 – 07:00) or through out mid noon timings being (14:00 - 19:00). It involves one vehicle running off from the road.
- It only includes one vehicle speeding off from the road.
- They occur mostly on the highways.
- He/she is mostly alone.
- He/she is commonly a young female or male of age between sixteen to twenty five years.
- The signs of skidding or application of brakes are nill.

In respect to the above characteristics, some people use the below conditions to find out various accidents that are being caused because of fatigue:

- The alcohol level in blood being below the legalized limit for driving.

- The vehicle skid off from the road, onto the hind of some other vehicle.
- There is no sign indicating the use of brakes.
- The automobile not showing any internal mechanical defect.
- Presence of clear visibility and good weather conditions.
- Driving very close to the vehicle before our vehicle and crossing the speed limit are eliminated as potential causes.

1.1.2 Fatigue Driving

Fatigue driving is a very important issue for inflicting catastrophe in traffic. Behaviors that are associated with fatigue are within the kind of closing of eyes, nodding the head or any activity in the brain. People are able to gauge differences in physiologic signs like eye closing and opening, waves emitted by the mind, vital sign to observe fatigue state or think about physical changes like when the driver's head is leaning forward, droopy posture.

Many countries have been engaged in analysis in this space at present actively, there have been various detectors to detect fatigue that may be categorized into touchable and untouchable sorts within the industry, the fundamentals of those are as below:

- Electroencephalogram (EEG) changes: The electroencephalogram isn't on the performance of an equivalent when the cerebral cortex of the brain is in the state of inhibition or excitement. According to the EEG's distribution of the inverse of time period and wave forms, we can conclude that the working behavior of the brain thus to analyze whether or not the person in charge of the vehicle is in fatigue state. But the electroencephalogram is very much prone to any activity from outside factors and possibility could be for a big amount of variations in individual physiological response of a driver.
- Posture of Head: Once the driver is fatigued the face can perpetually be seen to be downward sloping. As per various statistics, the parametric statistic of

fatigue level and head posture is considered to be around 0.7. However, there are chances that the driver's position of head might not amend and parametric statistics turn out to be negative and therefore the system can't decide and can cause early warning to fail.

- The rotation amplitude and hold strength of steering wheel: This model detects the status of the driver by keen observance on the driving wheel's designs and motions. With driver fatigue, quantity of greatly rotation could increase and the hold strength or grip can become stronger.
- The road agent: This methodology keeps track of the time thereby finding the level of deviation of the vehicle from white strokes on the road by having a camera within the same angle of view of the driver on the automobile. This process needs a fixed starting point to be maintained and a clean view of the road and also that the involvement of outdoor environmental factors are extremely nice.

1.2 MOTIVATION

The advancement of technology permits the introduction of additional innovative answers to problems in lifestyle.

These days a lot of professions need full term concentration. Folks that service in transportation field jobs like truck and car drivers, steer-men, aeronauts of aircraft, etc. should maintain an in-depth keen concentration on the roadway so as to retaliate to sudden events like animals on road, accidents, etc.

The concentration level of driver devalues thanks to long undisturbed driving, less rest in sleeping state or any medicinal problems or disorders. Many surveys done have indicated the fact of almost thirty % of accidents occurring

due to the fatigue in drivers. Once driver keeps driving to a great extent of time than traditional amount of time, it results in an uncontrollable fatigue being caused which in turn causes exhaustion that impulses the driver to have loss of awareness or the eyes being sleep deprived.

Supported by an experiment conducted by British journal employing a driving machine, the people there have come to the point that an exhausted driver proves to be way more harmful compared to someone who has alcohol in their blood which is twenty five percent higher than the permissible limit.

Fatigue may be a complicated development that tells us the presence of a decline in alerts and acutely awareness degree of motive force. The fatigue of the person driving the vehicle typically causes a small sleep i.e. a brief dozing off or loss of total awareness lasting between a second to thirty seconds. Even if there's no direct way to discover the fatigue state many indirect strategies are often used for this purpose.

1.3 EXISTING SYSTEM

- Feature-based eye detection:

Image binarization and projection presume the fact which is the colour of eye is of a brighter colour than the dermis. High error rate and complicated processing.

- Color-based face recognition:

Identification of face is supported by the dermis' colour and the facial features formation. Problems in noisy images or low-illuminated images.

- Drowsiness Detection using Raspberry Pi:

- Raspberry Pi doesn't perform quick video processing as the processing speed is only 700 megacycle per second.
- It can't measure up to the demands of some latest software.
- It has restricted quantity of memory.

1.4 LIMITATIONS OF EXISTING SYSTEMS

The foregoing methodology, though has good accuracy, it isn't rational since delicate electrodes have to be connected straight to the driving person's figure. This can cause annoyance and intrusion to the person steering the vehicle. Additionally, while operating it would cause a lot of sweat to form on the sensory devices decreasing their power to perceive keenly.

Color-based face recognition: In this technique, the facial aspect is recognized depending on the face forms and dermis' color. Different tint space including RGB which is red, green and blue are often used for hue based face recognition. These algorithms have low accuracy in images with low lightings and in images with disturbances or unwanted data.

Feature-based eye detection: This procedure has both Image prognosis and binarization. This method assumes a fact which is the optical of a person is of a shadowy color compared to the dermis or face. As all the mentioned techniques are straight forward and consist of a higher erroneous proportion, further complex processing is required to detect the exact position of the eyeballs.

1.5 PROPOSED SYSTEM

We have proposed a system where we intend to use HAAR feature based cascade classifiers and facial landmark(present in dlib library) to localize the face and detect the regions such as eyebrows, eye , mouth and nose.

Our approach is concentrated on the state of eye in each frame being captured and calculation of optical aspect ratio, and a decision is taken to identify that the eyes are unlatched or shut. In case the eyes are shut, an alarm starts ringing.

Additionally it is free from interference, therefore it ceases from causing any problem to the driver's condition and the person driving also experiences complete congeniality with the described model. Various atmospheric factors like the state of the pathway don't influence this approach. A slight dozing off case is also identified based on the given point of limit.

Due to an increase in quantity for the automobiles on the pathway, that is a reason for direct traffic collisions, mobilizing a vehicle with a model to detect the fatigue is a necessity for the safety of the person and other different vehicle drivers and animals.

1.6 OVERVIEW

System	Sensors/Parameters	Algorithm	Accuracy
[1]	EEG, ECG	Mean power frequency	-
PODS [3]	Respiration Rate, Heart Rate, Heart Rate Variability	Power Spectrum	-
[4]	Cameras/Eyelid movement, gaze movement, gaze movement, head movement and facial expression	Kalman filtering tracking	Yawn-82% PERCLOS-86% AECS-95%
[5]	IR Camera	Thresholding, Mean	-
[6]	Camera/facial features of eyes, mouth and head	Fuzzy reasoning	Only focused on detection rate for facial tracking and face tracking rate
BCI [9]	EEG	Principal Component Analysis (PCA)	Training-92.6% Testing-74.6%
[11]	ECG, EEG	dynamic Bayesian network, first-order Hidden Markov Model	Drowsy (best)-91% Active (best)-91%
IVIS [15]	Eye movement, driving performance data	Support Vector Machines (SVMs)	Distraction detection (average)-81.1%

Fig 1.1: Overview of other existing systems

The above table illustrates the summary of existing fatigue detection systems developed by various researchers. Some researchers are concentrating on distinguishing the regions within which fatigue or temporary drowsiness state could occur however are not that great in identification of performances within the long run. Various others have showcased the system performance that

supported every outlined parameter, however failed to mix them to develop a trustful fatigue detection system. Therefore, the planned system has bigger benefits than that of the prevailing systems in varied aspects.

First, the planned system isn't just simply stricken by various external factors. For example, within the above mentioned systems, the infrared sensors or cameras aren't ready to notice lane markings faithfully and properly beneath

significant rain conditions. Moreover, these systems conjointly performed badly once the road was lined with thick snow or mud. Additionally as the sensors or cameras may be mounted, vehicle structure modifications and particularly in terms of electrical wiring is very crucial. These further tools and accessories are sometimes big-ticket and most drivers don't have any intention of paying more money on such modifications. Extra efforts could be needed to vary or replace devices within the vehicle if a sensor integration within the vehicle approach.

The industrial product have targeted on essentially the driving behavior rather than the physical state of the person driving. Additionally, the lane marking analysis system may be operated solely at speeds of between sixty and two hundred and fifty km/h and solely only if the system had detected a lane marking. In summary, the planned developed system isn't affected by the restrictions mentioned higher than and is ready to perform comprehensive fatigue analysis.

2. SYSTEM REQUIREMENTS

2.1 SOFTWARE REQUIREMENTS

- OS Supported are:
Windows 7 or higher
- Supported Development Environment:
Python 3.0 or higher

2.2 HARDWARE REQUIREMENTS

- Memory: 8GB RAM
- Processor: Intel Pentium 4 processor
- Hard Disk: 64GB or more
- Input devices: Webcam
- Output devices: Audio

3. TECHNOLOGIES USED

3.1 TENSOR FLOW

Tensor Flow is an associate open supply library for information flow and software engineering covering a spread of various processes. It is used for several practices of machine learning like NLP or natural language processing, neural networks or NN, etc and is also an allusive science library. It's a broad, adaptable scheme of libraries, social welfare and machinery that allows researchers to pursue high progression in machine learning and various programmers simply design, build and implement those routines based on Machine learning. Examination and manufacturing is also done using Tensor Flow. Stateful dataflow graphs are used for Tensor Flow computations as a unit. Tensor Flow derives its name from the operations that those types of neural networks perform on 4D information giving arrays.

- Easy model building: You can select from the multiple levels of abstractions that are offered by Tensor Flow by deciding the proper one for your desires. We can start by building and training the models by applying the high-level Keras API, that makes obtaining started with TensorFlow and machine learning very simple and efficient.

If we want some additional flexibility, we can use an early execution process which permits for iteration and debugging. For big machine learning coaching tasks, use the Distribution Strategy API for distributed coaching on completely different hardware configurations while not modifying the model definition.

- Vigorous machine learning building: Tensor Flow has invariably provided an on the spot path to production. If it's on servers, edge devices, or the web,

TensorFlow permits you to coach and deploy your model merely, notwithstanding what language or platform you utilize.

TensorFlow Extended also called as TFX, can be used if we want a complete machine learning pipeline production. For running reasoning on mobile and edge devices, we can utilize Tensor Flow light. We can execute it using TensorFlow.js for preparing the models and employing them in any JavaScript environment.

- Strong trial to research: Constructing and preparing progressive frameworks without decreasing the output or pace. The Tensor Flow software offers high elasticity and management providing choices rather like the “Model Sub classing API and Keras sensible API” for formation of advanced physiography. We use keen implementation for simple archetype and quick debugging.

Tensor Flow conjointly provides assistance to a combination of advance supplementary libraries and various prototypes to investigate, together with BERT, Tensor Flow Probability, Ragged Tensors, Tensor2Tensor.

3.2 MACHINE LEARNING

It is defined as a method involving schematizing that provides computer devices the potential for mechanically learning using information while not being expressively calculated. It implies with alternative vocabulary that the schemas amend its behavior because of studying from facts and statistical figures. Python being the finest language for machine learning contains certain collection of predefined classes particularly “numpy, pandas and scipy”, that are great to calculate in algebra and to apprehend basic methods of the expert retrieval process. This language proves as the best one for usage while operating using

machine learning and its algorithms and consists of straight forward semantics comparatively.

Machine learning is considered to be a method composed with analyzing information which helps in self-regulating the constructing of analytical model. It comes under Artificial Intelligence which is sustained by the notion stating that models would comprehend from the data, statistical similarity determination and create options that require borderline intervention with living beings. Machine learning could be a special part under Artificial Intelligence which teaches a device the way to study, while Artificial Intelligence is the discipline of imitating mortal talents.

3.2.1 Evolution of machine learning

Owing to the newest enumeration mechanics, machine learning lately is very different from the past expert systems. This has been fully inbred from sample detection and even the hypothesis stating machines can comprehend whilst not being schematized for carrying out particular chores; researchers engrossed with AI invested time in establishing if machines could pick up from knowledge. The repetitive aspect related to expert systems is vital being a consequence of archetypes that are unmasked to fresh knowledge, they have the ability to sincerely adjust. They comprehend through their earlier calculations to supply dependable, recapitulate options and outputs. This discipline is not a recent one, however it has gained strength in the present-day. Machine Learning will reach higher levels of potency once applied to Internet of Things.

While several algorithms of expert systems existed for a long duration of time, the ability for mechanically putting in advanced arithmetic computations on Big Data repeatedly, faster outputs could be a latest achievement. These are some popularly advertised samples in applications of expert systems that we are acquainted with:

- The self-driving Google car which is hyped heavily.
- Amazon and Netflix are examples of online suggestion proffers of applications of these expert systems as a way of living.
- Getting to know what the clients or buyers are talking regarding a person on Twitter is “Linguistic rule creation combined with Machine learning”.
- Fraud detection is also one in every of a lot of obvious vital operations in this world nowadays.

3.2.2 Importance of machine learning

Re-emergence of curiosity in machine learning is because of similar factors that have created Bayesian analysis and data mining a lot more famous compared to earlier times. Objects like growth of capacities and styles related to data present out there, machine activity that is low-priced and more strong, and reasonable costs for information storing.

Each of these indicate that it is possible for perfunctorily creating structural models which will examine larger, a lot of complicated data and deliver it quicker, with a lot of more correct outputs on a giant scope too. Beyond constructing accurate models, the corporation contains a higher likelihood of avoiding unprecedented risks and exclusive commercial chances. With usage of various designs to construct moulds which identify relationships, companies don't need any human intervention to take more reasonable decisions.

3.2.3 Applications of ML

Majority of the industrial units working with loads of information came across realizing the power of the technology that is machine learning. Using several learning perceptions in real time derived using this data, companies can do tasks expeditiously and receive an upper hand over other businesses.

- Financial services: Banking sectors and various trading negotiators use money in machine learning technology for 2 key purposes: to spot vital insights in data and to forestall fraud. The insights will establish investment opportunities or facilitate investors to apprehend the trade. Data processing may also establish purchasers with the profiles that are of high risk or use cyber surveillance to point out the warning signs of fraud.
- Government: Various official agencies such as those for citizen utilities and people security have an important want towards expert systems as they might require information from various origins that are used to mine for perceptions. Analysing sensor data, like the example of identifying methods in which we can increase potential and store currency. Machine learning may even make it smooth in minimizing identity theft and detecting impostors.
- Health care: It is an assertive method among the medical trade, due to the visible presence of sensory and customary devices that can evaluate a patient's health using the data in real time. This technology may even allow doctors to study the information to identify or notice danger signs that will help in improvement of the therapy and prognosis.
- Retail: Things suggested by web pages you would possibly adore supported by prior buuyings are done with the help of expert systems to investigate the shopping records. The marketers depend on expert systems for capturing information, analyzing it and using it to make the shopping experience unique for each user, implement their marketing campaigns, optimise costs, planning supplies for their merchandise, and giving insights to clients.
- Oil and gas: Searching latest sources for energy. Analyzing minerals present within the ground. Predicting the failure of refinery sensors. Oil supplying is done for the purpose of preparing it to be additionally productive and reasonable. The number of use cases of expert systems with respect to the business appears to be vast and is yet expanding.

- **Transportation:** Studying the data for detecting designs and tendencies has always been important to the business, as it relies on making paths more low-cost and foreseeing problems to obtain advantages extensively. The scrutiny of data and designing facets related to expert systems are vital instruments for citizen transport, other forms of transports in institutions and companies.

3.3 OPENCV

“Open Source Computer Vision” is the abbreviation of OPEN CV. This is an associate degree supplied B S D authorized collection which has many algorithms of developed CV procedures which are square measured for optimizing the usage of advancement rate in hardware. It is often used for expert systems, draining of images, controlling of image, many such objects. The computer vision incorporates such standard architecture. There exist shared and static libraries and a CV Namespace. In short, OpenCV is employed in our application to simply load bitmap files that contain landscaping photos and perform a mix operation between two photos so to form one image and is often seen within the background of another image. This image manipulation is well performed in a very few lines of code victimization among OpenCV and different methods. OpenCV.org may be a should if you wish to explore and dive deeper into image process and machine learning.

3.3.1 Image processing using OpenCV

Visual information is one of the most vital form of information that is first perceived, then comes cleaning, then lastly elucidated by the cerebral cortex. Cleaning images might be the methodology for carrying out few functions with a picture, so as to extract information that might be useful from it. A picture is

nothing but a 2 dimensional matrix or sometimes it may be 3 dimensional for colored pictures that was outlined using a statistical relation $g(a,b)$ function, here the b , a being the 2 co related axes being perpendicular and parallel. Relation $g(a,b)$ during any purpose offers a picture element range in a picture at that instant, here picture element range reports however scintillating the constituent may be, which RGB value ought to exist.

In case of monochrome pictures the picture element range may be of one range which constitutes the scintillating glaze of the element in the picture, here foremost regular picture element range that exists in computer memory unit picture that is the byte picture, that consists of a whole number of 1byte having such variety in doable range that is zero up to two hundred and fifty five. Since black is considered the zero value and white is considered the two hundred and fifty fifth value and rest form many various remaining shadows in grey.

Color depiction in pictures, individual parts of red color, green color, blue color should have description in every picture element, here we assume a Red Green Blue prototype and then the picture element range converts into a relation of vectors with 3 integers. Usually the 3 totally different parts are held as 3 separate 'grayscale' pictures referred to surface levels of colors, individual part in every unit of red, one for every unit of green and one for every unit of blue, that got to be recombined when displaying or processing.

A color archetype is nothing but a statistical prototype that explains the techniques using what the colors are usually depicted as tuples of numerals, mostly with 3 or 4 values or shares of colors. Once this model is related to an explicit description of however the parts are to be taken, the ensuing set of colours is termed color space.

Underlying piece of work in picture process may well have the below classification:

- Procuring of images.
- Strengthening of image and Repairing of images
- Information Extraction
- ❖ Procurement of images: Computer Vision offers such an ability in seizing the picture from a streaming video which is premade from a directory path else from camera capture feed.
- ❖ Strengthening of images and Repairing of images: Used in extemporising the photographic data. Depending on the utilization case there are many numerous strategies that may well get used to, few quite regularly used can be seen below:
 - Expansion and abrasion: These are associated with a cluster of amendments that are geomorphological and they are popularly worked in detecting the intensity bumps and treatment in noise.
 - Histogram equalization: Representing the number of picture element and disposition of strength in a picture is considered to be a histogram or bar graph. Equalization prolongs strength aim for matching with the distinction stages of dissimilarity fittingly.
 - De blurring of image: The noise encompasses one terribly unusual behaviour of the mean having value as zero. It is often which plays a major role by finding the mean value out during removal. Computer Vision exhibits many different properties in the system:
 - Fast Nl Means Denoising (): co operate in one scale grayed pictures.
 - Fast Nl Means Denoising Multi (): co operate in picture sequence captured in brief amount of your time that is the gray scale pictures.
 - Fast Nl Means Denoising Colored (): co operate using images that are colored.
 - Fast Nl Means Denoising Colored Multi() — same as on top of, except for color pictures.

- ❖ **Information Extraction:** Once the pre processed picture is prepared, data may well get extricated from the picture. The process is often half away to wherever the basic program starts the urge in achieving certain application strictly angled in the direction of top aim. The current half couldn't be normalized like the previous, however it should get distinguished on the path via increase in the units.

3.4 HAAR FEATURE BASED CLASSIFIER

Characteristics like HAAR constitute a unit image options that are digitally employed in identification of things. Such images are also specified as parallelogram polygon with sombre, lightweight locations consisting quite familiar options matching the front part of the head. Therefore, fundamentally you make an effort for positioning such choices on the entire frontal part in order to seek out each and every characteristic of the result.

In an instance:

Homogeneous tracts are divided by all faces

1. Region of eyes is sombre than that of higher buccal area.
2. Region of the nasal link is more illuminated compared to that of eyes.

Hence, these options in front part of the head can be involved in the creation of HAAR feature type options. For each and every distinguishable characteristic, there is a unique area within the frontal head.

The parallelogram characteristic outputs are:

You are going to rotate the associated parallelogram over the entire face for trying and achieving totally unrelated ranges.

1. Value of picture elements = \sum of picture element at hand in blackish area - \sum of picture element at hand in whitish area.
2. Two, three, four are the triple kinds of parallelograms used.

3.5 PYTHON

Guido van Rossum, during the late nineteen eighties has created python computer technology. In distinction the alternative well known languages like Java, C++, C and C# programming in python attempts for producing an ordinary yet well built structure of rules.

Programming in python is mostly utilized in system backend package development in organizations, companies like Yahoo, NASA, Google, ILM and CERN. Well skilled programmers have the ability to reach significant heights using programming in Python, however the greatness of it being reachable for beginners and permits them to handle fascinating issues a lot more quickly than several alternative, a lot of advanced languages that have a sharp learning curve.

3.5.1 IDLE's Interactive Shell

Python IDLE is a python program instance for integrated building that is accessible to Mac OS, Unix and Windows. In Python IDLE correspondence framework, we can write any statement of Python software straight in the framework and execute the code by clicking on Enter. Result is shown in the IDLE interactive shell.

Since it doesn't offer the simplest method to save the entered programs, the framework isn't of simplest mechanism for creating codes that are large. The intercommunication framework will be beneficial to manipulate using little statements of code from the programs.

3.5.2 IDLE's Editor

Python IDLE consists of an editor which is created inside of it. To open it, "From the menu bar of IDLE select New Window", Type the text into the editor. We can see the result in the editing area consisting of the statements of an easy

python code. We need to save the code by making use of the Save option. Save the program within a file of name ‘example.py’. The real naming convention of the document is mostly of no context, however a file named “example” tries to clearly provide description of the characters in the code. The ‘.py’ extension is compulsory in the main file of Python. You will execute the code from inside the Python editor with clicking “the F5 function key or by clicking in the Run menu by following: Run → Run Module”. Output of the program can be viewed within the window of the editor.

Shell editing tool permits the storage of the code and also to create modifications in future to them. Shell tool acknowledges the semantics of the technology. It also takes the help of multiple colors to spot light on multiple units a program may be made of. Majority of tasks happen in the editing area for program development.

3.5.3 Inbuilt Functions

abs()	delattr()	hash()	memoryview()	set()
all()	dict()	help()	min()	setattr()
any()	dir()	hex()	next()	slice()
ascii()	divmode()	id()	object()	sorted()
bin()	enumerate()	input()	oct()	staticmethod()
bool()	eval()	int()	open()	str()

breakpoint()	exec()	ininstance()	ord()	sum()
bytearray()	filter()	issubclass()	pow()	super()
bytes()	float()	iter()	print()	tupple()
callable()	format()	len()	property()	type()
chr()	frozenset()	list()	range()	vars()
classmethod()	getattr()	locals()	repr()	zip()
compile()	globals()	map()	reversed()	import()
complex()	hasattr()	max()	round()	

Fig 3.1: Inbuilt functions in the language

3.5.4 Fundamental Datatypes

The fundamental datatypes can be:

- Integers (numerical values)

For example, `c = 7 / 4` # result being 1, whole number slicing.

- Floating point data

For example, `c = 10.78`

- String data

will use either double quotes or single quotes for mentioning the string.

“xyz” ‘xyz’ are the same.

- Unmatched could possibly come among the same pattern.
- Utilization of three and two quotations to represent string data than contain both and for multiple line strings ‘ and “ within them.

3.5.5 Sequence Types

Tuple

- An ordered and immutable succession consisting things.
- An object is made with any varieties, along with mixed type and collection variety.

String Data

- They are unchangeable.
- Theoretically can be said as a tuple.

Lists

- They are mixed types of mutable ordered sequence of items.
- Ration ample of the similar functionality or syntax by all the progressive kinds being sequence of characters, lists, tuples.
- Differences being:
- Strings along with Tuples are viewed as unchangeable.
- Lists show themselves as changeable.
- Various functions that are being displayed here are allowed to be used for every progressive varieties.
- Samples that are shown have the operation performed on only one variety.

Dictionaries

Plotting kind:

- The dictionaries are developed to save the relational pointing from the group with keys and values.
- Keys are made up by any immutable type.
- Values comprise of random variety.
- Dictionaries can be used for holding the values of several unique types.

- For various key-value couples present in it, we have options to view, describe, develop, remove and change them.

Functions

- Def is used to create it by also giving to it a label.
- The Return is used to send to the calling function, any resultant output.
- Arguments are forwarded with using an assigning operator.
- Declaration of return types as well as arguments is undone.

Functions accepting the Arguments advanced

- Assigning operators are used for advancing arguments.
- Local naming conventions are allocated with the arguments advanced.
- Calling functions have no effect on assigning the names.
- Calling functions have an effect when changeable arguments are modified.

Gotchas

- An answer or reply value exists in majority of the various functionalities within Python.
- Although there is no return statement present within the program.
- None is a unique valuable that is given back to functions with no returning type.
- Function overloading concept doesn't exist in this language.
- No multiple dissimilar functions in Python could use a similar naming, though there are dissimilar arguments.
- Just like any data type, functions in this language can be utilized.

It could have:

- From parameters to the participating function.
- From returning utilities to the participating functions.
- From allocated to the identifiers.

- From set containing lists or dictionaries, etc.

Modules

- Modules are nothing but the variables and functions specified in different files.
- We can import valuables with the help of import function:

```
import module  
module.function()
```

- Namespaces are nothing but modules.
- Variable naming schemas can be arranged using these.

Whitespace

Whitespace is mainly used for placing the fresh lines of code and indentation in Python programming.

- The utilization of the new line till the finish statement of program.
- Utilization of at what time we can move to next statement ahead of time.
- Usage of such {} flower braces to highlight the parts of program in Python is avoided.

We can utilize the indentation that is standard in place of this.

- Statements with low indent are generally present out of the group.
- Nested blocks are usually used to begin statements with high indent.
- During the beginning of a fresh block a colon appears.

Example: class and function definitions.

Comments

- If we start the statements using '#', the remaining part of the statement is then neglected.
- For a newer class or function that we mention in our code, we could mention a 'documentation string' for the initial statement.

- Various software such as debugger, environment developer utilize them.

Assignment

- To set some naming schema to make a referential hold on any object is stated by the binding of an identifier within Python.
- Referential are generated and no duplicates using assignment.
- A congenital kind of naming isn't present in the naming used in Python. Several varieties are present in the objects.
- Based on data object assigned to it python can determine the type of reference.
- It is viewed towards the left hand direction of the assigning statement, when creating a title, during the initial step: a=10
- When various names attached to the citation might have gone out of range, a citation can be erased with the help of garbage collector.

4. UNDERSTANDING UML DIAGRAMS

4.1 INTRODUCTION TO UML

The “Unified Modelling Language” (UML) is the normalized lingo used in stating, envisioning, building and recording the architectural parts related to computer coding devices, in addition to corporation modeling, alternative hardware machines. The “Object Management Group” also popular as OMG in January 1997 created UML. To the union the UML 1st provision sketch was offered.

The most effective components of precursor notations that is possible to be used by all the object oriented methods are picked and integrated after producing a typical notation is the main aim of Unified Modelling Language. UML is structured to facilitate the huge variety of implementations. Therefore, it produces and builds variety of a huge span of tasks and machines. The first issue to note concerning the UML is that there are many various diagrams to use. The explanation for this can be that it makes a possibility to view at a system from many alternative viewpoints. UML is an incredibly necessary area of program enhancing activity and evolving object based program. Graphical notations are mainly used by UML for showcasing of depiction in programming work.

In IBM logically coalesced job dialect, Unified Modeling Language isn't a technique and there isn't any requirement for some particular set of project object like “artifacts”. But it gives us many varieties of architectural depictions which then used inside of a particular approach, enhances the simplicity of getting to know of an app undergoing progress. Presently we have additional features in unified modeling language other than existing architectural depictions. By putting any customary UML diagrams in your particular working product, it becomes

simpler for skilled individuals in comprehending an activity thereby enhancing the yield very soon.

Majority of the help, conventional unified modeling language diagrams consist of: class, state chart, use case, deployment, sequence , component, activity. UML have two broad categories. They are:

- Structural Diagrams
- Behavioral Diagrams

4.1.1 Structural Diagrams

These foremost depict the fixed side in the computer structure. Such fixed features depict such components in the figure, that form a major archetype even providing stability.

UML consists of below four forms in structural figurines and they have:

- Class diagram
- Object diagram
- Component diagram
- Deployment diagram

4.1.2 Behavioral Diagrams

They primarily represent the non-fixed side in the computer structure. The non-fixed part may have additionally explained for being dynamic components in the computer machine.

UML consists of below quintuple forms in behavioral figurines –

- Use case diagram
- Sequence diagram
- Collaboration diagram
- State chart diagram
- Activity diagram

4.2 UML DIAGRAMS

4.2.1 Class Diagram

It is an assemblage of classifying group, coherence, collaborations and existing links getting represented by the class diagram. Such figures are commonly existing ones for the structuring of OO classified computers. Fixed structural representation in the computer is addressed by these figures. These structures which contain functioning classifying group label fixed tasks representation in a computer. Class figures look like component structures with slight variations.

They are necessary for envisioning, stating, recording structural models, additionally in building practicable computer devices using onward and backward engineering processes. Using UML, we can utilize these figures for ascertaining of fixed aspects in the building blocks, the existing links and in defining the features to build.

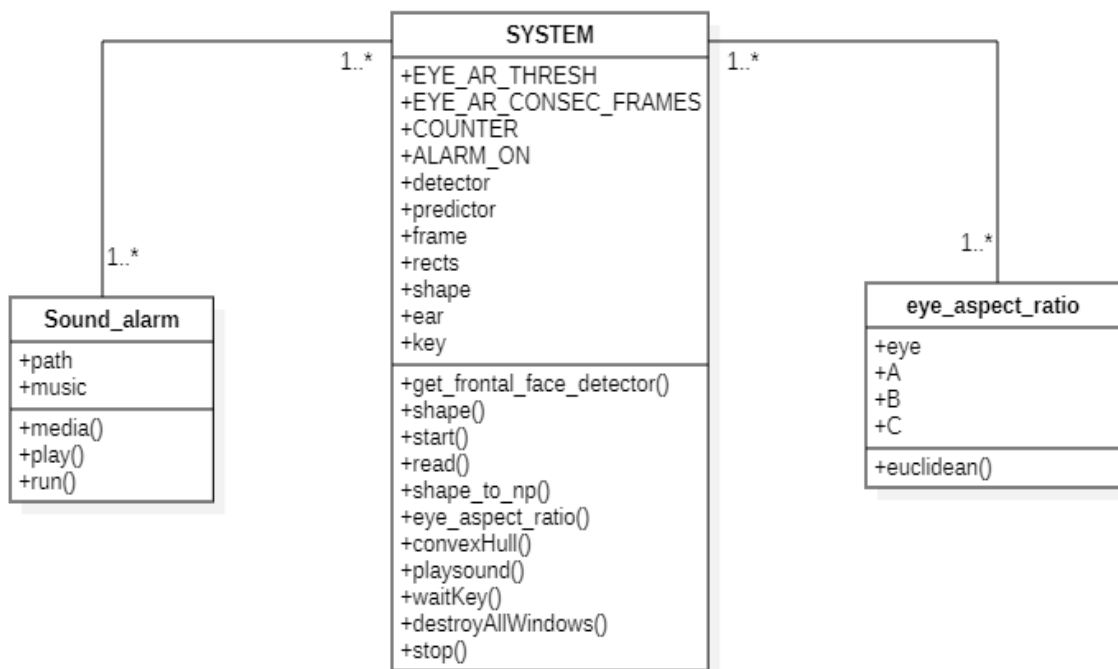


Fig 4.1: Class diagram

4.2.2 Component Diagram

It is among the different types of structures existing in the designing of materialistic facet of OO machines. These arrangements and reliance among the group of parts is shown with the help of a component diagram. A component diagrams can be used to design the fixed execution perspective in the computer device. These include the structuring of materialistic objects which are residing in the apex for example libraries, runnable programs, files documents and inventories. These figures are basically classifying structures which concentrate mostly in the computer elements. These diagrams are necessary for us to envision, state, record component-based systems but they're also useful in the process of constructing executable systems using the support of both upward and backward engineering process.

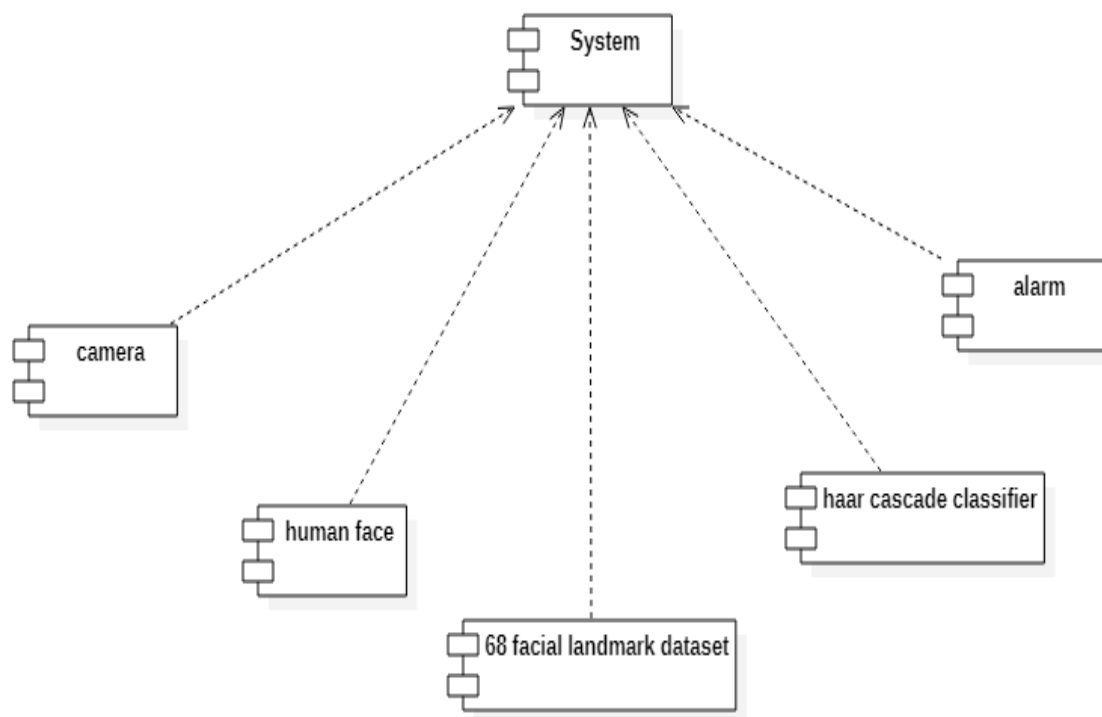


Fig 4.2: component diagram

4.2.3 Deployment Diagram

It is one among the different types of figures found within the designing of the materialistic facet of OO machines. These figures indicate the run time computational layout of the components and the nodes which rely on it. The fixed deployed representation of the computer device can be designed using these. In the major portion it includes the designing of the layout of equipment on whom the computer device runs. These figures basically class diagrams that focus mostly on the system's nodes. These diagrams are necessary for us to envisioning, stating, recording implanted, server client and scattered devices, directing runnable devices via upward and backward engineering processes.

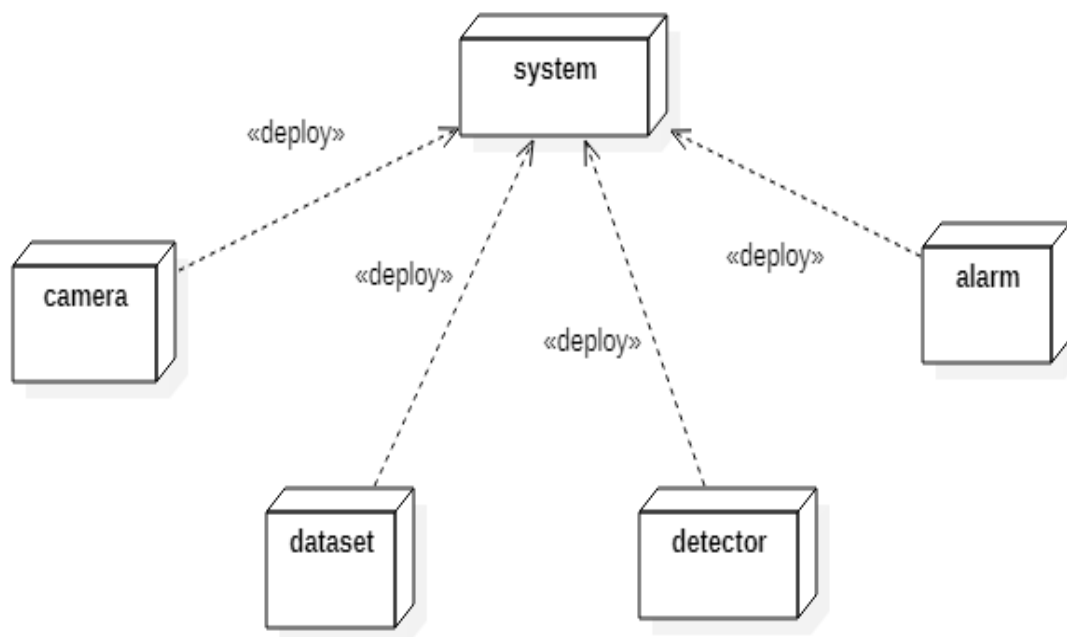


Fig 4.3: deployment diagram

4.2.4 Use Case Diagram

These belong to a collection among various plots which explain the inter activity among an actor and machine. These figures depict the action representation in the computer machine. The interactive service depicts a certain service in the computer device. Hence, these figures are employed for explaining the links between various services including the interior and exterior controllers. They can be referred to the word actors in this context.

Use case may be the illustration regarding the collection of order of steps together with variation, which the device does for producing a viewable output of the range for the actor. They also encapsulate the supposed characteristics of the computer device not including the need for clarifying however the characteristics are enacted.

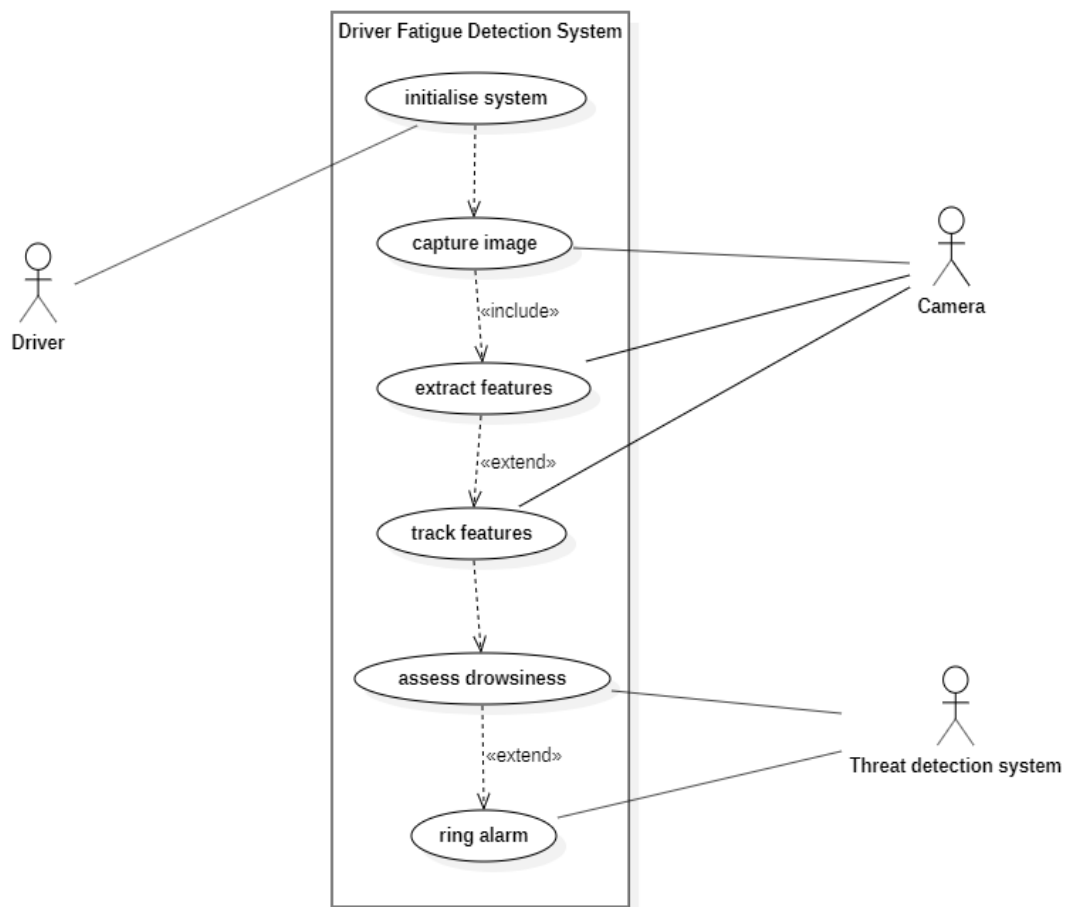


Fig 4.4: Use case diagram

4.2.5 Sequence Diagram

These figures are part of the communion structures. It's clear regarding the term interaction, stating the figure which is employed in explaining a variety of interchanges amongst various components in the structure. These interchanges may be an aspect of non-fixed characteristic in the computer device that focuses on time-based commands of communication. They depict simple normal ideas and every figure focuses on various representations of the topics. It also emphasizes on time related decree. A time-based figure displays the specific moment during which the communication was interchanged.

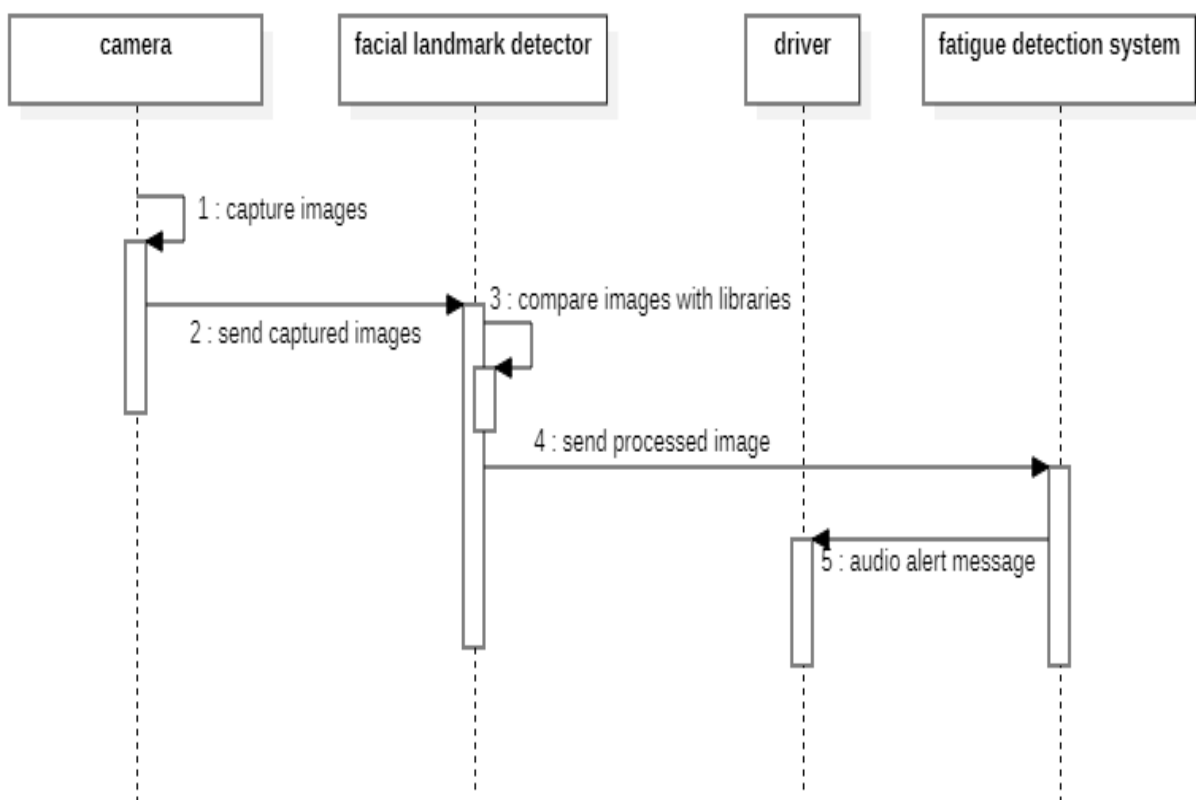


Fig 4.5: Sequence diagram

4.2.6 Activity Diagram

These figures depict the architecture in an activity or arithmetic calculation showcasing the movement of power and information from one point to another among various calculations being performed. Such non fixed representation in the computer device is addressed by these figures. They share particular pride in designing process related to functioning in a computer device and enhance the movement of power amongst the components.

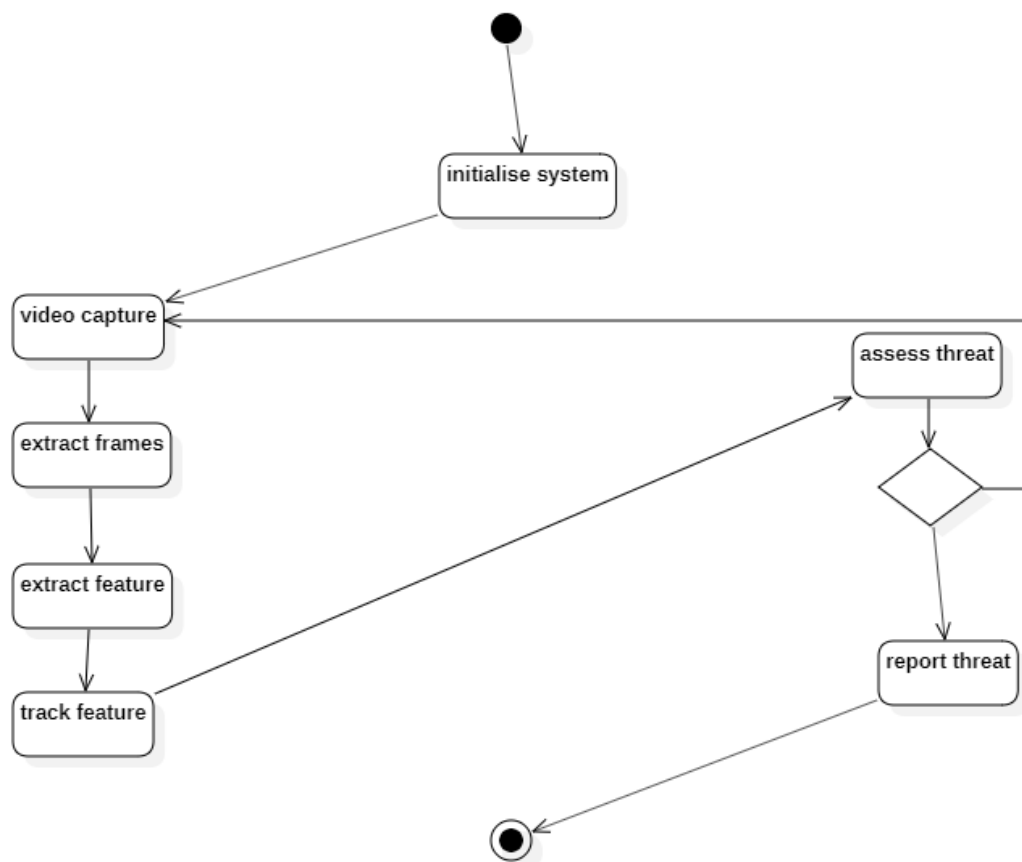


Fig 4.6: Activity diagram

5. SYSTEM DESIGN

5.1 FLOW OF THE SYSTEM

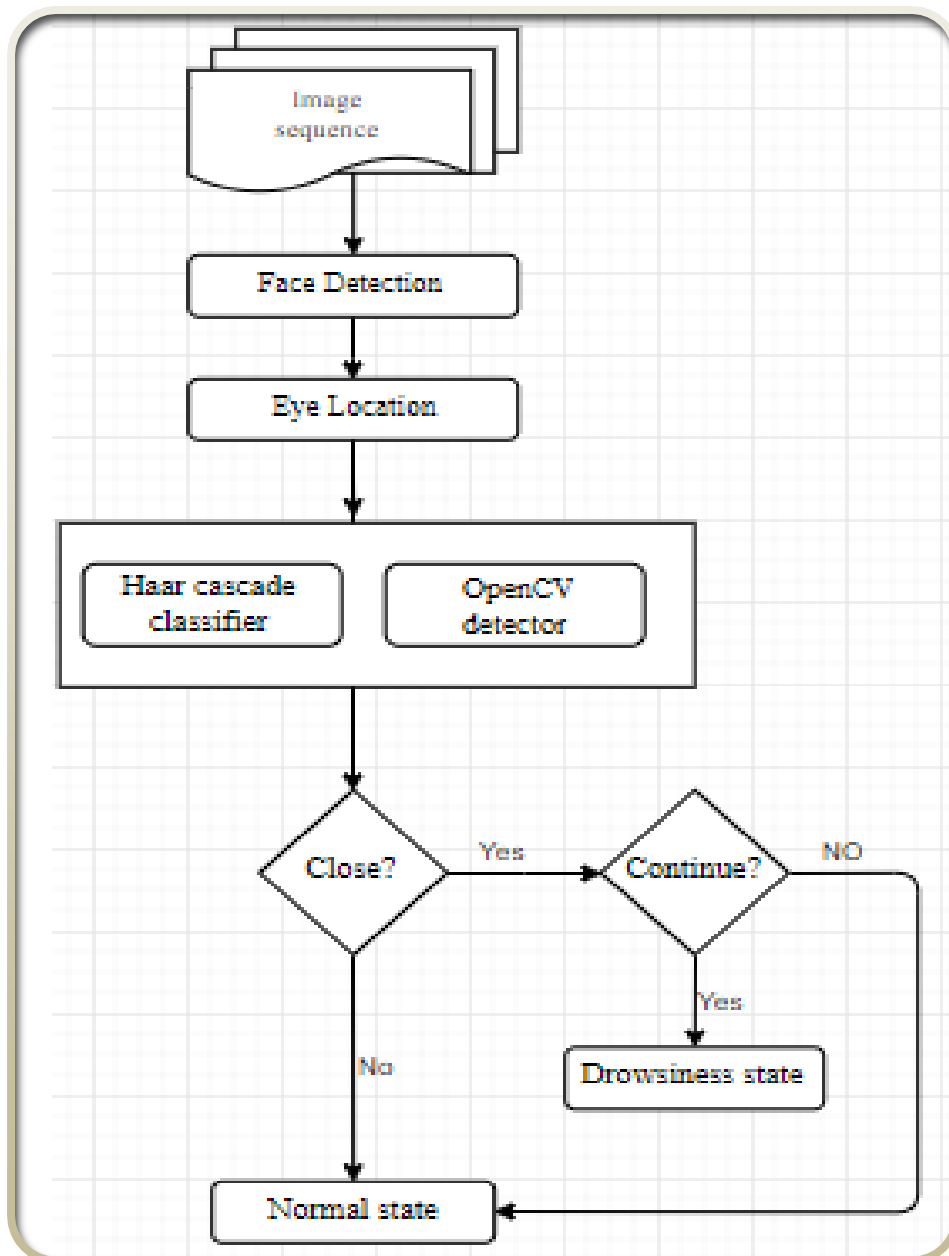


Fig 5.1: Flowchart of our system

5.2 LOGIC FLOW

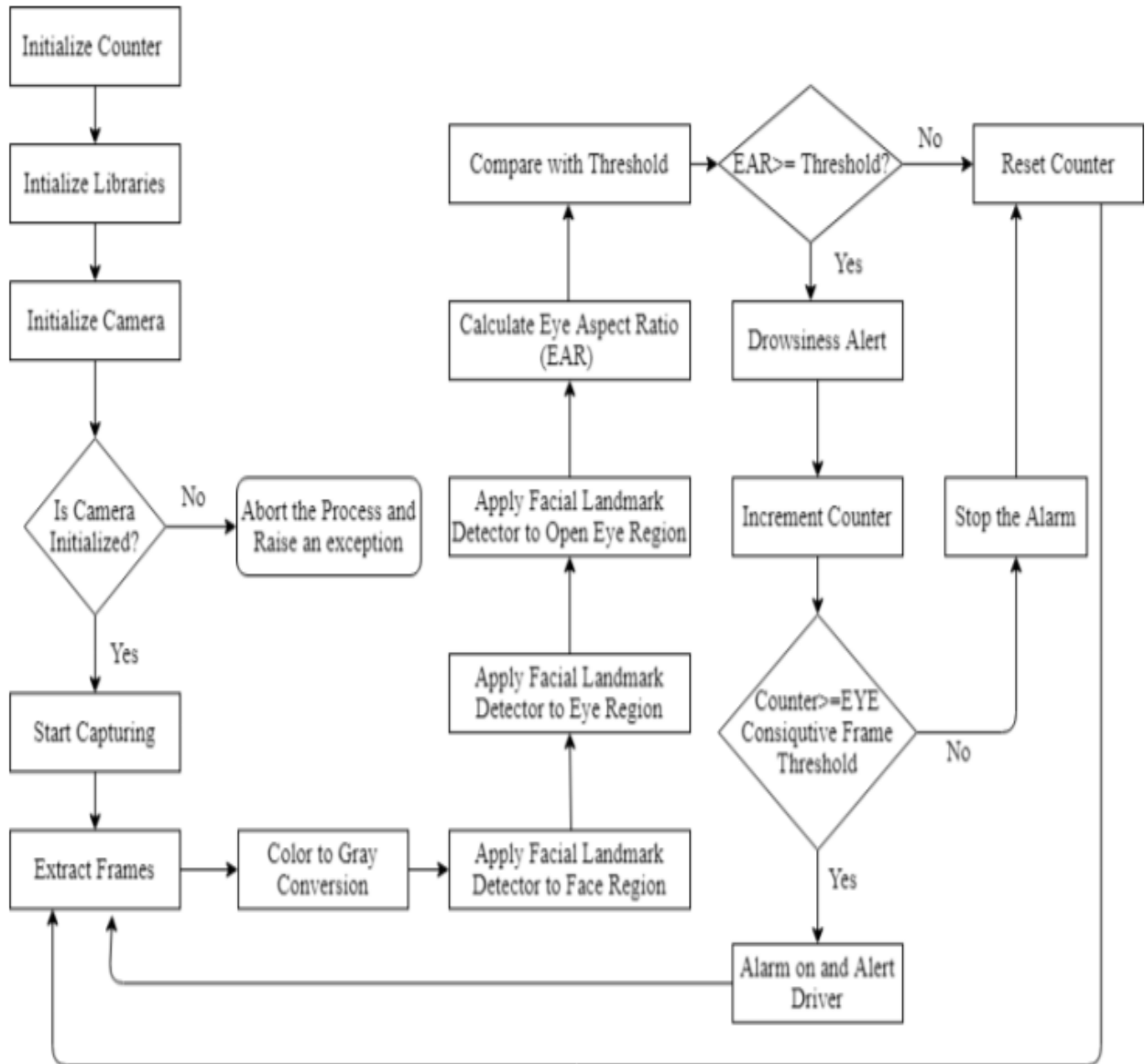


Fig 5.2: Logic flow of the system

5.3 SYSTEM ARCHITECTURE

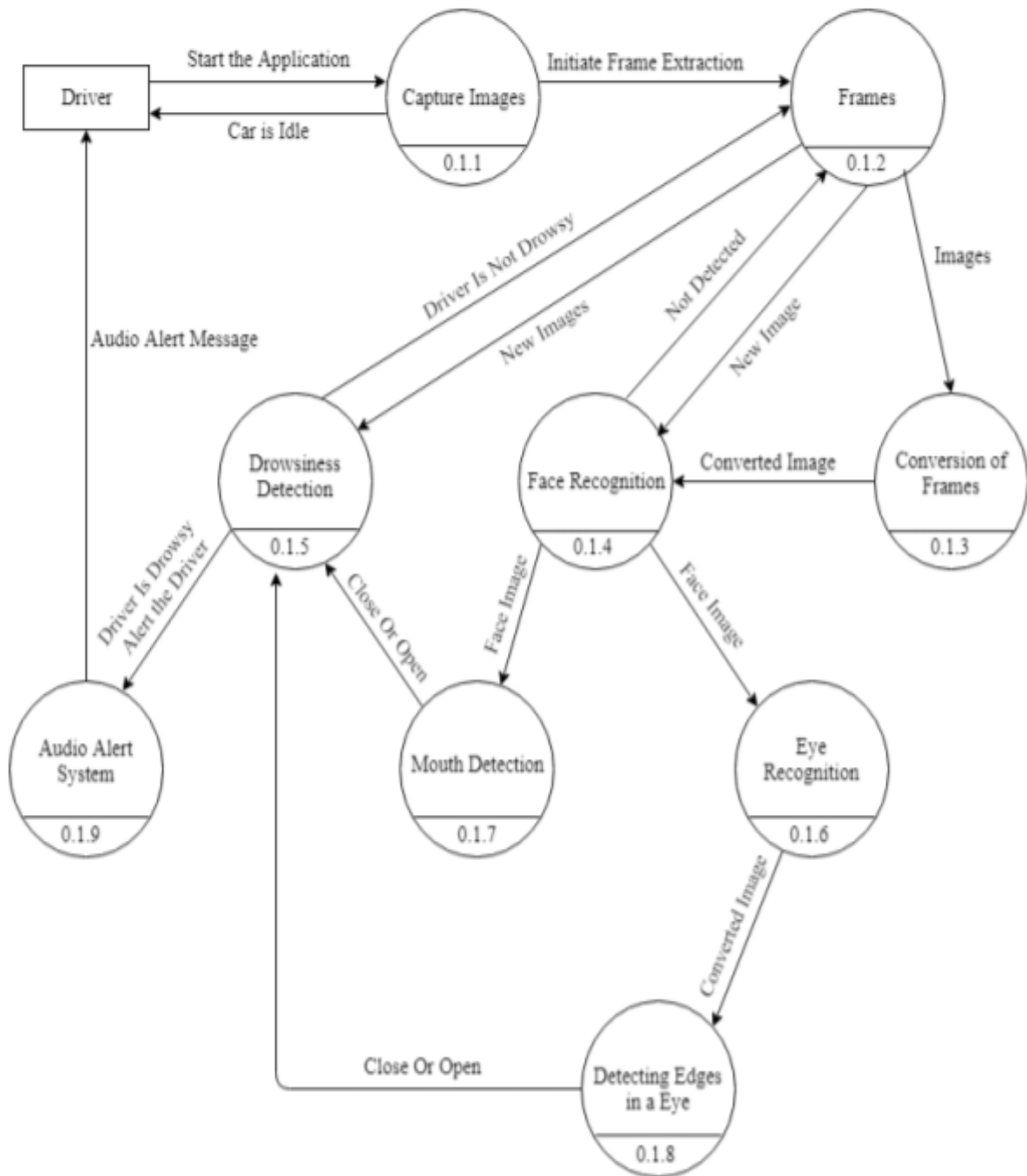


Fig 5.3: System Architecture

5.4 MODULES

5.4.1 Face Detection

For this module we are utilizing the cascade classifier HAAR. It is said to be a very efficient methodology for identifying objects which was submitted by Mr. Paul and Mr. Michael, "Rapid Object Detection using a Boosted Cascade of Simple Features" during year 2001 as part of a research. It's an ML oriented methodology wherein a surge operation is tutored using loads of constructive and destructive pictures. They are later utilized for finding components within alternative pictures. During the initial phase, the pseudo code requires many pictures of faces and pictures that don't have any faces for tutoring the semantic class. We now begin to extricate attributes. To use in this purpose, features of the HAAR displayed in the underlying pictures are utilized. It is similar to the kernel with series of layers. Every part is said to be a singular utility that is gained from removing the addition of picture elements present in the bright parallelogram from the addition of picture elements in the dark parallelogram.

To find out the face region, a classifier with HAAR like features called cascaded Ada boost is exploited. At the start a stipendiary picture has to be divided to several parallelogram locations, in various co-ordinates and ranges for the initial picture. As there is distinction in structuring of face, HAAR properties are at reasonable cost in detecting the face instantaneously. The values are computed based on the contrast of total number of picture elements in the parallelogram location. These could be depicted while using various distinguishable relations in the darker areas and brighter areas. The facial area is detected for a sample face when it goes directly into the classifier of Ada boost. Facial specimens usually get progressed while only the anti-facial specimens are not passed.

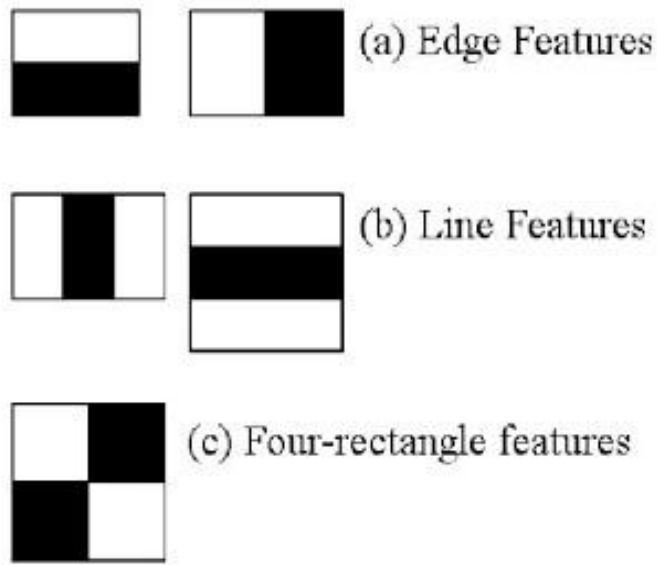


Fig 5.4: The five haar like features

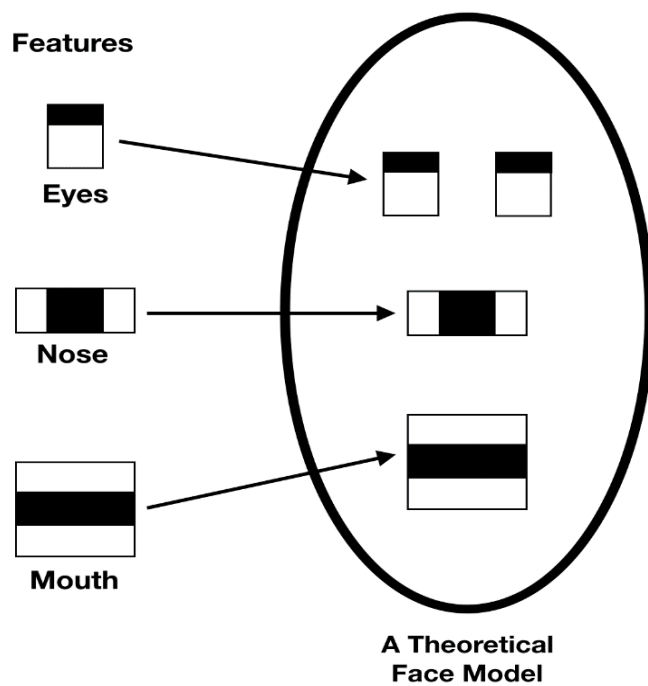


Fig 5.5: An example of HAAR feature based face detection

5.4.2 Eye Detection

Our prototype has an inclination for utilizing the face areas predictor for identifying the eye. These location markings would mainly be useful in localizing and representing primary sectors on the human facial area which are:

- Eyes
- Eyebrows
- Nose
- Mouth
- Jawline

These landmarks would be used for aligning the facial area, interchanging the frontal head, guessing the posture of the skull, detection of immediate eye opening and closing with many others successfully. For this view of detecting using landmarks, main aim is to detect major frontal head locations in one's face with the help of various techniques for estimating the shape. These detectors consist of a pair of stages:

- Within a picture we are isolating the face.
- The main regions of interest of face are predicted.

Localize within the image:

The picture of the face is constrained by the classifier based on HAAR characteristics which was stated within the initial stages of our algorithm i.e. face detection.

Detect the main region of interest on face:

Many varieties of detectors for face exist, but many techniques basically attempt to isolate subsequent regions of face:

- Mouth
- Right eyebrow
- Left eyebrow
- Right eye

- Left eye
- Nose

In the year 2014 a paper “Millisecond Face Alignment with an Ensemble of Regression Trees” written by two individuals Kazemi and Sullivan described the associate implementation of dlib library which encloses the cosmic indicator detector. The methodology is initiated by the steps:

1. The pictures used in the training set of detectors in face area locations are labelled by the hand by mentioning certain co-ordinates (a, b) around every part of the face.
2. The probable distance between pairs of input picture elements are additionally specified.

The location of these sixty-eight (a, b) positional points perform mapping of facial fabrication on the frontal head are estimated by using already trained detector present in dlib library.

The below displayed picture may be used to envisioning the indexes of the 68 location points.



Fig 5.6: 68 facial landmark

We can find and access both the eye regions by using the following facial landmark index shown below

- The co ordinates of dextral eye being (36, 42).
- The co ordinates of sinistral eye being (42, 48)

The “68 point dataset of iBUG” was used for training the dlib facial landmarks estimator. These annotations belong to it.

5.4.3 Recognition of Eye’s State

By using adaptive approach and frame to frame ferocity variation, sparse trailing and optimal passage of the area of eye can be estimated. At last, the choice is taken if the lids of the eyes shadow the eyes. A unique method is used to detect from a singular picture the revealing position of an eye, for example, while performing a relational operation finding similarity in templates of unshut eye and shut eye. The parametric model fitting to find the eyelids, a rule of longitudinal or latitudinal picturing intensity above the portion of eyes or existing figure systems.

The earlier methods generally completely inflict very powerful necessities above a set up and that is their downside. They do this hypothetically to create comparable positions of the face taken by the camera, picture resolution, lighting, movement changes, etc. Particularly commonsense rule-based strategies which utilize unfiltered pictures’ strength are more inclined to get highly effected regardless of the accomplishment they show instantaneously. We therefore consider an easy yet quite powerful algorithm for ascertaining minute opening and closing of the eye with the help of the face location finder. One variable value which reflects the grade of the unshutting of the eye can be found out using location predictors. We can find the flickering of an eye, by noting down a succession of each frame value of approximates when the eye is unshut, by using

a classifier of SVM which has been skilled by using sample made up using both fluttering and non-fluttering designs.

Eye Aspect Ratio Calculation:

The eye indicator is analyzed with each frame in the reel. The EAR value among the length and breadth is calculated for the eye.

$$\text{EAR} = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|} \quad (1)$$

here points 1 to 6 into two-dimensional indicator areas. Eye Aspect Ratio becomes very near to null in case of a shut eye but usually the same value once the eye is unshut. This is insensitive to persons somewhat with their posture of the dome. AR of the unshut eye encompasses less variation amongst people, this isn't in any variation for rotation of face plainly and consistent measuring of the picture. The mean is taken for eye aspect ratio of our eyes as the flickering of eye is done using harmony between each eye.

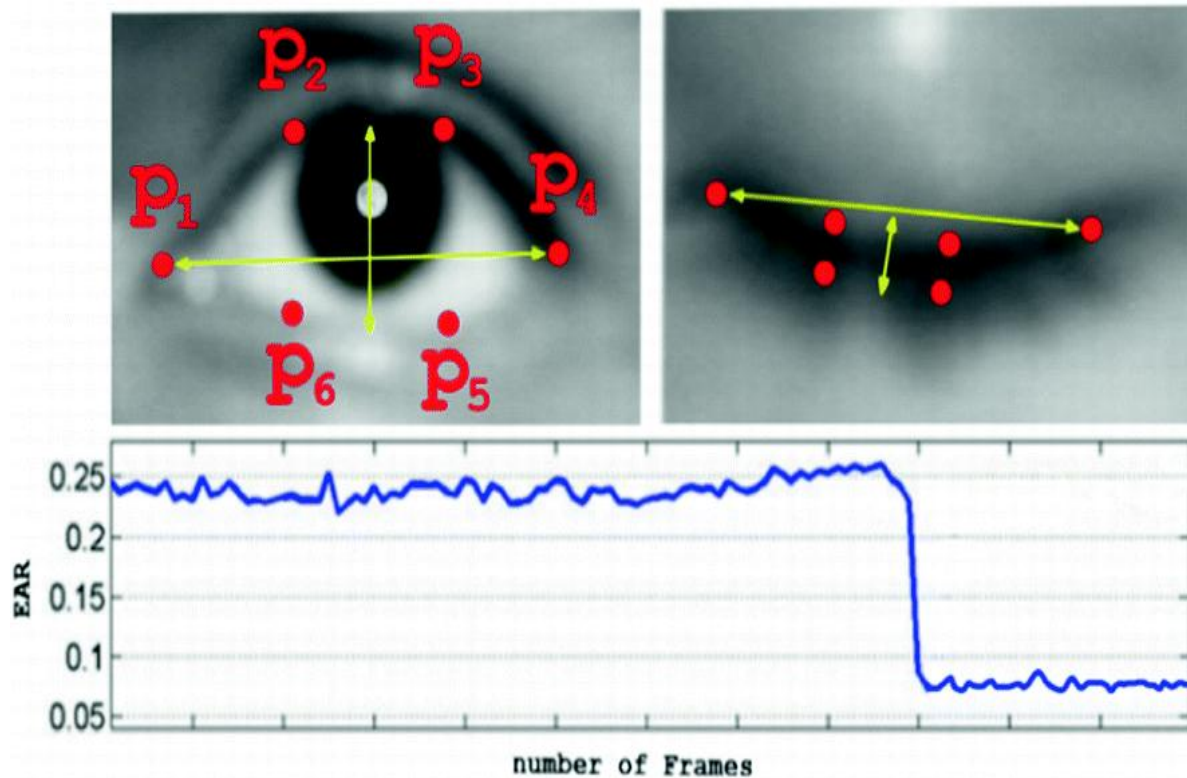


Fig 5.7: Open and close state of eye and frames of a video sequence

5.4.4 Eye State Determination

In determining the condition of the optics, we use the deliberated EAR value from the earlier stage. The condition of the eye is determined to be “closed” if the gap between eyes is zero else if it is almost 0, it is classified “open” in any other scenario.

5.4.5 Fatigue Detection

In the last part of the process, a previously set algorithm for fatigue determination is used to work out the state of the person steering the vehicle. 100ms to 400ms i.e. 0.1s to 0.4s of a second is found to be the flicker period of an average individual. Therefore, in case someone is fatigued their closure of eye should be beyond the given limit. A timeframe equal to five seconds is fixed in the model. Fatigue can be detected in case the eyes stay shut for 5 seconds or higher and an alert pop regarding this is triggered.

6. IMPLEMENTATION

6.1 CODE

```
#import all the required packages
from scipy.spatial import distance as dist
from imutils.video import VideoStream
from imutils import face_utils
from threading import Thread
import numpy as np
import pygame
import argparse
import imutils
import time
import dlib
import cv2
from playsound import playsound

#function to ring an alarm to alert fatigue
def sound_alarm(path):
    music = pygame.resource.media('alarm.wav')
    music.play()
    pygame.app.run()
```

```
#function to determine if the eye is open or closed
def eye_aspect_ratio(eye):
    A = dist.euclidean(eye[1], eye[5])
    B = dist.euclidean(eye[2], eye[4])
    C = dist.euclidean(eye[0], eye[3])
    ear = (A + B) / (2.0 * C)
    return ear

ap = argparse.ArgumentParser()
ap.add_argument("-w", "--webcam", type=int, default=0,
    help="index of webcam on system")
args = vars(ap.parse_args())

EYE_AR_THRESH = 0.3
EYE_AR_CONSEC_FRAMES = 48
COUNTER = 0
ALARM_ON = False
print("[INFO] loading facial landmark predictor...")

detector = dlib.get_frontal_face_detector()
predictor = dlib.shape_predictor("68 face landmarks.dat")

#finding the eye index
(lStart, lEnd) = face_utils.FACIAL_LANDMARKS_IDXS["left_eye"]
(rStart, rEnd) = face_utils.FACIAL_LANDMARKS_IDXS["right_eye"]
```

```
print("[INFO] starting video stream thread...")
vs = VideoStream(src=args["webcam"]).start()
time.sleep(1.0)

while True:
    frame = vs.read()
    frame = imutils.resize(frame, width=450)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    rects = detector(gray, 0)

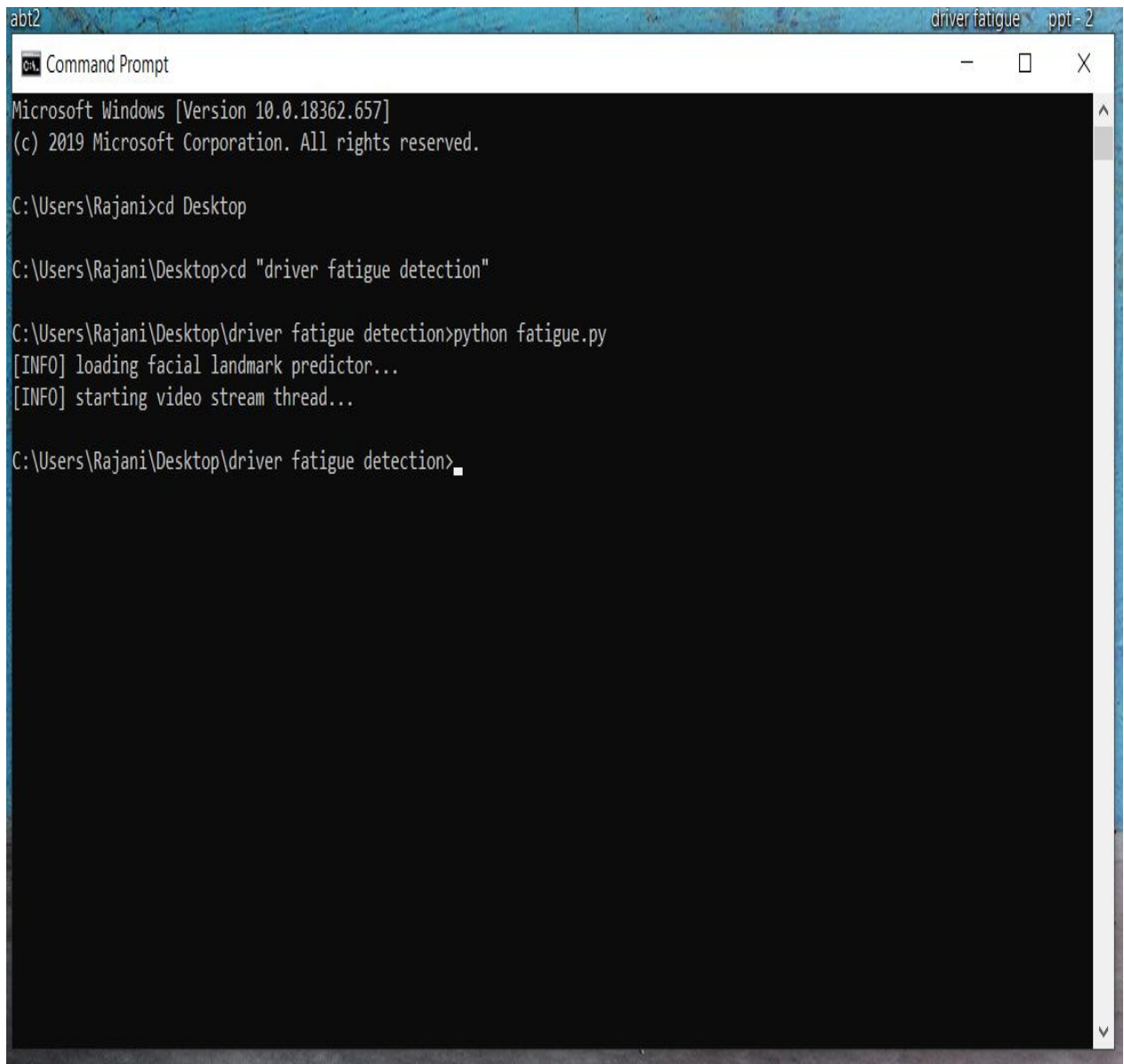
    for rect in rects:
        shape = predictor(gray, rect)
        shape = face_utils.shape_to_np(shape)
        leftEye = shape[lStart:lEnd]
        rightEye = shape[rStart:rEnd]
        leftEAR = eye_aspect_ratio(leftEye)
        rightEAR = eye_aspect_ratio(rightEye)

        ear = (leftEAR + rightEAR) / 2.0
        leftEyeHull = cv2.convexHull(leftEye)
        rightEyeHull = cv2.convexHull(rightEye)
        cv2.drawContours(frame, [leftEyeHull], -1, (0, 255, 0), 1)
        cv2.drawContours(frame, [rightEyeHull], -1, (0, 255, 0), 1)
        if ear < EYE_AR_THRESH:
            COUNTER += 1

    if COUNTER >= EYE_AR_CONSEC_FRAMES:
        if not ALARM_ON:
            ALARM_ON = True
```

```
        cv2.putText(frame, "DROWSINESS ALERT!", (10, 30),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
        playsound('alarm.wav')
    else:
        COUNTER = 0
        ALARM_ON = False
        cv2.putText(frame, "EYE: {:.2f}".format(ear), (300, 30),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
    cv2.imshow("Frame", frame)
    key = cv2.waitKey(1) & 0xFF
    if key == ord("q"):
        break
cv2.destroyAllWindows()
vs.stop()
```


6.2 OUTPUT



```
ab22 driver fatigue ppt - 2
Command Prompt
Microsoft Windows [Version 10.0.18362.657]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Rajani>cd Desktop

C:\Users\Rajani\Desktop>cd "driver fatigue detection"

C:\Users\Rajani\Desktop\driver fatigue detection>python fatigue.py
[INFO] loading facial landmark predictor...
[INFO] starting video stream thread...

C:\Users\Rajani\Desktop\driver fatigue detection>
```

Fig 6.1: Console

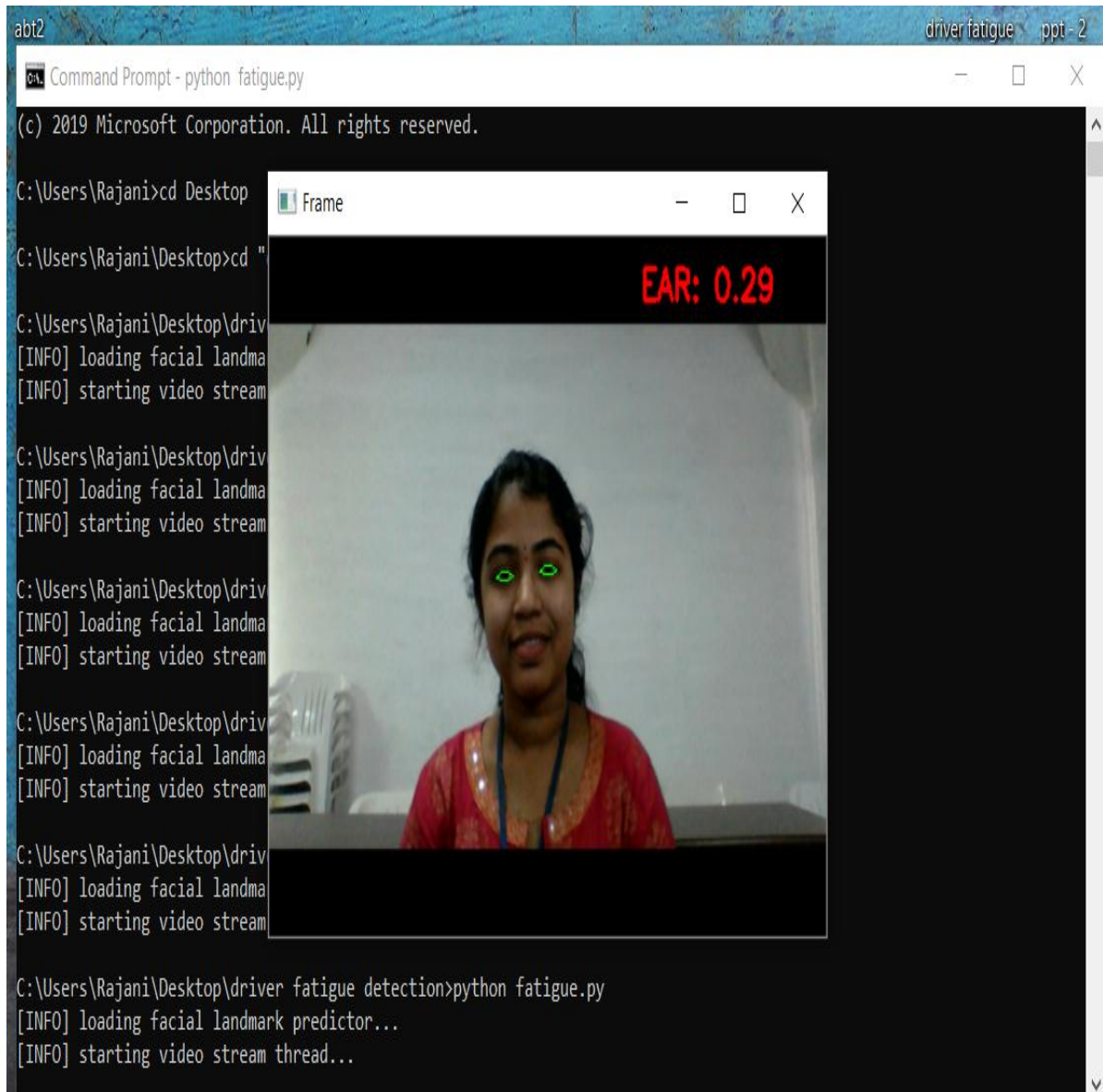


Fig 6.2: Locating of eye and calculating the state

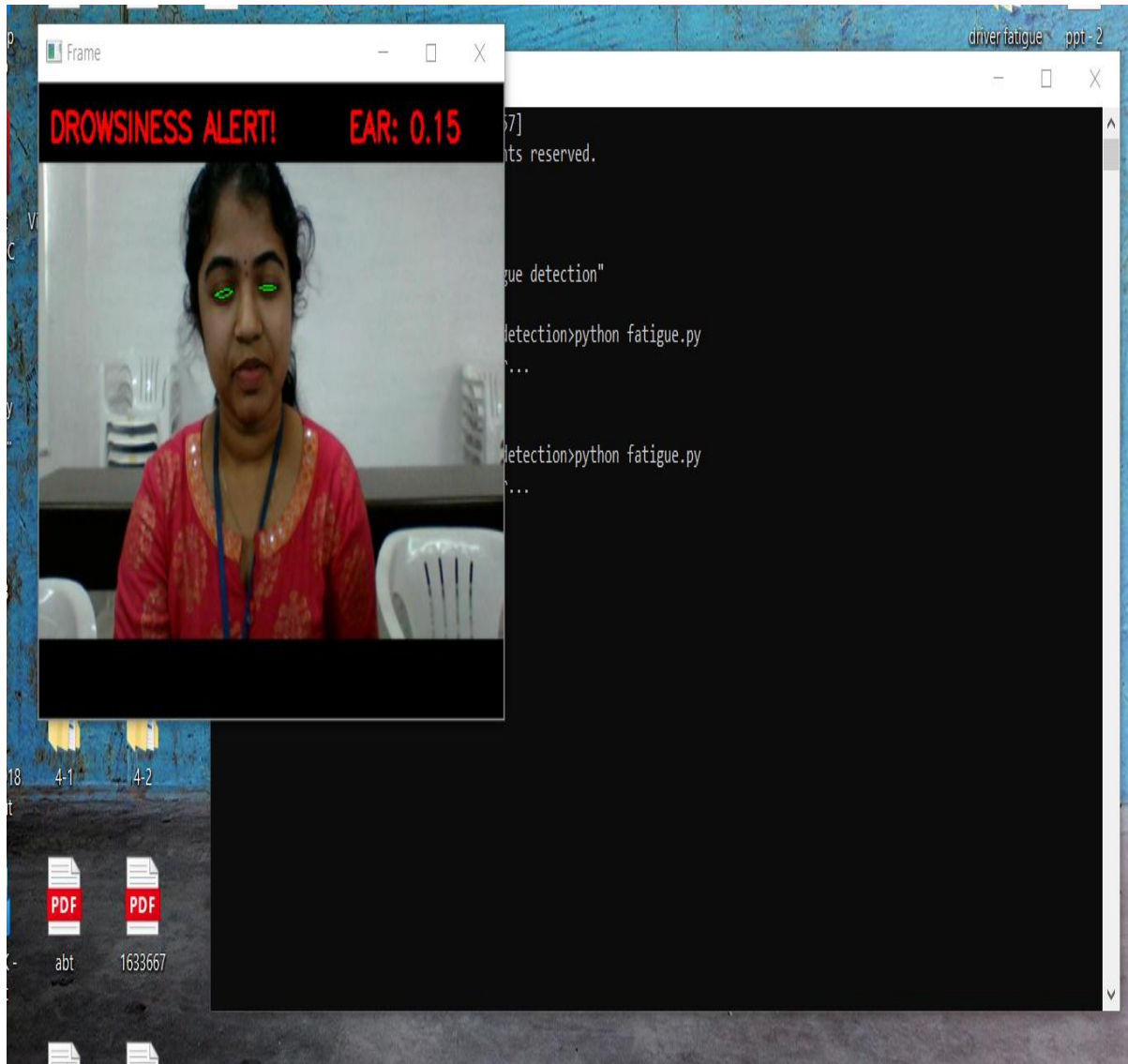


Fig 6.3: Fatigue Detection

7. TESTING

7.1 DEFINING TESTING

“Testing” is said to be a technique used for validating the device and the various components having the interest for searching if the model developed assures the desired requirements or doesn't. The duty ends up in a specified, anticipated and difference among the outputs. If needed in other lines, validation gets used for performing the execution of the device in order to find any loop holes, defects or missed requirements in context to a specific need or requirements. Testing is an important component of computer device quality checking and showcases the final word evaluation of detailing, style, modelling, committal to writing, coding. It shows an engrossing peculiarity in the computer program.

Testing is that the method of detecting and investigating the errors. It performs a really important aspect in verification of standard and in guaranteeing of requirements in computer programs. Here outputs of verification get used throughout the conservation phase conjointly. Generally, the validation level consists of the verification of a developing machine using multiple validating sets of data. Collecting and organizing of data and information has a significant part within the validation of machine. Faults which are detected when validating the machine code are then modified to gain correct results with the help of validating rules and such modifications get jotted down to be used in further days. Hence a set of validations and verifications are done on the planned program, prior to the program getting prepared to perform execution in user's environment. Thus main aim for validating is depicting that the code is working fine by displaying that there are zero defects or bugs. Here the fundamental reason for the validation

plane is mainly to identify any faults which could be existant in the system code. Thus validating provisions many programming developers to develop programs that would reach all necessities of the customer, avoid sudden events and there by working to improvise the prolonged conservation of the software. Based on various method used for validating and the requirements of programs, various suitable techniques are being used. Wherever automation of the process could be done is being done.

7.2 TESTING OBJECTIVES

For the most important purpose we perform evaluation for identifying the number of defects, methodically and involving minimal time, effort. Affirm organizing, altogether, performing testing is indicated to be the method in which running a code having the intention of detecting a bug. The so far uncovered and undetected defects are discovered by a victorious check. Decent scenario of testing consists of the features that have the best chance of detecting errors, in case they exist. This software has greater and dependable benchmarks.

7.3 TESTING STRATEGIES

7.3.1 Unit testing

This testing could be the magnitude for assessing application methodologies wherever single parts or elements in the programming package or computer device get cross checked. It even additionally referred to be known as testing of components i.e it points out to the checking done for the verification of each operation regarding any particular block of program. Its aim being validation

of every component in the programming system to work as modeled. All modules should gain high praise within this testing prior to the beginning of integrated units assessing. This kind of testing has a main aim of separating every component in the code to state if the unique components are valid in terms of functionality and practical requirements.

This checking concentrates accreditation effort in the similar component in program device modelling. It is also termed to be form validation. In such validating step, each component is said to be functioning adequately, the relations are the results which are anticipated from the piece of code. Every piece of code is validated by handing totally unrelated group of content, once while evolving present piece of code and additionally during completion of the building process for it to make each piece of code operate with no defects. Its initial contents are checked for sureness after being received from the user. Each piece of code is validated by considering the subsequent plan of actions:

White Box Testing

Such testing is one among the program validating methodology within whom the basic implementation or architecture or design in the units undergoing testing are realized by the tester. The person testing selects contents for trying various methods via the program and finds out all acceptable results. Such validation is cross checked on the last part in interaction with user phase and into the kernel of a system. It is a validating methodology wherever any component is considered during any instance of time and validated completely in a descriptive phase to seek out the utmost attainable errors which could occur. You have performed testing in an order on each bit of program, ensuring validation that each descriptive line within the program is run a minimum of one time. These are often used on the components, integrated unit and device planes in program

validating methodology. This is generally executed at the component plane. This would have a glance at ways within a component, techniques among various components during the combination of units, among the child devices at the device plane test.

This testing can be referred as “Glass Box Testing”. We've got to generate a tuple of various validating check basic points, that are employed by ascertaining the attainable mixtures for running methods via the program in each block phase. Such validation has been used for seeking out information within the following categories:

Confirming the rationality of the execution of inner capability structures.

Guarantee that all the non dependent paths are carried out.

Carry out the rational choices based by the correct and wrong sections.

Black Box Testing

This type of validation is the program validating methodology during the phase where test data gets derived from desired specific necessities with no reference to the ultimate program structure. This handles the system program to be a “black box”, scrutinizing practicality with no realization of internal implementation. Such validating methodology contemplates the piece of code to be one component and verifies all components at interaction and exchanges of messages within different pieces of code other than taking in at all indications in declaration side i.e., at present all modules are considered to be black box which may grab some contents and provide results. The results in case of given group of content combos gets moved forward to different pieces of code. Such assessing is being utilized to find out within the below mentioned kinds:

The starting point and the ending point of errors.

Wrong or lost operations

Production faults

7.3.2 Integration Testing

Such testing is one among the type of program assessing methodology in which various distinguished components are square measured and assessed into a whole single unit. Main aim of such stages of validation is to show defects within all combination between combining components. When the unit testing is done the we've got to do the component combining validation. Main objective at present is to visualize the blocks if they may get combined perfectly, majority stress lies above existing modules whose interfaces are getting tested. It works to show faults within the interfaces and interaction among integrated parts i.e modules. More and more larger units of tested software system components in the structural pattern are combined and validated till our application code executes as a complete part.

The validation of combined elements of our application is done to see if they perform properly along with each other thus being the main concept behind of integration testing. There exist two ways of doing integration testing. They're “bottom up integration testing” and “top down integration testing”. The blocks of code are then integrated within the validating phase. After which the whole code is validated as an entire unit.

7.3.3 Validation Testing

In the conclusion result of the performed above validation, the machine code is entirely summoned like a whole unit, the existing defects are discovered, corrected and then the last set of machine code checking gets started. Such verification validates that the merchandise style meets the required satisfaction of the supposed usage i.e., validating machine code levels up to the customer necessities.

7.3.4 System Testing

Such assessing is one among the type of program validating method in which an entire, combined units or program code is validated. Once all parts are integrated, the application as an entire package is tested strictly to ascertain that it meets quality standards. In this the entire package is validated. A referable doc for such method are that of necessities files i.e. requirement files, then the aim is to ascertain that the application code reaches the necessities level. Then the complete software gets validated with necessities in software and then verified if or not entire necessities of software are reached.

7.3.5 Output Testing

Thereafter the execution of above testing, all subsequent steps are results based on validation regarding the planned model, as no device may be helpful in case it doesn't turn out mere required output within the said method. Here outputs derived are then shown by the device are taken into thought or validated by considering the customer regarding the specification needed by those customers. Then the resulting specifications are taken into account in a pair. First is the display window and second is on the document that is printed.

7.3.6 Acceptance Testing

This has to be one among the type application assessment methodology in which the machine is assessed to find satisfactoriness and fittingness. The aim pertaining to the test is to gauge a machine's docility along with trading needs and identify if the result is sufficient to be dispatched. Pragmatic information related to the customer is used for carrying out acceptance testing to depict the fact that packages are functioning convincingly. The inner cognition of the code isn't highlighted as this testing is concentrated on visible conduct of the machine.

We've obtained certain amount of information and assessed if it is functioning properly with our project work. In order that the biggest kinds related features with uniform category are implemented directly at a time, sample scenarios ought to have been hand-picked. A crucial component related to application package building is this testing. This method involves pointing out faults and absent functionalities along with conjointly the whole accreditation for seeing if or not goals and client needs are pleased. The major issue related to prosperity of the model is software machine being accepted by the end customer. By perpetually having grips over the models end clients during the phase of building and creating modifications when required, the model taken into account has been assessed if clients accept it. Subsequent ideas are often used for finishing it off.

- Input screen pattern

- Output screen pattern

- Menu driven system

7.4 TEST APPROACH

There are a pair of methods for performing testing:

- Bottom up approach

- Top down approach

Bottom up Approach

Assessment is often conducted ranging to minute and stunted stage units and continuing with each of those in particular instances. With respect to every unit present as part of the this assessment, a brief application enforces all the units in addition to supplying all required information. Once lodged at spans as part of a bigger machine, the units execute those methods. Once the base stage units are

assessed, then the concentration is put for ensuing stage. This utilizes the bottom stage things and those are evaluated on an individual basis and then conjoined to certain beforehand investigated base stage segments.

Top down Approach

Testing begins with top stage segments in this approach. Since the careful actions carried out at certain times inside the base stage procedures barely seem to be satisfied then the receipts are being noted. A shell of units can be a stub known for top stage segments that attained correctly at one time are allowed back with the help of communications for a vocation unit depicting the apt exchange that took place. For ascertaining the surety level of the base stage units, no effort is created.

7.5 VALIDATION

Prototype developed was thoroughly assessed in addition to being enforced with success by also guaranteeing that each one of those necessities like mentioned within front end description of code prerequisite cataloguing to be fully consummated. Just if there are any inaccurate fault alerts for loaded data they will be exhibited.

Test Case ID	Test Scenario	Expected Output	Actual Output	Status
1	Both eyes are open	Eyes detected	Eyes detected	Pass
2	Both eyes are closed	Eyes detected and alarm is sounded	Eyes detected and alarm is sounded	Pass
3	Wearing lenses	Eyes detected and alarm is sounded	Eyes detected and alarm is sounded	Pass
4	Spectacles	Eyes detected and alarm is sounded	Eyes detected and alarm is sounded	Pass
5	Goggles	Eyes detected and alarm is sounded	Eyes detected and alarm is sounded	Pass
6	Blinking	Eyes are detected and alarm is sounded	Eyes are detected and alarm is sounded	Pass

7	Looking down	Eyes detection is lost, and alarm is sounded	Eyes detection is lost, and alarm is sounded	Pass
8	Face partially turned to the side	Eyes detected and alarm is sounded	Eyes detected and alarm is sounded	Pass
9	Face fully turned to the side	Detection is completely lost	Detection is completely lost	Pass
10	One eye open and one eye closed (wink)	Eye detection is partially lost, alarm is sounded	Eye detection is partially lost, alarm is sounded	Pass

Table 7.1: Testing table

8. CONCLUSIONS AND FUTURE ENHANCEMENT

8.1 CONCLUSION

Here we demonstrate how quantitatively incontestable the HAAR feature-based cascade classifiers and detectors of facial landmark based on regression which are used for faithfully gauging the positive pictures in the face and calculate the level of eye openness. Whereas they're strong to low quality of image i.e the less resolution of image in a very giant level and in a wide area.

Limitations:

Spectacle use: For observing driver's eye state here, if the user utilizes goggles or spectacles it would be troublesome. Because the state is vastly dependent on lighting thence spectacle reflection could provide shut optic like an open optic as the result. For this reason, it is necessary to avoid lighting for the capturing camera to see the amount of the eye that is closed.

Number of faces: The capturing camera can see many number of faces and an unsought result could come up if multiple faces arises within the frame. Due to completely uniquely identified features of various people. Hence, you want the creation of a certain method that solely focuses on the face of the person driving at a range the vary based on the camera. Also, the speed of detection might decrease due to operation on various faces in a single image.

The model is proficient of accurately pointing out the eye's position. We could successfully observe status of the driving person's fatigue with great help of the quadruple specifications of the state of eye. So as to boost the accuracy grade, our system could also use various other means of additional support that,

- o Pursuit of Road
- o Positioning of Head

8.2 FUTURE SCOPE

The developed prototype is meant to identify the fatigue state of eye to provide an alarm warning or notification prompts in the type resembling a phonic alarm. However there are chances that the immediate action of the person driving when alerted is not sufficient in preventing of infliction of the crash which means that the accident could still happen if the driver is delayed in reacting after the alert. Thus for making sure it doesn't happen one could plan, work towards a system driven by motors and attach it to the alert in order for the automobile to hamper when obtaining the alert mechanically.

We can additionally give the user with an automated application which is able to give with the knowledge of his/her sleepiness level throughout the entire journey. The user can recognize his/her traditional state, drowsy state and the amount of times he/she blinked the eyes according to the amount of frames captures.

Future work will mostly focus on the outer factors as stated by the above points i.e vehicle status, various information and data regarding the driver and sleeping hours. Currently there is no facility of automatic zoom of camera during the video stream. If the technology advances in this direction then we can work on automatic zooming on the eyes once they are located.

9. BIBLIOGRAPHY

Names of Websites referred are:

<https://realpython.com/face-recognition-with-python/>

<https://www.pyimagesearch.com/2017/04/24/eye-blink-detection-opencv-python-lib/>

<https://www.pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/>

<https://www.pyimagesearch.com/2017/04/10/detect-eyes-nose-lips-jaw-dlib-opencv-python/>

<https://www.codeproject.com/Articles/26897/TrackEye-Real-Time-Tracking-Of-HumanEyesUsing-a-camera/>

https://docs.opencv.org/3.4/d7/d8b/tutorial_py_face_detection.html/

<https://www.learnopencv.com/training-better-haar-lbp-cascade-eye-detector-opencv/>

10. REFERENCES

- 1. Omkar**, Revati Bhor, H.V. Kumbhar, Pranjali Mahajan, Survey on Driver's Drowsiness Detection System, Vol 132, 2015.
- 2. Christy**, Jasmeen Gill, A Review: Driver Drowsiness Detection System, IJCST, Vol.3 Issue 4, Jul-Aug 2015.
- 3. Rajasekar.R**, Vivek Bharat Pattni, S.Vanangamudi Drowsy Driver Sleeping Device and Driver Alert System, IJSR, Vol.3, Issue 4, 2014.
- 4. Rashmi Bhadke**, Swathi Kale, Nanasaheb Kadu, Anuja Sali, Drowsiness Detection and Warning System, IJARCSST, Vol.2, Issue.02, 2014.
- 5. Eriksson M** and Papanikolopoulos, N.P, Eye-tracking for Detection of Driver Fatigue, IEEE Intelligent Transport System Proceedings, pp 314-319, September 2014.
- 6. Zutao Zhang** and Jiashu Zhang, A New Real-Time Eye Tracking for Driver Fatigue Detection, 6th International conference on ITS telecommunication proceedings, IEEE 2006.
- 7. Chun Hai Fan**, Chih-Yuan Chen, Yi Chang, Driver Fatigue Detection Based on Eye Tracking and Dynamic Template Matching, International conference on networking, sensing and control, IEEE 2004.
- 8. Perez**, Claudio A. et al, Face and Eye Tracking Algorithm Based on Digital Image Processing, Man and Cybernetics Conference, IEEE System, vol.2, pp. 1178-1188, February 2012.
- 9. K Das**, R N Behera, A Survey on Machine Learning: Concept, Algorithms and Applications International Journal of Innovative Research in Computer and Communication Engineering, Volume 5, Issue 2, pp. 1301 – 1309, 2017.

- 10. Kriegman D J**, Ahuja N, Yang M H, Detecting Faces in Images: A Survey ,
IEEE Trans. Pattern Analyzation, Machine Intelligence, Vol 24, No 1, Jan
2002, pp 34–58.
- 11. Shuming Tang**, Guanpi Lai, Hong Cheng, Fei-Yue Wang, Qing Li, Nan-
Ning Zheng, “Toward Intelligent Driver-Assistance and Safety Warning
Systems”, Intelligent Transportation System, IEEE 2004.
- 12. Clausi David**, Chakraborty Samrjit, Scharfenberger Christian, Zelek John,
“Anti-Trap Protection for an Intelligent Smart Car Door System”, 15th
International IEEE Conference on Intelligent Transportation System,
Anchorage, Alaska, USA, September 16-19, 2012.

10.1 PLAGIARISM REPORT

CSE-IV-Projects			
ORIGINALITY REPORT			
1%	1%	0%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS
PRIMARY SOURCES			
1	www.tensorflow.org Internet Source	<1%	
2	ethesis.nitrkl.ac.in Internet Source	<1%	
3	www.irjet.net Internet Source	<1%	
4	Submitted to University of Hong Kong Student Paper	<1%	
5	mca.griet.ac.in Internet Source	<1%	
6	Submitted to Tshwane University of Technology Student Paper	<1%	
7	ts.catapult.org.uk Internet Source	<1%	
8	documents.mx Internet Source	<1%	