A Project Report On

**Driver drowsiness detection**

Submitted to KL University for the Partial fulfilment of the requirement for the

Award of Degree in

**BACHELOR OF TECHNOLOGY**

# In

**COMPUTER SCIENCE & ENGINEERING**

Submitted by

G. Srihari (190339321)

K. Charan Sai (190330098)

G. Pavan Kalyan (190330275)

Ch. Sushant (190330271)

Under the Esteemed Guidance of

**Mrs. Arpita Gupta Professor (CSE Dept)**

## ABSTRACT

Drowsy Driving is one of the major causes of road accidents and death. A direct way of measuring driver fatigue is measuring the state of the driver i.e., drowsiness. So, it is very important to detect the drowsiness of the driver to save life and property. This project is aimed towards developing a prototype of drowsiness detection system. This system is a real time system which captures image continuously and measures the state of the eye according to the specified algorithm and gives warning if required. Though there are several methods for measuring the drowsiness but this approach is completely non-intrusive which does not affect the driver in any way, hence giving the exact condition of the driver.

In this system, a digital camera continuously records the video capturing driver's face and changes happening over time**.** In this system, a webcam records the video and driver’s face is detected in each frame employing image processing techniques**.** Facial landmarks on the detected face are pointed out using Support Vector Machine (SVM) based pre-trained facial landmark predictor. From this, the eye aspect ratio is computed. Drowsiness is detected from these calculated values. Then SVM is used to check whether detected object is face or non-face. It further monitors the Eye Aspect Ratio (EAR) and Mouth Opening Ratio (MOR) of the driver up to a fixed number of frames to check the sleepiness and yawning. If sleepiness is detected, a warning alert is given to the driver through an alarm.

TABLE OF CONTENTS

PAGE NO.

### LIST OF TABLES i LIST OF FIGURES ii

1. **Introduction 1-4** 
   1. **Motivation 1 1.2 Existing System 2**
   2. **Problem Statement 3**
   3. **Proposed System 3**

**1.1.1 Objectives 4**

1. **LITERATURE REVIEW 5-8**

1. **REQUIREMENTS & DOMAIN INFORMATION 9-20** 
   1. **Requirement Specifications 9** 
      1. **Hardware Requirements 9**

* + 1. **Software Requirements 9**
  1. **Domain Information 10** 
     1. **Python 10**
     2. **Machine Learning 13**
     3. **OpenCV 18**

1. **SYSTEM METHODOLOGY 21-30** 
   1. **Architecture of Proposed System 21**
   2. **Algorithm 26**
   3. **System Design 28** 
      1. **Data Flow Diagram 29**
      2. **UML Diagrams 30**
2. **EXPERIMENTATION & ANALYSIS 30-42** 
   1. **Experimentation 30**
   2. **Results 31** 
      1. **Screen Shots 31**
   3. **Testing 39** 
      1. **Types of Testing 39**
      2. **Test cases 42**

**6.CONCLUSION AND FUTURE SCOPE 43-44**

* 1. **Conclusion 43**
  2. **Future Scope 43**

**REFERENCES 45-46**   i

## LIST OF TABLES

**Table NO Name of Table Page no.**

**5.3.2.1 Testing Result 42**

ii

## LIST OF FIGURES

**FIGURE NO.**  **Name of the Figure**  **Page no.**

### 4.1.1 System Architecture 22 4.1.2 Facial Landmark predictor 24 4.1.3 Eye Aspect Ratio 24 4.1.4 Mouth Opening Ratio 25

**4.3.1.1**  **Data Flow diagram**  **30**  **4.3.2.1**  **Use Case Diagram**  **31**  **4.3.2.2**  **Class Diagram**  **31**  **4.3.2.3**  **Sequence Diagram**  **32**  **4.3.2.4**  **Activity Diagram**  **33**  **5.2.1.1**  **Home Page**  **34**  **5.2.1.2**  **Person is active with spectacles**  **35**  **5.2.1.3**  **Drowsiness Alert**  **3**  **5.2.1.4**

**Yawn Count=1**  **37**  **5.2.1.5**  **Yawn Count=2**  **38**   **5.2.1.6 Person is active 38 5.2.1.7 Person is drowsy 39**

iii

## 1.INTRODUCTION

Drowsy Driving is the major causes of deaths occurring in road accidents. Drowsiness is the state of feeling tired or sleepy. In recent years, not much improvement has been seen in the reduction of road accidents. Amongst the various reasons, the major one is the driver’s fatigue and drowsy state. This reduces driver’s decision-making ability to control the car. Symptoms of being drowsy sleepy include difficulty in focusing, frequent eye blinking, day-dreaming, missing traffic signs, yawning repeatedly etc. Therefore, it is necessary that the next set of cars coming out in the market should have an additional safety feature to alert sleepy driver’s and handling auxiliary task using hand gesture. Such a system can be created using various approaches which may be intrusive or non-intrusive. Hence, analysis of facial expressions which is considered to be the most appropriate method is used. This requires a camera to be placed inside the car for capturing driver’s image. Further processing of the captured image is achieved in which we extract feature descriptor to detect the faces in each frame. Support Vector Machine (SVM) is trained to classify the face and nonface region.

Face recognition feature is included in starting to keep a separate timer for every new driver. Determine the drowsiness from these parameters are Eye blink and Yawning. The aim of this project is to develop a prototype drowsiness detection system. The focus will be placed on designing a system that will accurately monitor the open or closed state of the driver’s eyes in real-time. By monitoring the eyes, it is believed that the symptoms of driver fatigue can be detected early enough to avoid a car accident. Detection of fatigue involves the observation of eye movements and blink patterns in sequence of images of a face. Our objective of the project is to ensure the safety system.

In this manner, a system which can keep a check of driver’s condition for drowsiness and alert the driver before it’s too late. For this we need a system which will focus on the open or closed state of driver’s eyes as by monitoring the state of the eyes detection of drowsiness is easy. Detection in real-time is the major challenge in the field of accident prevention system. The purpose of this study is to provide a real-time monitoring system using video processing, face/eye detection techniques. This system deals with automatic driver drowsiness detection based on visual information. Our system will capture the video through camera and after processing, it will alert the driver based on the results. This system has overcome few of the limitations of the existing systems. Our System will not only alert the driver but also the co-passengers with a loud alarm. Parameters like efficiency, speed and light conditions will also be considered. A low cost, real time driver’s drowsiness detection system is developed with acceptable

accuracy.

### 1.1 MOTIVATION

Monitoring the drivers action while driving by examining the manoeuvred of the vehicle can be a very prominent task in order to enhance safety while driving. To differentiate between unintentional and intentional car steering wheel inputs, will be the main key element to be discovered, such as a sudden large steering input could indicate the driver’s level of alertness. Driver drowsiness is a genuine risk in transportation frameworks. It has been recognized as an immediate or contributing reason for street mishap. Drowsiness can truly slow response time, decline mindfulness and weaken a driver's judgment. It is inferred that driving while drowsy is like driving affected by liquor or medications. In industrialized nations, drowsiness has been assessed to be associated with 2% to 23% everything being equal. Frameworks that distinguish when drivers are turning out to be drowsy and sound an admonition guarantee to be a significant guide in forestalling mishaps. Conceivable procedure for distinguishing drowsiness in drivers can be commonly separated into the accompanying classifications: detecting of physiological qualities, detecting of driver activity, detecting of vehicle reaction, observing the reaction of driver

### 1.2 EXISTING SYSTEM

In existing system, the driver drowsiness detection system involves controlling accident due to unconsciousness through Eye blink. Here one eye blink sensor is fixed in vehicle where if driver loses consciousness, then it alerts the driver through buzzer to prevent vehicle from accident. Causes irritation in the eye, May damage retina Highly expensive and Distract the Driver. The existing system of driver drowsiness detection system has following disadvantages. Mainly, using of two cameras in the system one for monitoring the head movement and the other one for facial expressions. The other disadvantage is aging of sensors and all these sensors are attached to the driver’s body which may affect the driver. So to overcome all these disadvantages we designed a system in which a live camera is used for monitoring the driver drowsiness condition and alert the driver which reduces the road accidents.

### 1.3 PROBLEM STATEMENT

Designing a Drowsiness Detection System which will focus on continuously and accurately monitoring the state of the driver’s eyes in real time to check whether the Eyes are open or closed for more than a given period of time. The proposed work is to alert the driver when the person is drowsy or distracted.

To Design a System that will detect Drowsiness and take necessary steps to avoid accidents. Fatigue is a safety problem that has not yet been deeply tackled by any country in the world mainly because of its nature. Fatigue, in general, is very difficult to measure or observe unlike alcohol and drugs, which have clear key indicators and tests that are available easily. Probably, the best solutions to this problem are awareness about fatigue-related accidents and promoting drivers to admit fatigue when needed.

### 1.4 PROPOSED SYSTEM

In Proposed System, we 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 portable camera will be positioned in front of the driver to capture the front face image. From the Video, the frames are extracted to obtain images. From the facial landmarks, eye aspect ratio and mouth opening ratio are quantified and using these features and machine learning approach, a decision is obtained about the drowsiness of the driver. If drowsiness is detected, an alarm will be sent to the driver to alert him/her.

#### 1.4.1 Objective

The Project focuses on these objectives, which are:

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

## 2.LITERATURE SURVEY

Wei-Liang Ou,Ming-Ho Shih,Chien-Wei Chang,Xue-Han Yu,Chih- Peng Fan [1]An intelligent video-based drowsy driver detection system, which is unaffected by various illuminations, is developed in this study. Even if a driver wears glasses, the proposed system detects the drowsy conditions effectively. By a near- infrared-ray (NIR) camera, the proposed system is divided into two cascaded computational procedures: the driver eyes detection and the drowsy driver detection. The average open/closed eyes detection rates without/with glasses are 94% and 78%, respectively, and the accuracy of the drowsy status detection is up to 91%. By implementing on the FPGA-based embedded platform, the processing speed with the 640×480 format video is up to 16 frames per second (fps) after software optimizations.

W. B. Horng, C. Y. Chen, Y. Chang, C. H. Fan [2] A vision-based real-time driver fatigue detection system is proposed for driving safely. The driver's face is located, from color images captured in a car, by using the characteristic of skin colors. Then, edge detection is used to locate the regions of eyes. In addition to being used as the dynamic templates for eye tracking in the next frame, the obtained eyes' images are also used for fatigue detection in order to generate some warning alarms for driving safety. The system is tested on a Pentium III 550 CPU with 128 MB RAM. The experiment results seem quite encouraging and promising. The system can reach 20 frames per second for eye tracking, and the average correct rate for eye location and tracking can achieve 99.1% on four test videos. The correct rate for fatigue detection is l00%, but the average precision rate is 88.9% on the test videos.

S. Singh, N. P. papa Nikolopoulos [3] It describe a non-intrusive vision-based system for the detection of driver fatigue. The system uses a color video camera that points directly rewards the driver's face and monitors the driver's eyes in order to detect microsleeps (short periods of sleep). The system deals with skin-color information in order to search for the face in the input space. After segmenting the pixels with skin like color, we perform blob processing in order to determine the exact position of the face. We reduce the search space by analyzing the horizontal gradient map of the face, taking into account the knowledge that eye regions in the face present a great change in the horizontal intensity gradient. In order to find and track the location of the pupil, we use gray scale model matching. We also use the same pattern recognition technique to determine whether the eye is open or closed. If the eyes remain closed for an abnormal period of time (5-6 sec), the system draws the conclusion that the person is falling asleep and issues a warning signal.

Belal Alshaqaqi,Abdullah Salem Baquhaizel,Mohamed El Amine Ouis,Meriem Boumehed [4] Drowsiness and Fatigue of drivers are amongst the significant causes of road accidents. Every year, they increase the amounts of deaths and fatalities injuries globally. In this paper, a module for Advanced Driver Assistance System (ADAS) is presented to reduce the number of accidents due to drivers fatigue and hence increase the transportation safety; this system deals with automatic driver drowsiness detection based on visual information and Artificial Intelligence. We propose an algorithm to locate, track, and analyze both the drivers face and eyes to measure PERCLOS, a scientifically supported measure of drowsiness associated with slow eye closure.Vandna Saini, Rekha Saini [5] Driver Drowsiness Detection System and Techniques According to the experts it has been observed that when the drivers do not take break they tend to run a high risk of becoming drowsy. Study shows that accidents occur due to sleepy drivers in need of a rest, which means that road accidents occurs more due to drowsiness rather than drink-driving. Attention assist can warn of inattentiveness and drowsiness in an extended speed range and notify drivers of their current state of fatigue and the driving time since the last break, offers adjustable sensitivity and, if a warning is emitted.

M.J. Flores et. al [6] proposed that to reduce the amount of such fatalities, a module for an advanced driver assistance system, which caters for automatic driver drowsiness detection and also driver distraction, is presented. Artificial intelligence algorithms are used to process the visual information in order to locate, track and analyse both the driver’s face and eyes to compute the drowsiness and distraction indexes. This real time system works during nocturnal conditions as a result of a near-infrared lighting system. Finally, examples of different driver images taken in a real vehicle at night time are shown to validate the proposed algorithms.

M. A. Assari, M. Rahmati [7] proposed a system that used infrared camera, to capture the video stream of the driver, in order to resolve the issues imposed by lightning conditions, presence of glasses, beard etc. Further face region is detected and facial components are extracted from the stream and then compared with that of a drowsy person using template matching.

Zhang, Wei; Cheng, Bo; Lin, Yingzi [8] They presented a nonintrusive drowsiness recognition method using eye-tracking and image processing. A robust eye detection algorithm is introduced to address the problems caused by changes in illumination and driver posture. Six measures are calculated with percentage of eyelid closure, maximum closure duration, blink frequency, average opening level of the eyes, opening velocity of the eyes, and closing velocity of the eyes. These measures are combined using

Fisher’s linear discriminated functions using a stepwise method to reduce the correlations and extract an independent index. Results with six participants in driving simulator experiments demonstrate the feasibility of this video-based drowsiness recognition method that provided 86% accuracy**.**

Raoul Lopes, D.J Sanghvi, Aditya Shah [9] research on the drowsiness detection is classified into three different aspects Vehicle-based, Behavioural based, and Physiological based. The complete view on these methods in all the corners will give the required details, and some changes need to be made to get fruitful results. Chisty, Jasmeen Gill [10] It is a proven fact that one of the main reasons for the traffic accident is driver’s drowsiness. If we can have some technology driven mechanism with which the drivers can be warned and alarmed in the phase I or early phase II itself, before they enter the phase III of deep sleep or transition into drowsiness, a number of these unfortunate incidents could be averted. To reliably sight the drowsiness, a lot is dependent on the real time data presentation and its analysis, thereby providing timely warnings of drowsiness and alert to the drivers of the vehicle. Such detection methods are often found to be limited by their failure to think about individual variations and contexts, the kind of information used etc. Drowsiness detection is often categorized into two categories intrusive and non-intrusive approaches. In one of the study conducted, it was found that using a non-intrusive method such as calculating eye ratios through a camera to detect drowsiness provided the best results.

Vitabile et al. [11] implement a system to detect symptoms of driver drowsiness based on an infrared camera. By exploiting the phenomenon of bright pupils, an algorithm for detecting and tracking the driver's eyes has been developed. When drowsiness is detected, the system warns the driver with an alarm message.

Bin Yang et. al. [12] proposed that measures of the driver’s eyes are capable to detect drowsiness under simulator or experiment conditions. The performance of the latest eye tracking based in-vehicle fatigue prediction measures are evaluated. These measures are assessed statistically and by a classification method based on a large dataset of 90 hours of real road drives. The results show that eye-tracking drowsiness detection works well for some drivers as long as the blinks detection works properly. Even with some proposed improvements, however, there are still problems with bad light conditions and for persons wearing glasses. As a summary, the camera based sleepiness measures provide a valuable contribution for a drowsiness reference, but are not reliable enough to be the only reference.

## 3.REQUIREMENTS & DOMAIN INFORMATION

### 3.1 REQUIREMENT SPECIFICATIONS

This section elaborates on the functional requirements of the application. The SRS itself can be divided into module, each module having specifications. In order to carry out the project, the following hardware and software is required.

#### 3.1.1 HARDWARE REQUIREMENTS

The selection of hardware is very important in the existence and proper working of any software. In the selection of hardware, the size and the capacity are also very important.

‾ **Processor :** Pentium IV or higher ‾ **RAM :** 256 MB  ‾ **Hard Disk :** min 512 MB

#### 3.1.2 SOFTWARE REQUIREMENTS

The software requirements specification is produced at the culmination of the analysis tasks. One of the most difficult tasks is that, the selection of the software, once system requirement is known by determining whether a particular software package fits the requirements.

Operating System: Windows-10

Coding Language : Python Server: Flask

##### Software Requirement Specifications

Software Requirement Specification (SRS) is the starting point of the software developing activity. As system grew more complex it became evident that the goal of the entire system cannot be easily comprehended.

Hence the need for the requirement phase arose. The software project is initiated by the client needs. The SRS is the means of translating the ideas of the minds of clients (the input) into a formal document (the output of the requirement phase).

The SRS phase consists of two basic activities: **1)Problem/Requirement Analysis**  The process is order and more nebulous of the two, deals with understand the Problem, the goal and constraints.

###### 2) Requirement Specification

Here, the focus is on specifying what has been found giving analysis such as representation, specification languages and tools, and checking the specifications are addressed during this activity.

The Requirement phase terminates with the production of the validate SRS document.

Producing the SRS document is the basic goal of this phase.

### ROLE OF SRS

The purpose of the Software Requirement Specification is to reduce the communication gap between the clients and the developers. Software Requirement Specification is the medium though which the client and user needs are accurately specified. It forms the basis of software development. A good SRS should satisfy all the parties involved in the system.

#### 3.2 DOMAIN INFORMATION

##### 3.2.1 Python

Python is a general purpose high level language.It was launched officially in 1991.Python programming language can be used in many applications like Web basedapplications, desktop based applications, games, artificial intelligence, Neural networks and many more. Facebook, Google, Yahoo, Netflix are the users of Python programming language. Features of Python are:

* Simple and easy to learn
* Platform independent
* Dynamically typed
* Both procedure and object oriented
* Embedded • Extensive Library.

##### History of Python

Right, both start with ABC. If we are talking about ABC in the Python context, it's clear that the programming language ABC is meant. ABC is a general-purpose programming language and programming environment, which had been developed in the Netherlands, Amsterdam, at the CWI (Centrum Wiskunde &Informatica). The greatest achievement of ABC was to influence the design of Python. Python was conceptualized in the late 1980s. Guido van Rossum worked that time in a project at the CWI, called Amoeba, a distributed operating system. In an interview with Bill Venners1, Guido van Rossum said: "In the early 1980s, I worked as an implementer on a team building a language called ABC at Centrum voor Wiskundeen Informatica (CWI). I don't know how well people know ABC's influence on Python. I try to mention ABC's influence because I'm indebted to everything I learned during that project and to the people who worked on it."Later on in the same Interview, Guido van Rossum continued: "I remembered all my experience and some of my frustration with ABC. I decided to try to design a simple scripting language that possessed some of ABC's better properties, but without its problems. So I started typing. I created a simple virtual machine, a simple parser, and a simple runtime. I made my own version of the various ABC parts that I liked. I created a basic syntax, used indentation for statement grouping instead of curly braces or beginend blocks, and developed a small number of powerful data types:

a hash table (or dictionary, as we call it), a list, strings, and numbers." **Advantages**  **of Python** :

Let’s see how Python dominates over other languages. **a)**

###### 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. **b) 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. **c)Embedded**

Complimentary to extensibility, Python is embedded 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. **d)Improved Productivity**

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

###### e) 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.

###### f) Simple and Easy

When working with Java, you may have to create a class to print ‘HelloWorld’. But in Python, just a print statement will do. It is also quite easy tolearn, understand,and code.This is why when people pick up Python, they have a hard time adjusting to other more verbose languages like Java. **g) 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. **h) Object-Oriented**

This language supports both the procedural and object- orientedprogramming paradigms. While functions help us with code reusability, classes and objects let us model the real world. A class allows the encapsulation of dataand functions into one. **i) Free and Open-Source**

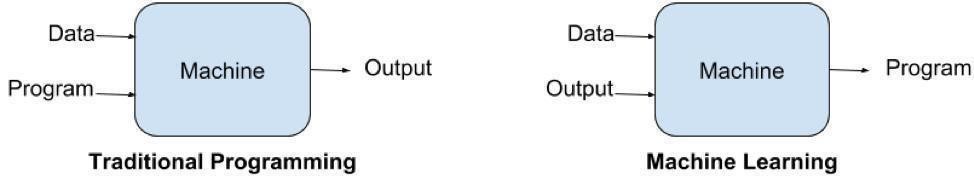
Python is freely available**.** But not only can you download Python for free, but you can also download its source code, make changes to it, and even distribute it. It downloads with an extensive collection of libraries to help you with your tasks. **j) Portable**

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. **k) Interpreted**

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

###### 3.2.2 Machine Learning

Machine learning is the study of computer algorithms that improve automatically through experience. It is seen as a subset of [artificial intelligence. Mach](https://en.wikipedia.org/wiki/Artificial_intelligence)ine learning algorithms buil[d a mathematical model ba](https://en.wikipedia.org/wiki/Mathematical_model)sed on sample data, known [as "training data",](https://en.wikipedia.org/wiki/Training_data) in order to make predictions or decisions without being explicitly programmed to do so. Machine learning algorithms are used in a wide variety of applications, such a[s email](https://en.wikipedia.org/wiki/Email_filtering) filtering an[d computer vision, w](https://en.wikipedia.org/wiki/Computer_vision)here it is difficult or infeasible to develop conventional algorithms to perform the needed tasks.



Few examples of machine learning that we use every day and perhaps have no idea that they are driven by ML.

**Virtual Personal Assistants:**Siri, Alexa, Google Now are some of the popular examples of virtualpersonal assistants. As the name suggests, they assist in finding information, when asked over voice.

**Predictions while Commuting:** We all have been using GPS navigation services. While we do that,our current locations and velocities are being saved at a central server for managing traffic.

**Videos Surveillance:** Thevideo surveillance system nowadays are powered by AI that makes itpossible to detect crime before they happen. They track unusual behavior of people like standing motionless for a long time, stumbling, or napping on benches etc.

Need for Machine Learning

Human beings, at this moment, are the most intelligent and advanced species on earth because they can think, evaluate and solve complex problems. On 19the other side, AI is still in its initial stage and haven’t surpassed human intelligence in many aspects. The most suitable reason for doing this is, “to make decisions, based on data, with efficiency and scale”. Lately, organizations are investing heavily in newer technologies like Artificial Intelligence, Machine Learning and Deep Learning to get the key information from data to perform several real-world tasks and solve problems. We can call it datadriven decisions taken by machines, particularly to automate the process. These datadriven decisions can be used, instead of using programming logic, in the problems that cannot be programmed inherently. The fact is that we can’t do without human intelligence, but other aspect is that we all need to solve real-world problems with efficiency at a huge scale. That is why the need for machine learning arises.

**How ML works?**

Gathering past data in the form of text file, excel file, images or audio data. The better the quality of data, the better will be the model learning. Sometimes, the data collected is in the raw form and it needs to be rectified. Example: if data has some missing values, then it has to be rectified. If data is in the form of text or images then converting it to numerical form will be required, be it list or array or matrix. Simply, Data is to be made relevant and understandable by the machine. Building up models with suitable algorithms and techniques and then training it. Testing our prepared model with data which was not feed in at the time of training and so evaluating the performance – score, accuracy with high level of precision.

Machine learning uses two types of techniques: supervised learning, which trains a model on known input and output data so that it can predict future outputs, and unsupervised learning, which finds hidden patterns or intrinsic structures in input data.

Supervised Learning

[Supervised machine learning buil](https://www.mathworks.com/discovery/supervised-learning.html)ds a model that makes predictions based on evidence in the presence of uncertainty. A supervised learning algorithm takes a known set of input data and known responses to the data (output) and trains a model to generate reasonable predictions for the response to new data. Use supervised learning if you have known data for the output you are trying to predict. Supervised learning uses classification and regression techniques to develop predictive models.

Classification techniques predict discrete responses—for example, whether an email is genuine or spam, or whether a tumor is cancerous or benign. Classification models classify input data into categories.

Typical applications include medical imaging, speech recognition, and credit scoring. Use classification if your data can be tagged, categorized, or separated into specific groups or classes. For example, applications for hand-writing recognition use classification to recognize letters and numbers. In image processing and computer vision, [unsupervised pattern recognition tech](https://www.mathworks.com/discovery/pattern-recognition.html)niques are used for object detection and image segmentation.

Regression techniques predict continuous responses—for example, changes in temperature or fluctuations in power demand. Typical applications include electricity load forecasting and algorithmic trading. Use regression techniques if you are working with a data range or if the nature of your response is a real number, such as temperature or the time until failure for a piece of equipment.

Unsupervised Learning

[Unsupervised learning find](https://www.mathworks.com/discovery/unsupervised-learning.html)s hidden patterns or intrinsic structures in data. It is used to draw inferences from datasets consisting of input data without labeled responses. Clustering is the most common unsupervised learning technique. It is used for exploratory data analysis to find hidden patterns or groupings in data. Applications for [cluster analysis incl](https://www.mathworks.com/discovery/cluster-analysis.html)ude gene sequence analysis, market research, and object recognition. For example, if a cell phone company wants optimize the locations where they build cell phone towers, they can use machine learning to estimate the number of clusters of people relying on their towers. A phone can only talk to one tower at a time, so the team uses clustering algorithms to design the best placement of cell towers to optimize signal reception for groups, or clusters, of their customers.

Reinforcement Learning

Reinforcement learning is the training of machine learning models to making a sequence of decisions. The agent learns to achieve a goal in an uncertain, potentially complex environment. In reinforcement learning, an artificial intelligence faces a gamelike situation. The computer employs trial and error to come up with a solution to the problem. To get the machine to do what the programmer wants, the artificial intelligence gets either rewards or penalties for the actions it performs. Its goal is to maximize the total reward.

Although the set of the reward policy–that is, the rules of the game–he gives the

model no hints or suggestions for how to solve the game. It’s up to the model to figure out how to perform the task to maximize the reward, starting from totally random trials and finishing with sophisticated tactics and superhuman skills. By leveraging the power of search and many trials, reinforcement learning is currently the most effective way to hint machine’s creativity. In contrast to human beings, artificial intelligence can gather experience from thousands of parallel gameplays if a reinforcement learning algorithm is run on a sufficiently powerful computer infrastructure.

Applications of Machines Learning

Machine Learning is the most rapidly growing technology and according to researchers we are in the golden year of AI and ML. It is used to solve many real- world complex problems which cannot be solved with traditional approach.

Following are some real-world applications of ML −

1. Emotion analysis
2. Sentiment analysis
3. Error detection and prevention
4. Weather forecasting and prediction
5. Stock market analysis and forecasting
6. Speech synthesis
7. Speech recognition
8. Customer segmentation
9. Object recognition
10. Fraud detection

11)Fraud prevention

12) Recommendation of products to customer in online shopping

**Advantages of Machine Learning**  **1. Easily identifies trends and patterns**  Machine Learning can review large volumes of data and discover specific trends and patterns that would not be apparent to humans. For instance, for an e-commerce website like Amazon, it serves to understand the browsing behaviours and purchase histories of its users to help cater to the right products, deals, and reminders relevant to them. It uses the results to reveal relevant advertisements to them.

2. No human intervention needed (automation)

With ML, you don’t need to babysit your project every step of the way. Since it means giving machines the ability to learn, it lets them make predictions and also improve the algorithms on their own. A common example of this is anti-virus softwares; they learn to filter new threats as they are recognized. ML is also good at recognizing spam.

3. Continuous Improvement

As ML algorithmsgain experience, they keep improving in accuracy and efficiency.

This lets them make better decisions. Say you need to make a weather forecast model. As the amount of data you have keeps growing, your algorithms learn to make more accurate predictions faster.

4. Handling multi-dimensional and multi-variety data

Machine Learning algorithms are good at handling data that are multi-dimensional and multi-variety, and they can do this in dynamic or uncertain environments.

5. Wide Applications

You could be an e-tailer or a healthcare provider and make ML work for you. Where it does apply, it holds the capability to help deliver a much more personal experience to customers while also targeting the right customers. **Disadvantages of Machine**

Learning 1. Data Acquisition

Machine Learning requires massive data sets to train on, and these should be inclusive/unbiased, and of good quality. There can also be times where they must wait for new data to be generated.

2. Time and Resources

ML needs enough time to let the algorithms learn and develop enough to fulfil their purpose with a considerable amount of accuracy and relevancy. It also needs massive resources to function. This can mean additional requirements of computer power for you.

3. Interpretation of Results

Another major challenge is the ability to accurately interpret results generated by the algorithms. You must also carefully choose the algorithms for your purpose.

4. High error-susceptibility

Machine Learningis autonomous but highly susceptible to errors. Suppose you train an algorithm with data sets small enough to not be inclusive. You end up with biased predictions coming from a biased training set. This leads to irrelevant advertisements being displayed to customers. In the case of ML, such blunders can set off a chain of errors that can go undetected for long periods of time. And when they do get noticed, it takes quite some time to recognize the source of the issue, and even longer to correct it.

##### 3.2.3 OPENCV

OpenCV(Open Source Computer Vision Library) is a [library of programming functions](https://en.wikipedia.org/wiki/Library_(computing)) mainly aimed at real-time [computer vision.](https://en.wikipedia.org/wiki/Computer_vision) Originally developed by [Intel, i](https://en.wikipedia.org/wiki/Intel_Corporation)t was later supported by [Willow Garage t](https://en.wikipedia.org/wiki/Willow_Garage)hen Itseez (which was later acquired by Intel. The library is [cross-platform an](https://en.wikipedia.org/wiki/Cross-platform)d free for use under the [open-sou](https://en.wikipedia.org/wiki/Open-source_software)[rce Apache 2 License.](https://en.wikipedia.org/wiki/Apache_License) Starting with 2011, OpenCV features GPU acceleration for real-time operations. OpenCV (Open-Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code.

The first OpenCV version was 1.0. OpenCV is released under a BSD license and hence it’s free for both academic and commercial use. It has C++, C, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. When OpenCV was designed the main focus was real-time applications for computational efficiency. All things are written in optimized C/C++ to take advantage of multi-core processing. OpenCV was designed for computational efficiency and having a high focus on realtime image detection. OpenCV is coded with optimized C and can take work with multicore processors. If we desire more automatic optimization using Intel architectures

[Intel], you can buy Intel’s Integrated Performance Primitives (IPP) libraries [IPP].

These consist of low-level routines in various algorithmic areas which are optimized.

OpenCV automatically uses the IPP library, at runtime if that library is installed. One of

OpenCV’s goals is to provide a simple-to-use computer vision infrastructure which helps people to build highly sophisticated vision applications fast. The OpenCV library, containing over 500 functions, spans many areas in vision. Because computer vision and machine learning often goes hand-in-hand,

OpenCV also has a complete, general-purpose, Machine Learning Library (MLL). This sub library is focused on statistical pattern recognition and clustering. The MLL is very

useful for the vision functions that are the basis of OpenCV’s usefulness, but is general enough to be used for any machine learning problem.

**Applications of OpenCV** There are lots of applications which are solved using OpenCV, some of them are listed below

* face recognition
* Automated inspection and surveillance
* number of people – count (foot traffic in a mall, etc)
* Vehicle counting on highways along with their speeds
* Interactive art installations
* Anamoly (defect) detection in the manufacturing process (the odd defective products)
* Street view image stitching
* Video/image search and retrieval
* Robot and driver-less car navigation and control
* object recognition
* Medical image analysis
* Movies – 3D structure from motion
* TV Channels advertisement recognition

**OpenCV Functionality**

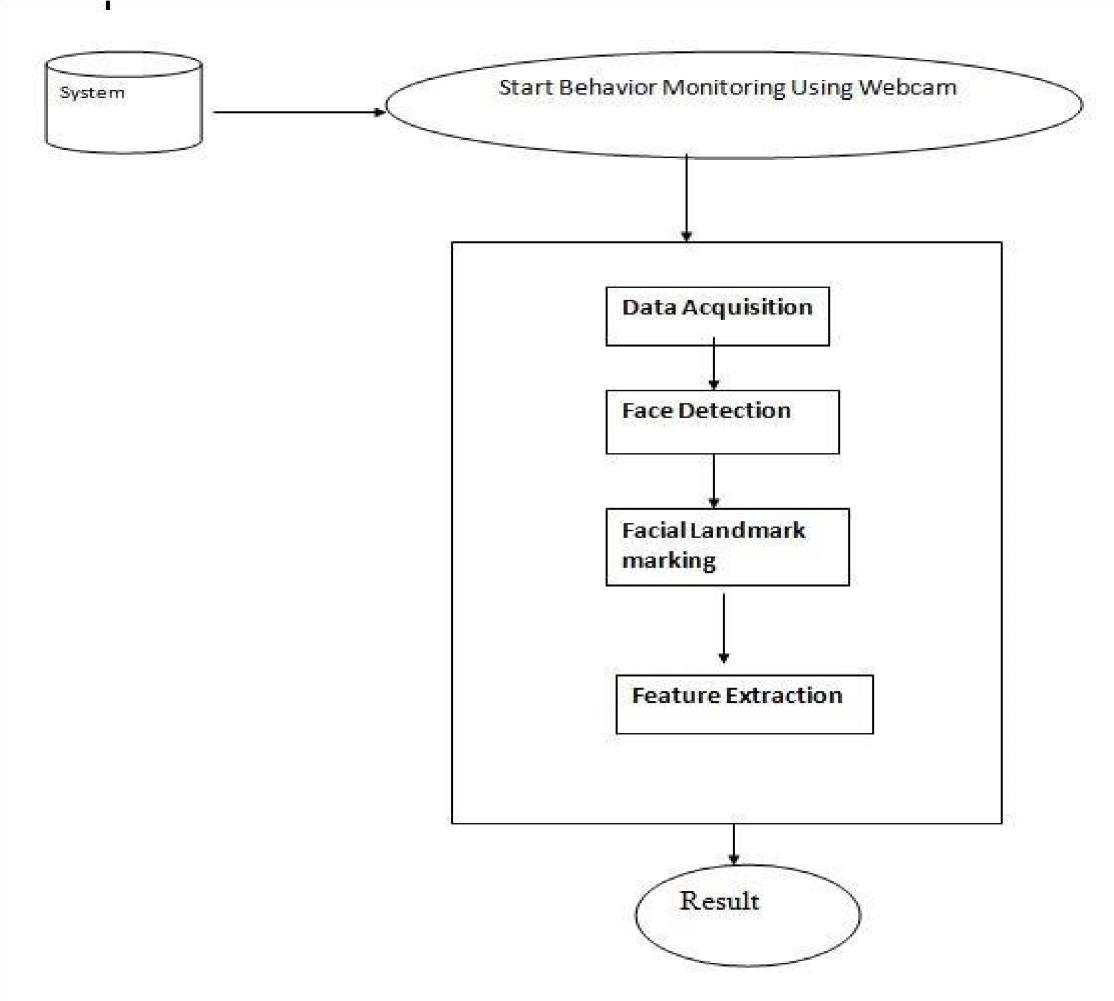
* Image/video I/O, processing, display
* Object/feature detection
* Geometry-based monocular or stereo computer vision
* Computational photography
* Machine learning & clustering
* CUDA acceleration

## 4.SYSTEM METHODLOGY

### 4.1 ARCHITECTURE OF PROPOSED SYSTEM

Architecture diagram is a diagram of a system, in which the principle parts or functions are represented by blocks connected by lines that shows the relationships of the blocks. The block diagram is typically used for a higher level, less detailed description aimed more at understanding the overall concepts and less at understanding the details of the implementation.

At first, the video is recorded using a webcam. The camera will be positioned in front of the driver to capture the front face image. From the video, the frames are extracted to obtain 2-D images. Face is detected in the frames using histogram of oriented gradients (HOG) and linear support vector machine (SVM) for object detection. After detecting the face, facial landmarks like positions of eye, nose, and mouth are marked on the images. From the facial landmarks, eye aspect ratio, mouth opening ratio and position of the head are quantified and using these features and machine learning approach, a decision is obtained about the drowsiness of the driver. If drowsiness is detected, an alarm will be sent to the driver to alert him/her.



**Figure:4.1.1 System Architecture**

Driver Drowsiness Detection System consists following modules

#### a) Data Acquisition

The video is recorded using webcam 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.

##### b) 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 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 SVM, 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 traning 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.

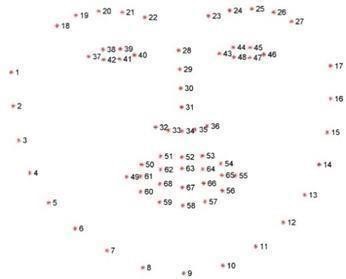
This non-maximum suppression step removes the redundant and overlapping bounding boxes.

##### c) 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 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 are determined. It is a inbuilt HOG SVM classifier used to determine the position of 68(x, y) coordinates that map to facial structures on the face.

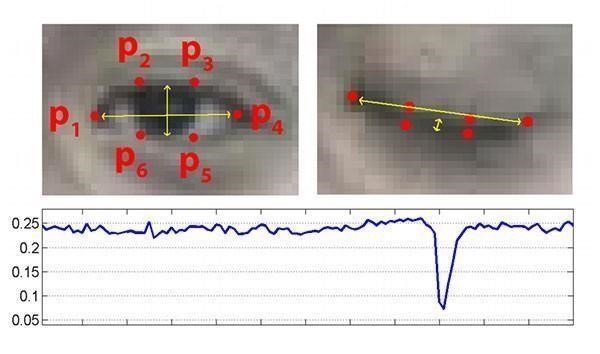
##### d) Feature Extraction

After detecting the facial landmarks, the features are computed as described below.

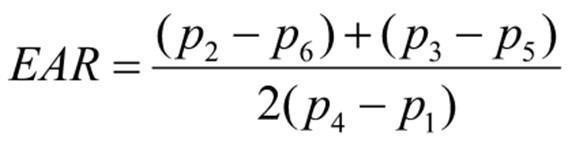


###### Fig:4.1.2. facial landmark predictor

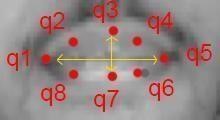
**Eye Aspect Ratio(EAR)** From the eye corner points, the eye aspect ratio is calculated as the ratio of height and width of the eye as given by



###### Fig:4.1.3. Eye Aspect Ratio



**Mouth Opening Ratio(MOR)** Mouth Opening Ratio is a parameter to detect yawning during drowsiness. Similar to EAR, it is calculated as



###### Fig 4.1.4: Mouth Opening Ratio

MOR=|| q2- q8|| + || q4 - q6 || / 2× || q1 - q5||

As defined, it increases rapidly when mouth opens due to yawning and remains at that high value for a while due to yawn (indicating that the mouth is open) and again decreases rapidly towards zero. As yawn is one of the characteristics of drowsiness, MOR gives a measure regarding driver drowsiness.

##### e) 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 4 of 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 is taken experimentally from the observations. If the test value is greater than this threshold then yawn (i.e., drowsiness) is detected.

### 4.2 ALGORITHM

#### 4.2.1 SUPPORT VECTOR MACHINE

“Support Vector Machine” (SVM) is a supervised machine learning algorithm which can be used for both classification and regression challenges. However, it is mostly used in classification problems. In this algorithm, we plot each data item as a point in ndimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiate the two classes very well (look at the below snapshot). Support Vectors are simply the co-ordinates of individual observation. Support Vector Machine is a frontier which best segregates the two classes (hyperplane/ line). More formally, a support vector machine constructs a hyper plane or set of hyper planes in a high- or infinite-dimensional space, which can be used for classification, regression, or other tasks like outliers detection. Intuitively, a good separation is achieved by the hyper plane that has the largest distance to the nearest training-data point of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier. Whereas the original problem may be stated in a finite dimensional space, it often happens that the sets to discriminate are not linearly separable in that space. For this reason, it was proposed that the original finite-dimensional space be mapped into a much higher-dimensional space, presumably making the separation easier in that space.

Machine learning involves predicting and classifying data and to do so we employ various machine learning algorithms according to the dataset. SVM or Support Vector Machine is a linear model for classification and regression problems. It can solve linear and non-linear problems and work well for many practical problems. The idea of SVM is simple: The algorithm creates a line or a hyperplane which separates the data into classes. In machine learning, the radial basis function kernel, or RBF kernel, is a popular kernel function used in various kernelized learning algorithms. In particular, it is commonly used in support vector machine classification. As a simple example, for a classification task with only two features (like the image above), you can think of a hyperplane as a line that linearly separates and classifies a set of data.

Intuitively, the further from the hyperplane our data points lie, the more confident we are that they have been correctly classified. We therefore want our data points to be as

far away from the hyperplane as possible, while still being on the correct side of it. So, when new testing data is added, whatever side of the hyperplane it lands will decide the class that we assign to it.

#### 4.2.2 Histogram Oriented Gradient (HOG)

Histogram Oriented Gradient (HOG) algorithm is used to pre-process the image which includes image resize and colour normalization in this project HOG is used to detect efficient features from for eye detection and extract HOG features from the image patterns and gives the exact region of eyes from the captured image of the driver. The histogram of oriented gradients (HOG) is a feature descriptor used in computer vision and image processing for the purpose of object detection. The technique counts occurrences of gradient orientation in localized portions of an image. This method is similar to that of edge orientation histograms, scale-invariant feature transform descriptors, and shape contexts, but differs in that it is computed on a dense grid of uniformly spaced cells and uses overlapping local contrast normalization for improved accuracy. Image descriptor, Histogram of Oriented Gradient (HOG) along with Linear Support Vector Machine (SVM) is used to set up highly accurate object classifiers.

**4.3 SYSTEM DESIGN**

### 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 and 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.

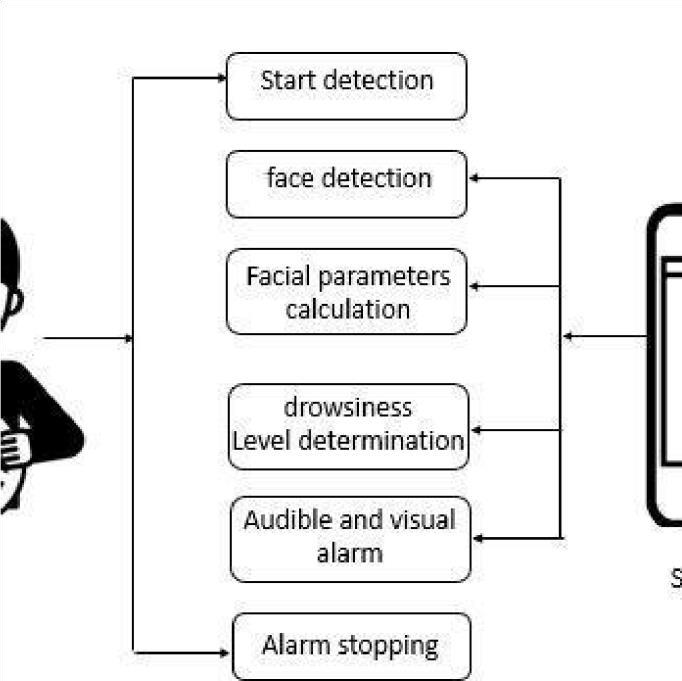
1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the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

#### 4.3.1 Data Flow Diagram

A data flow diagram is graphical tool used to describe and analyze movement of data through a system. These are the central tool and the basis from which the other components are developed. The transformation of data from input to output, through processed, may be described logically and independently of physical components associated with the system. These are known as the logical data flow diagrams. The physical data flow diagrams show the actual implements and movement of data between people, departments and workstations. A full description of a system actually consists of a set of data flow diagrams. Using two familiar notations Yourdon, Gane and Sarson notation develops the data flow diagrams. Each component in a DFD is labeled with a descriptive name. Process is further identified with a number that will be used for identification purpose. The development of DFD’S is done in several levels. Each process in lower level diagrams can be broken down into a more detailed DFD in the next level. The lop-level diagram is often called context diagram. It consists a single process bit, which plays vital role in studying the current system. The process in the context level diagram is exploded into other process at the first level DFD.



**Figure 4.3.1.1 Data Flow Diagram**

#### 4.3.2 UML DIAGRAMS

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 objects.

### USE-CASE DIAGRAM

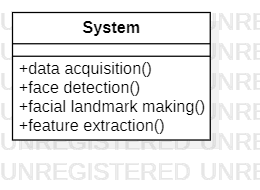
A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses.



**Figure 4.3.2. 1Use-Case Diagram**

### CLASS DIAGRAM

The class diagram is the main building block of object-oriented modelling. It is used for general Conceptual modelling of the structure of the application, and for detailed modelling translating the models into programming code. Class diagrams can also be used for data modelling. The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.



**Figure 4.3.2.2****Class Diagram**

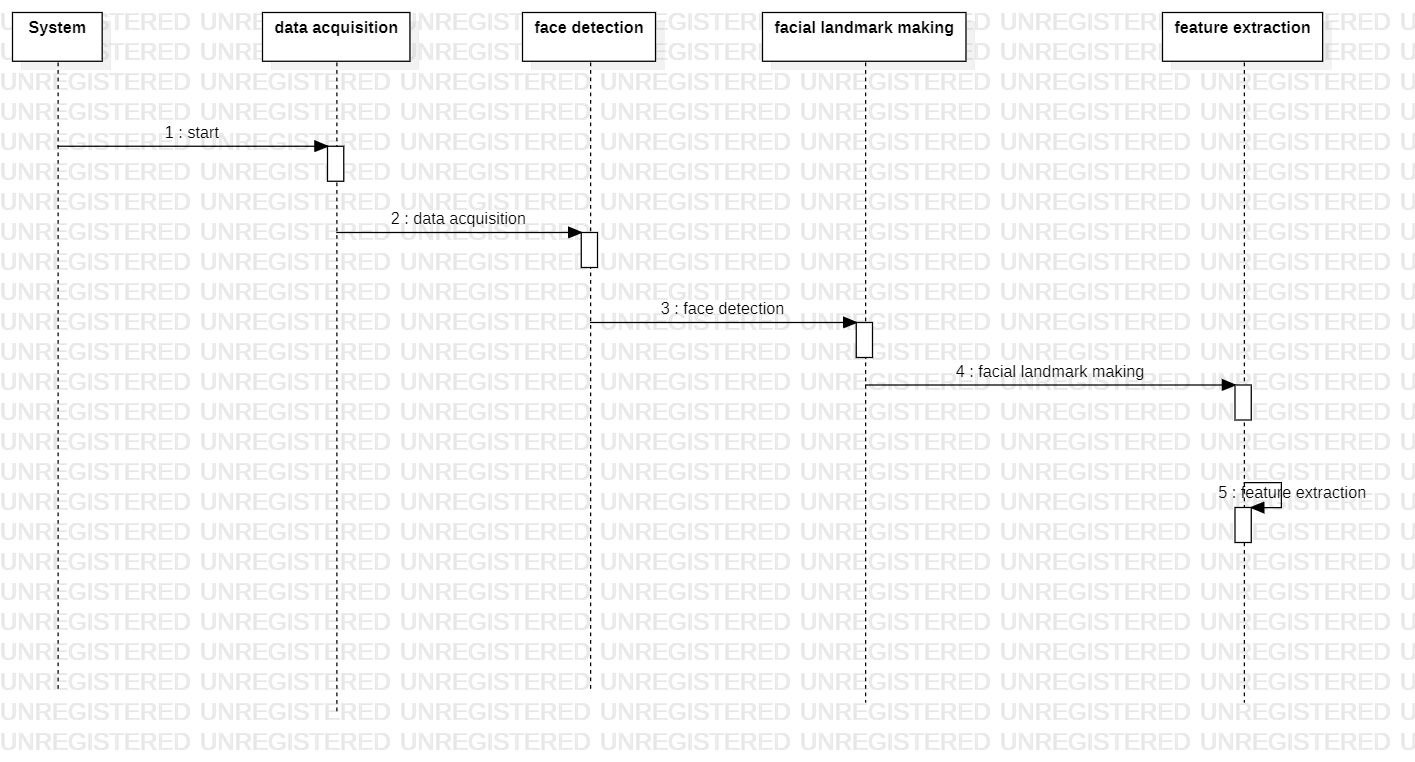
### SEQUENCE DIAGRAM

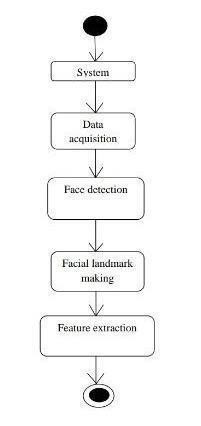
A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the [Logical View of](https://en.wikipedia.org/wiki/4%2B1_architectural_view_model) the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.

**Figure 4.3.2.3 Sequence Diagram**

### 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 are intended to model both computational and organizational processes (i.e., workflows), as well as the data flows intersecting with the related activities. Although activity diagrams primarily show the overall flow of control, they can also include elements showing the flow of data between activities through one or more data stores.





**Figure 4.3.2.6 Activity Diagram**

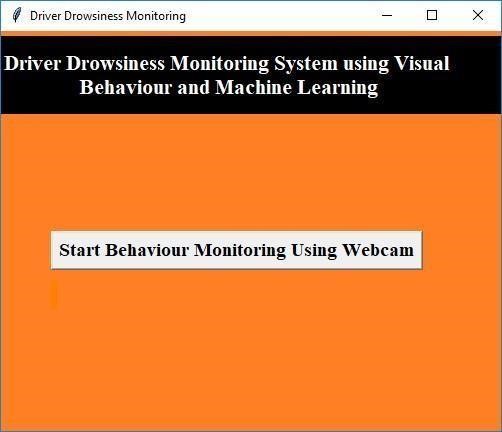
## 5.EXPERIMENTATION &ANALYSIS

### 5.1 Experimentation

In this Project by Monitoring Visual Behaviour of a driver with webcam and machinelearning SVM (support vector machine) algorithm we are detecting Drowsiness in a driver. This application will use inbuilt webcam to read pictures of a driver and then using OPENCV SVM algorithm extract facial features from the picture and then check whether driver in picture is blinking his eyes for consecutive 20 frames or yawning mouth then application will alert driver with Drowsiness messages. We are using SVM pre-trained drowsiness model and then using Euclidean distance function we are continuously checking or predicting EYES and MOUTH distance closer to drowsiness, if distance is closer to drowsiness then application will alert driver.

### 5.2 Results

#### 5.2.1 Screenshots



##### Figure 5.2.1.1 Home page

The above image shows the home page of the project. This is the first page. Here we can click on the start Behaviour Monitoring Using Webcam.

##### Fig 5.2.1.2 Person is Active with spectacles

The person is active and the person eyes are opened and the EAR and MOR are calculated. EAR value is 0.26 and MAR value is 0.47.

**Fig 5.2.1.3 Drowsiness Alert.**

The Drowsiness alert is shown in the above figure and the EAR and MAR are calculated. The EAR and MAR values are calculated. The EAR value is 0.16 and MAR value is 0.45

##### Figure 5.2.1.4.Yawn count=1

In the above figure the yawn count is calculated and the yawn count is 1 and the drowsiness alert is shown.

##### Figure 5.2.1.5. Yawn count=2

In the above figure the yawn count is calculated and the yawn count is 1 and the drowsiness alert is shown.

##### Figure 5.2.1.6 Person is active

The person is active and the person eyes are opened and the EAR and MOR are calculated. EAR value is 0.28 and MAR value is 0.27.

##### Figure 5.2.1.7 Person is Drowsy

The Drowsiness alert is shown in the above figure and the EAR and MAR are calculated. The EAR and MAR values are calculated. The EAR value is 0.21 and MAR value is 0.25

**5.3 TESTING**

### SYSTEM TEST

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, subassemblies, 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.

#### 5.3.1 TYPES OF TESTS

##### 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.

##### 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.

##### 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.

##### SYSTEM TESTING

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.

##### 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.

##### 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.

##### UNIT TESTING

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.

#### 5.3.2 TEST CASES

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.

###### Integration Testing

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.

##### 5.3.2.1 DESIGN OF TEST CASES AND SCENARIOUS

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

|  |  |  |
| --- | --- | --- |
| State | EAR | MOR |
| Normal | 0.35 | 0.34 |
| Yawning | 0.22 | 0.77 |
| Eye Closed | 0.15 | 0.419 |

**Table 5.3.2.1** **Testing Results**

## 6. CONCLUSION AND FUTURE SCOPE

### 6.1 CONCLUSION

A low cost, real time driver drowsiness monitoring system has been proposed based on visual behaviour and machine learning. Here, visual behaviour 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. The driver drowsiness is analysed and driver’s drowsiness is detected and alert system is also designed.

### 6.2 FUTURE SCOPE

We intend to extend the above work by considering following aspects of driving and driver of the vehicle which will further improve the system making it more robust.

* Using ‘k previous frames’ to stabilize the landmark points
* To use the head orientation of the driver as a fatigue monitoring approach
* To observe the stress factor of the person based on the behavioral traits as captured by camera and sensors in the vehicle.

The model can be improved incrementally by using other parameters like blink rate, yawning, state of the car, etc. If all these parameters are used it can improve the accuracy by a lot.

We plan to further work on the project by adding a sensor to track the heart rate in order to prevent accidents caused due to sudden heart attacks to drivers. Same model and techniques can be used for various other uses like Netflix and other streaming services can detect when the user is asleep and stop the video accordingly. It can also be used in application that prevents user from sleeping.

When drowsiness level exceeds a certain limit then a signal is generated which is communicated to the relay through the parallel port(parallel data transfer required for faster results).The relay drives the on delay timer and this timer in turn runs the stepper motor for a definite time period .The stepper motor is connected to a linear actuator.

The linear actuator converts rotational movement of stepper motor to linear motion.

This linear motion is used to drive a shaft which is directly connected to the hydraulic braking system of the vehicle. When the shaft moves it applies the brake and the vehicle speed decreases.

## REFERENCES

1. 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.

1. 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.

1. S. Singh, N. P. papanikolopoulos, “Monitoring Driver Fatigue using Facial

Analysis

Techniques”, IEEE Conference on Intelligent Transportation System, pp 314318.

1. 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.

1. Vandna Saini, Rekha Saini “Driver Drowsiness Detection System and Techniques”, IJCSIT, Vol. 5 (3).
2. M.J. Flores J. Ma Armingol A. de la Escalera, “Driver drowsiness detection system under infrared illumination for an intelligent vehicle” Published in IET Intelligent Transport Systems Received on 13th October 2009 Revised on 1st April 2011. [7] M. A. Assari, M. Rahmati, Driver drowsiness detection using face expression recognition, in: Signal and Image Processing Applications (ICSIPA), 2011 IEEE International Conference on, IEEE, 2011, pp. 337– 341.
3. Zhang, Wei; Cheng, Bo; Lin, Yingzi,” Driver drowsiness recognition based on computer vision technology.” Published in: Tsinghua Science and Technology Date of Publication: June 2012
4. Raoul Lopes, D.J Sanghvi, Aditya Shah, "Drowsiness Detection based on Eye Movement, Yawn Detection and Head Rotation," Vol. 2, No.6,2012.
5. Chisty, Jasmeen Gill, **“**Driver Drowsiness Detection System”, IJCST, Volume 3, Issue 4, Jul-Aug 2015.
6. S. Vitabile, A. Paola and F. Sorbello, "Bright Pupil Detection in an Embedded, Real-time Drowsiness Monitoring System", in 24th IEEE International Conference on Advanced Information Networking and Applications, 2010.
7. Fabian Friedrichs and Bin Yang, “Camera-based Drowsiness Reference for Driver

State Classification under Real Driving Conditions” 2010 IEEE Intelligent Vehicles Symposium University of California, San Diego, CA, USA June 21-24, 2010.