DRIVER DROWSINESS MONITORING SYSTEM USING VISUAL BEHAVIOUR AND MACHINE LEARNING

A PROJECT REPORT

Submitted in partial fulfillment of requirements to

R.V.R & J.C COLLEGE OF ENGINEERING

For the award of the degree

B.Tech in CSE

Submitted By

Batch - 3

V.Jahnavi(Y18CS179)

K.VenkateswaraRao(Y18CS175)

K.Chaitanya(L19CS186)

Under the guidance of

Dr.A. Sri Nagesh

Professor, CSE

April 2022

Department of Computer Science and Engineering

R.V.R & J.C COLLEGE OF ENGINEERING

(Autonomous) (NAAC – ‘A+’ Grade)

Approved by AICTE :: Affiliated to Acharya Nagarjuna University

Chowdavaram, Guntur – 522019, Andhra Pradesh, India

R.V.R. & J.C COLLEGE OF ENGINEERING (Autonomous)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project work (CS-461) titled **"Driver Drowsiness Monitoring System Using Visual Behaviour and Machine Learning"** is the work done by V.Jahnavi (Y18CS179), K.Venkateswara Rao (Y18CS175) and K.Chaitanya (L19CS186), under my supervision, and submitted in partial fulfillment of the requirements for the award of the degree Bachelor of Technology in Computer Science & Engineering, during the academic year 2021-2022.

Dr.A.Sri Nagesh	**Sri.M.Srikanth**	**Dr.M.Sreelatha**
Professor, CSE	Assistant Professor	Prof & HOD, CSE
Project Guide	Project Incharge	

ACKNOWLEDGEMENT

The successful completion of any task would be incomplete without proper guidance, suggestion and environment. Combination of these three factors act like backbone to our Project **"Driver Drowsiness Monitoring System Using Visual Behaviour and Machine Learning"**.

We would like to express our special thanks of gratitude to our guide **Dr.A.Sri Nagesh**, Professor of CSE department who helped us in doing this project successfully.

We are very glad to express our special thanks to our project Incharge **Sri.M.Srikanth**, who extended his encouragement and support to carry out this project successfully.

We express our sincere thanks to **Dr.M.Sreelatha**, Head of the Department of Computer Science and Engineering for her constant encouragement and the support.

We regard our sincere thanks to our principal **Dr.K.Ravindra**, for providing support and stimulating environment.

Finally we submit our sincere thanks to the supporting staff in Department of Computer Science and Engineering.

V.Jahnavi (Y18CS179)
K.Venkateswara Rao (Y18CS175)
K.Chaitanya (L19CS186)

ABSTRACT

Drowsy driving is one of the major causes of road accidents and death. Hence, detection of driver's fatigue and its indication is an active research area. Most of the conventional methods are either vehicle-based, behavioral-based or physiological-based. Few methods are intrusive and distract the driver, some require expensive sensors and data handling. Therefore, in this project, a low-cost, real-time driver's drowsiness detection system has been developed by us with acceptable accuracy. In the developed system, a webcam records the video and the driver's face is detected in each frame employing image processing techniques. Facial landmarks on the detected face are pointed and subsequently the eye aspect ratio (EAR), mouth opening ratio (MOR) and nose length ratio (NLR) are computed and depending on their values, drowsiness is detected based on developed adaptive thresholding.

CONTENTS

Abstract iv

Contents v

List of Figures vii

List of Tables viii

List of Abbreviations ix

1. Introduction 1

1.1 Problem Statement 4

1.2 Objectives 5

1.3 Need for Present study 5

2. Literature Review 7

2.1 Review of the Project 7

2.2 Limitations of Existing Techniques 14

3. System Analysis 20

3.1 Requirement Specification 20

3.2 UML Diagrams 22

4. System Design 35

4.1 Architecture of the Proposed System 35

4.2 Workflow of the Proposed System 36

4.3 Module Description 37

5. Implementation 43

5.1 Algorithms 43

5.2 Data Sources Used 46

6. Testing 47

6.1 Introduction to Testing 47

6.2 Test cases 48

7. Results **50**

7.1 Actual results 50

7.2 Analysis of obtained results 52

8. Conclusion **54**

9. References **55**

LIST OF FIGURES

1	Fig-3.2.1	Use Case Diagram	24
2	Fig-3.2.2	Activity Diagram	26
3	Fig-3.2.3	Sequence Diagram	28
4	Fig-3.2.4	Collaboration Diagram	30
5	Fig-3.2.5	State Chart Diagram	30
6	Fig-3.2.6	Class Diagram	32
7	Fig-3.2.7	Component Diagram	33
8	Fig-3.2.8	Deployment Diagram	34
9	Fig-4.1.1	Architecture	35
10	Fig-4.2.1	Workflow	37
11	Fig-4.3.1	Facial landmark points	39
12	Fig-5.3.1.1	HOG features extracted from open and closed eyes	44
13	Fig-5.3.1.2	HOG features extracted from nose	45
14	Fig-5.3.1.3	HOG features extracted from mouth	45
15	Fig-7.1.1	Frame when driver is not drowsy	51
16	Fig-7.1.2	Frame when driver closes eyes	51
17	Fig-7.1.3	Frame when driver yawn	52
18	Fig-7.1.4	Frame when driver bends head	52
19	Fig-7.2.1	Frame where driver closes eyes for a second but not Continuously	52
20	Fig-7.2.2	Frame where driver bends head for a second but not Continuously	53

LIST OF TABLES

1	Table 4.3.1	Facial landmark points	39
2	Table 6.2.1	Test case 1	48
3	Table 6.2.2	Test case 2	48
4	Table 6.2.3	Test case 3	49
5	Table 6.2.4	Test case 4	49
6	Table 7.2.1	Sample values of different parameters for different states	52

LIST OF ABBREVIATIONS

1	SVM	Support Vector Machine
2	HOG	Histogram of Oriented Gradients
3	FLDA	Fisher’s linear discriminant analysis
4	EAR	Eye Aspect Ratio
5	MOR	Mouth Opening Ratio
6	NLR	Nose Length Ratio

1. INTRODUCTION

Driver drowsiness is one of the major causes of deaths occurring in road accidents. The truck drivers who drive for continuous long hours (especially at night), bus drivers of long distance route or overnight buses are more susceptible to this problem. Driver drowsiness an overcast nightmare to passengers in every country. Every year, a large number of injuries and deaths occur due to fatigue related road accidents. Various studies have suggested that around 20% of all road accidents are fatigue-related, up to 50% on certain roads. Hence, detection of driver's fatigue and its indication is an active area of research due to its immense practical applicability.

The development of drowsiness detection technologies is both an industrial and academic challenge. In the automotive industry, Volvo developed the Driver Alert Control which warns drivers suspected of drowsy driving by using a vehicle-mounted camera connected to its lane departure warning system (LDWS). Following a similar vein, an Attention Assist System has been developed and introduced by Mercedes-Benz that collects data drawn from a driver's driving patterns incessantly ascertains if the obtained information correlates with the steering movement and the driving circumstance at hand. The driver drowsiness detection system, supplied by Bosch, takes decisions based on data derived from the sensor stationed at the steering, the vehicles' driving velocity, turn signal use, and the camera mounted at the front of the car.

Notably, the use of these safety systems which detect drowsiness is not widespread and is uncommon among drivers because they are usually available in luxury vehicles. An increased embedding and connecting of smart devices equipped with sensors and mobile operating systems like Android, which has the largest installed operating system in cars, was shown by surveys in 2015. Inaddition, machine learning has made ground-breaking advances in recent years, especially in the area of deep learning.

Thus, the use of these new technologiesand methodologies can be an effective way to not only increase the efficiencies of the existing real-time driver drowsiness detection system but also provide a tool that can be widely used by drivers.

The basic drowsiness detection system has three blocks/modules; acquisition system, processing system and warning system. Here, the video of the driver's frontal face is captured in acquisition system and transferred to the processing block where it is processed online to detect drowsiness. If drowsiness is detected, a warning or alarm is sent to the driver from the warning system.

Generally, the methods to detect drowsy drivers are classified in threetypes; vehicle based, behavioral based and physiological based. In vehicle based method, a number of metrics like steering wheel movement, accelerator or brake pattern, vehicle speed, lateral acceleration, deviations from lane position etc. are monitored continuously. Detection of any abnormal change in these values is considered as driver drowsiness. This is a non-intrusive measurement as the sensors are not attached on the driver.

In behavioral based method, the visual behavior of driver i.e.., hand movement, leg movement are analyzed to detect drowsiness. This is also a non-intrusive measurement as the sensors are not attached on the driver.

In physiological based method, the physiological signals like Electrocardiogram (ECG), Electooculogram (EOG), Electroencephalogram (EEG), heartbeat, pulse rate etc. are monitored and from these metrics, drowsiness or fatigue level is detected. This is intrusive measurement as the sensors are attached on the driver which will distract the driver. Depending on the sensors used in the system, system cost as well assize will increase. However, inclusion of more parameters/features will increase the accuracy of the system to a certain extent.

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

People in fatigue show some visual behaviors easily observable from changes in their facial features like eyes, head, mouth and face. Computer vision can be a natural and non-intrusive technique to monitor driver's vigilance. Facesas the primary part of human communication have been a research target incomputer vision for a long time. The driver fatigue detection is considered as one of the most prospective commercial applications of automatic facial expression recognition. Automatic recognition (or analysis) of facial expression consists of three levels of tasks: face detection, facial expression information extraction, and expression classification. In these tasks, the information extraction is the main issue for the feature based facial expression recognition from an image sequence. It involves detection, identification and tracking facial feature points under different illuminations, face orientations and facial expressions.

Some common assumptions in previous face related works were: frontal facial views, constant illumination, and the fixed lighting source. Unfortunately these assumptions are not realistic. In the application of real world facial expression understanding, one has to consider at least three issues: capturing the full features in a variety of lighting conditions and head motion, multiple and non rigid object tracking, and the self-occlusion of features.

The process of falling asleep at the wheel can be characterized by a gradual decline in alertness from a normal state due to monotonous driving conditions or other environmental factors; this diminished alertness leads to a state of fuzzy consciousness followed by the onset of fatigue. The critical issue that a fatigue detection system must address is the question of how to accurately and early detect fatigue at the initial stage.

Possible non intrusive techniques for detecting fatigue in drivers using computer vision are

- Methods based on eye and eyelid movements
- Methods based on head movement
- Methods based on mouth opening

The system uses Histogram Oriented Gradient (HOG) feature descriptor for face detection and facial points recognition. Then SVM is used to checkwhether detected object is face or non-face. It further monitors the Eye Aspect Ratio (EAR), Mouth Opening Ratio (MOR) and Nose Length Ratio (NLR) of the driver up to a fixed number of frames to check the sleepiness, yawning and head bending.

1.1 - PROBLEM STATEMENT

The main goal of this project is to develop a non-intrusive system for vehicles that can find the drivers tiredness and concern a warning with time. Because there are a great number of traffic accidents due to fatigue of the drivers, this system aspires to avoid many crashes on roads, thus saving money and minimizing personal suffering. The developed system continually monitors the driver's mouth, eye and head through the real-time camera which is focused at the driver's face. The changes in mouth and eyes are analyzed and then processed to find the tiredness of the drivers and also to send alarm. This approach is simple and less complex as no training is required compared to existing approaches. Three possible cases such as eye closure, yawing, and head tilt are considered for fatigue detection of the driver. Therefore, this approach helps to anticipate the fatigue of the driver and also gives a warning output in the form of alarm.

1.2 - OBJECTIVE

The objective is to build a detection system that identifies key attributes of drowsiness and triggers an alert when someone is drowsy before it is too late. Inthis project by monitoring visual behavior of a driver with webcam and machine learning SVM (support vector machine) algorithm we are detecting Drowsiness in a driver. This application will use inbuilt webcam to read images of a driver and then use OPENCV SVM algorithm extract facial features from the image and then check whether driver in image is blinking his eyes or yawning mouth or bending head for consecutive 10 frames and if this behavior is observed then application will send an alert to driver. We are using SVM pre-trained drowsiness model and using Euclidean distance function we are continuously predicting Eye Aspect Ratio, Mouth Opening Ratio and Nose Length Ratio. If these are closer to values of drowsy driver, then application will alert driver.

1.3 – NEED FOR STUDY

The developed system continually monitors the driver's mouth, eye, and head through the real-time camera which is focused at the driver's face. The changesin mouth, nose and eyes are analyzed and then processed to find the tiredness of the drivers. When tiredness is recognized, the system delivers a warning in the form of alarm. The purpose of these systems is to increase the safety of people by using detection and security alarm to prevent traffic incidents triggered by the drowsiness of the drivers. The need for this study is increasing rapidly. Withthe rapidly increasing number of deaths which occur due to fatigue related road accidents the need for advanced technologies to anticipate the fatigue of the driver with more accuracy within a short time is increasing. This study is assumed to solve most of the problems regarding accuracy as this study is observed to present more accurate results than the previously existing systems.

2. LITERATURE REVIEW

The basic drowsiness detection system operates on three blocks / modules namely acquisition system, processing system and warning system. In the acquisition system, the video of the driver's frontal face is captured. The changes in mouth, nose and eyes are analyzed and then processed to find the tiredness of the drivers in the processing system. Using SVM pre-trained drowsiness model and using Euclidean distance function we are continuously predicting Eye Aspect Ratio (EAR), Mouth Opening Ratio (MOR), Nose Length Ratio (NLR). If these are closer to values of drowsy driver, then application will send a warning or alarm to the driver from warning system. This approach is simple and less complex as no training is required compared to existing approaches. Three possible cases such as eye closure, yawing, and head tilt are considered for fatigue detection of the driver. Therefore, this approach helps to anticipate the fatigue of the driver and also gives a warning output in the form of alarm.

2.1 - REVIEW OF THE PROJECT

Some protocols developed previously will be discussed in this section. A significant amount of work has been done over the last couple of years on image processing technique and deep learning for object detection purpose. Several different recognition and detection algorithms for driver drowsiness have evolved in this field. We can see different current techniques occurring from literature review.

A Multimodal System for Assessing Alertness Levels Due to Cognitive Loading (https://doi.org/10.1109/TNSRE.2017.2672080)

A scheme for assessing the alertness levels of an individual using simultaneous acquisition of multimodal physiological signals and fusing the information into a single metric for quantification of alertness. The system takes electroencephalogram, high-speed image sequence, and speech data as inputs.Certain parameters are computed from each of these measures as indicators of alertness and a metric is proposed using a fusion of the parameters for indicating alertness level of an individual at an instant. The scheme has been validated experimentally using standard neuropsychological tests, such as the Visual Response Test (VRT), Auditory Response Test (ART), a Letter Counting (LC) task, and the Stroop Test. The tests are used both as cognitive tasks to induce mental fatigue as well as tools to gauge the present degree of alertness of the subject. Correlation between the measures has been studied and the experimental variables have been statistically analyzed using measures such as multivariate linear regression and analysis of variance. Correspondence of trends obtained from biomarkers and neuropsychological measures validate the usability of the proposed metric.

A Real-time Driving Drowsiness Detection Algorithm with Individual Differences Consideration (https://ieeexplore.ieee.org/document/8930504)

The research work about driving drowsiness detection algorithm has great significance to improve traffic safety. Presently, there are many fruits and literature about driving drowsiness detection method. However, most of them are devoted to find a universal drowsiness detection method, while ignore the individual driver differences. This paper proposes a real-time driving drowsiness detection algorithm that considers the individual differences of driver. A deep cascaded convolutional neural network was constructed to detect the face region, which avoids the problem of poor accuracy caused by artificial feature extraction. Based on the dlib toolkit, the landmarks of frontal driver facial in a frame are found. According to the eyes landmarks, a new parameter, called Eyes Aspect Ratio, is introduced to evaluate the drowsiness of driver in the current frame. Taking into account differences in size of driver's eyes, the proposed algorithm consists of two modules: offline training and online monitoring. In the first module, a unique fatigue state classifier, based on Support Vector Machines, was trained which taking the Eyes Aspect Ratio as input. Then, in the second module, the trained classifier is application to monitor the state of driver online. Because the fatigue driving state is gradually produced, a variable which calculated by number of drowsy frames per unit time is introduced to assess the drowsiness of driver. Through comparative experiments, we demonstrate this algorithm outperforms current driving drowsiness detection approaches in both accuracy and speed.

In simulated driving applications, the proposed algorithm detects the drowsy state of driver quickly from 640 * 480 resolution images at over 20fps and 94.80% accuracy. The research result can serve intelligent transportation system, ensure driver safety and reduce the losses caused by drowsy driving.

A review on driver drowsiness based on image, bio-signal, and driver behavior (https://ieeexplore.ieee.org/document/8011855)

The ratio of accidents caused by drowsiness, increases slightly year by year. The most victims of this case are young adult and mostly happens in developed country. Therefore, to reduce the number of accidents caused by drowsiness, researchers around the world develop some methods for detecting drowsiness on driver's face automatically. They propose various features such as visual, non- visual, and vehicular. Visual features are extracted from driver's face and recorded by camera. Non-visual features are signals emerged from driver's body and to acquire those signals, they use special sensor attached to driver's body. Vehicular features are obtained by observing the behavior of driver during driving. From those features which are proposed by researchers, we discussed 3 ideas that can be considered as guidance to lead researcher in developing drowsiness detection. The first idea is creating the dataset of drowsiness facial expression because it can predict drowsiness and fatigue. Second idea is to combine visual, non-visual, and vehicular features into one for better detection. And last one is developing wearable hardware such as smart-watch for drowsiness detection which are easy to use and user friendly.

An Accurate ECG-Based Transportation Safety Drowsiness Detection Scheme (https://doi.org/10.1109/TII.2016.2573259)

Many traffic injuries and deaths are caused by the drowsiness of drivers during driving. Existing drowsiness detection schemes are not accurate due to various reasons. To resolve this problem, an accurate driver drowsiness classifier (DDC) has been developed using an electrocardiogram genetic algorithm-based support vector machine (ECG GA-SVM).

In existing studies, a cross correlation kerneland a convolution kernel have both been applied for performing the classification. The DDC is designed by a Mercer kernel KDDC formed by commuting the cross correlation kernel K xcorr,ij and the convolution kernel K conv,ij . K xcorr,ij , and captures the symmetric information among ECG signals from different classes, while Kconv,ij captures the anti-symmetric information among ECG signals from the same class. The final KDDC (a pre-computed kernel) is obtained by a genetic mutation using a multi-objective genetic algorithm. This renders an optimal KDDC that confidently serves as the full descriptor of the drowsiness. The performance of KDDC is compared with the most prevailing kernels. The obtained DDC yields an overall accuracy of 97.01%, sensitivity of 97.16%, and specificity of 96.86%. The analysis reveals that the accuracy of KDDC is better than those of both K xcorr,ij and K conv,ij by more than 11%, and typical kernels including linear, quadratic, third order polynomial, and Gaussian radial basis function by 17-63%, respectively. Comparing with related works using the image-based method and the biometric signal-based method, KDDC improves the accuracy by 48.4- 87.2%. Testing results showed that KDDC has a less than 1% deviation from simulated results. Also, the average delay of DDC was bounded by 0.55 ms. This renders the real time implementation. Thus, the developed ECG GA-SVM provides an accurate and instantaneous warning to the drivers before they fall into sleep. As a result this ensures the public transport safety.

Bright Pupil Detection in an Embedded, Real-Time Drowsiness Monitoring System (https://doi.org/10.1109/AINA.2010.151)

Driver's drowsiness is stated as an important cause of road and highway accidents. Therefore, the development of a system for monitoring the driver's level of fatigue is desirable in order to prevent accidents. The paper presents the design and the implementation of a system able to find and evidence the drowsiness level of a driver in an ordinary motor vehicle, in order to prevent car accidents. The system, made up of a car installed infrared video camera connected to the Celoxica RC203E FPGA based board, is able to perform a real time video stream processing. The system exploits the “bright pupil” phenomenon produced by the retina, that reflects the 90% of the incident light when a radiation of 850 nm wavelength hit the retina itself. While acquiring the video, a processing chain is executed to detect driver's eyes and to compute a PERCLOS (Percentage of Eye Closure) function linked to the drowsiness level of a driver. The achieved experimental results show that an entire 720*576 frame processing requires only16.7 ms, so that the system is able to perform real-time PAL video stream processing and has the potentiality to process 60 frames/sec. The effectiveness of the proposed drowsiness detection system has been successfully tested with a human subject in real operating condition, tracking driver's eyes and detecting drowsiness failures.

Camera-based Driver Drowsiness State Classification Using Logistic Regression Models (https://ieeexplore.ieee.org/document/9282918)

Drowsiness at the wheel is a major problem for traffic road safety. A drowsydriver suffers from decreased vigilance, increased reaction time and degraded decision-making ability, all of which have a huge impact on the driving performance. A driver monitoring system that warns the driver of his or her critical drowsiness state is a worthwhile contribution to traffic road safety. A drowsy driver typically exhibits some observable behaviors, such as eye blinking and head movements, that can be tracked using a camera. In this study, we analyze the potential of eye closure and head rotation signals, provided by a driver camera, to classify the driver's drowsiness state using logistic regression models. This analysis is based on a large dataset collected from 71 subjects in driving simulator experiments. A reliable and independent reference for drowsiness, however, is required in order to perform this analysis. For this purpose, we devise a methodology that merges several drowsiness monitoring approaches to construct a reliable reference for drowsiness. Furthermore, we describe our approach to extract eye blink and head rotation features. Ultimately, we design logistic regression classifiers and combine them using the one-vs-one binarization technique. Our approach achieves a global balanced validation accuracy of 72.7% on a three-class classification problem (awake, questionable and drowsy) by adopting a strict and rigorous evaluation scheme (i.e., leave-one-drive-out cross- validation).

Combined EEG-Gyroscopet DCS Brain Machine Interface System for Early Management of Driver Drowsiness(https://ieeexplore.ieee.org/document/8080143)

Here the design and implementation of a wireless, wearable brain machine interface (BMI) system dedicated to signal sensing and processing for driverdrowsiness detection (DDD) is presented. Owing to the importance of driverdrowsiness and the possibility for brainwaves-based DDD, many electroencephalogram (EEG)-based approaches have been proposed. However,few studies focus on the early detection of driver drowsiness and on the early management of driver drowsiness using a closed-loop algorithm. The reported wireless and wearable BMI system is used for 1) simultaneous EEG and gyroscope-based head movement measurement for the early detection of driver drowsiness and 2) simultaneous EEG and transcranial direct current stimulation (tDCS) for the early management of driver drowsiness. To achieve the purposes ofeasy-to-use and distraction-free driving, a Bluetooth low-energy module is embedded in this BMI system and used to communicate with a fully wearable consumer device, a smartwatch, which coordinates the work of drowsiness monitoring and brain stimulation with its embedded closed-loop algorithm. The proposed system offers a 128 Hz sampling rate per channel, 12-bit and 16-bit resolution for a single-channel EEG and a three-channel gyroscope, and a maximum 2 mA current for the tDCS. The current consumption of the whole headset system is 56 mA. The battery life of the smartwatch is 9 h. The DDD experimental results show that the proposed system obtained a 93.67% five-level overall accuracy, a 96.15% two-level (alert versus slightly drowsy) accuracy, and maximum 16- to 23-min wakefulness maintenance.

Detecting driver drowsiness based on single electro-encephalography channel (https://doi.org/10.1109/SSD.2016.7473671)

In the recent years, driver drowsiness has been considered one of the major causes of road accidents, which can lead to severe physical injuries, deaths and important economic losses. As a consequence, a reliable driver drowsiness-detection-system is necessary to alert the driver before an accident happens. For this reason, an Electroencephalogram (EEG) has recently drawn attention in the field of brain- computer interface and cognitive neuroscience to control and predict the human drowsiness state.

Our objective in this work, is to proposed an automatic approach to detect the occurrence of driver drowsiness onset based on the Artificial Neuronal Network (ANN) and using only one EEG channel. In this study, an experiment has been conducted on ten human subjects using nine features computed from one EEG channel using the Fast Fourier Transform (FFT). After introducing these features in an ANN classifier, we have obtained a classification accuracy rate of 86.1% and 84.3% of drowsiness and alertness detection. All features used in this work are easy to calculate and can be determined in real time, which makes this approach adapted for embedded implementation.

Detecting Driver Drowsiness Using Wireless Wearables

The National Highway Traffic Safety Administration data show that drowsy driving causes more than 100,000 crashes a year. In order to prevent these devastating accidents, it is necessary to build a reliable driver drowsiness detectionsystem which could alert the driver before a mishap happens. In the literature, the drowsiness of a driver can be measured by vehicle-based, behavior-based, and physiology-based approaches. Comparing with the vehicle-based and behavior- based measurements, the physiological measurement of drowsiness is more accurate. With the latest release of wireless wearable devices such as biosensors that can measure people's physiological data, we aim to explore the possibility of designing a user-friendly and accurate driver drowsiness detection system using wireless wearables. In this paper, we use a wearable biosensor called Bio Harness3 produced by Zephyr Technology to measure a driver's physiological data. We present our overall design idea of the driver drowsiness detection system and the preliminary experimental results using the biosensor. The detection system will be designed in two phases: The main task of the first phase is to collect a driver's physiological data by the biosensor and analyze the measured data to find the key parameters related to the drowsiness. In the second phase, we will design a drowsiness detection algorithm and develop a mobile app to alert drowsy drivers. The results from this project can lead to the development of real products which can save many lives and avoid many accidents on the road. Furthermore, our results can be widely applied to any situation where people should not fall asleep: from the applications in mission-critical fields to the applications in everyday life.

Detection and classification of eye state in IR camera for driver drowsiness identification (https://doi.org/10.1109/ICSIPA.2009.5478674)

An eye detection and eye state (open/close) classification methodology for driver drowsiness identification using IR camera has been presented in this paper. In this proposed methodology, thresholding is used to extract face region. Eye localization is done by locating facial landmarks such as eyebrow and possible face center. Morphological operation and K-means is used for accurate eye segmentation. Experiment shows that the proposed methodology gives excellent segmentation results for both open eyes (both bright and dark pupil) and closed eyes and also classifies correctly.

Histograms of oriented gradients for human detection

After reviewing existing edge and gradient based descriptors, we show experimentally that grids of histograms of oriented gradient (HOG) descriptors significantly outperform existing feature sets for human detection. We study the influence of each stage of the computation on performance, concluding that fine- scale gradients, fine orientation binning, relatively coarse spatial binning, and high-quality local contrast normalization in overlapping descriptor blocks are all important for good results. The new approach gives near-perfect separation on the original MIT pedestrian database, so we introduce a more challenging dataset containing over 1800 annotated human images with a large range of pose variations and backgrounds.

Vision based system for driver drowsiness detection

Drowsiness of drivers is amongst the significant causes of road accidents. Every year, it increases 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 driver's fatigue and hence increase the transportation safety; this system deals with automatic driver drowsiness detection based on visual information and Artificial Intelligence. We proposed 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.

2.2 - LIMITATIONS OF EXISTING TECHNIQUES

A survey of existing works was undertaken in a four- folded manner namely Subjective measures, Vehicle-based measures, Behavioural measures, Physiological measures.

A. SUBJECTIVE MEASURES

Subjective measures that evaluate the level of drowsiness are based on the driver's personal estimation and many tools have been used to translate this rating to a measure of driver drowsiness. The most commonly used drowsiness scale is the Karolinska Sleepiness Scale (KSS), a nine-point scale that has verbal anchors for each step. Hu *et al.* measured the KSS ratings of drivers every 5 min and used it as a reference to the EoG signal collected. Portouli *et al.* evaluated EEG data by confirming driver drowsiness through both a questionnaire and a licensed medical practitioner. Some researchers compared the self-determined KSS, which was recorded every 2 min during the driving task, with the variation of lane position (VLP) and found that these measures were not in agreement.Ingre *et al.* determined a relationship between the eye blink duration and the KSS collected every 5 min during the driving task.

Researchers have determined that major lane departures, high eye blink duration and drowsiness-related physiological signals are prevalent for KSS ratings between 5 and 9. However, the subjective rating does not fully coincide with vehicle-based, physiological and behavioral measures.

Because the level of drowsiness is measured approximately every 5 min, sudden variations cannot be detected using subjective measures. Another limitation to using subjective ratings is that the self-introspection alerts the driver, thereby reducing their drowsiness level. In addition, it is difficult to obtain drowsiness feedback from a driver in a real driving situation. Therefore, while subjective ratings are useful in determining drowsiness in a simulated environment,the remaining measures may be better suited for the detection of drowsiness in a real environment.

B. VEHICLE - BASED MEASURES

Another method to measure driver drowsiness involves vehicle-based measurements. In most cases, these measurements are determined in a simulated environment by placing sensors on various vehicle components, including the steering wheel and the acceleration pedal; the signals sent by the sensors are then analyzed to determine the level of drowsiness. Liu et al. published a review on current vehicle-based measures. Some researchers found that sleep deprivation canresult in a larger variability in the driving speed. However, the two most commonly used vehicle-based measures are the steering wheel movement and the standard deviation of lane position.

Steering Wheel Movement (SWM) is measured using steering angle sensor and it is a widely used vehicle-based measure for detecting the level of driver drowsiness. Using an angle sensor mounted on the steering column, the driver's steering behavior is measured. When drowsy, the number of micro- corrections on the steering wheel reduces compared to normal driving. Fairclough and Graham found that sleep deprived drivers made fewer steering wheel reversalsthan normal drivers. To eliminate the effect of lane changes, the researchersconsidered only small steering wheel movements (between 0.5° and 5°), which are needed to adjust the lateral position within the lane. Hence, based on small SWMs,it is possible to determine the drowsiness state of the driver and thus provide an alert if needed. In a simulated environment, light side winds that pushed the car to the right side of the road were added along a curved road in order to create variations in the lateral position and force the drivers to make corrective SWMs. Car companies, such as Nissan and Renault, have adopted SWMs but it works in very limited situations. This is because they can function reliably only at particular environments and are too dependent on the geometric characteristics of the road and to a lesser extent on the kinetic characteristics of the vehicle.

Standard Deviation of Lane Position (SDLP) is another measure through which the level of driver drowsiness can be evaluated. In a simulated environment, the software itself gives the SDLP and in case of field experiments the position of lane is tracked using an external camera.

Ingre et al. conducted an experiment to derive numerical statistics based on SDLP and found that, as KSS ratings increased, SDLP (meters) also increased. For example, KSS ratings of 1, 5,8, and 9 corresponded to SDLP measurements of 0.19, 0.26, 0.36 and 0.47, respectively.

The SDLP was calculated based on the average of 20 participants; however, with some drivers, the SDLP did not exceeded 0.25 m even for a KSS rating of 9. In the experiment by performing correlation analysis on a subject to subject basis significant difference is noted. Another limitation of SDLP is that it is purely dependent on external factors like road marking, climatic and lighting conditions. In summary, many studies have determined that vehicle-based measures are a poor predictor of performance error risk due to drowsiness. Moreover, vehicular-based metrics are not specific to drowsiness. SDLP can also be caused by any type of impaired driving, including driving under the influence of alcohol or other drugs, especially depressants.

C. BEHAVIORAL MEASURES

A drowsy person displays a number of characteristic facial movements, including rapid and constant blinking, nodding or swinging their head, and frequent yawning. Computerized, non-intrusive, behavioral approaches are widely used for determining the drowsiness level of drivers by measuring their abnormal behaviors. Most of the published studies on using behavioral approaches to determine drowsiness, focus on blinking. PERCLOS (which is the percentage of eyelid closure over the pupil over time, reflecting slow eyelid closures, or “droops”, rather than blinks) has been analyzed in many studies. This measurement has been found to be a reliable measure to predict drowsiness and has been used in commercial products such as Seeing Machines and Lexus. Some researchers used multiple facial actions, including inner brow rise, outer brow rise,lip stretch, jaw drop and eye blink, to detect drowsiness.

The main limitation of using a vision-based approach is lighting. Normal cameras do not perform well at night. In order to overcome this limitation, some researchers have used active illumination utilizing an infrared Light Emitting Diode (LED). However, although these work fairly well at night, LEDs are considered less robust during the day.

. In addition, most of the methods have been tested on data obtained from drivers mimicking drowsy behavior rather than onreal video data in which the driver gets naturally drowsy. Mostly, image is acquired using simple CCD or web camera during day and IR camera during night at around 30 fps. After capturing the video, some techniques, including Connected Component Analysis, Cascade of Classifiers or Hough Transform, Gabor Filter, Haar Algorithm are applied to detect the face, eye or mouth. After localizing the specific region of interest within the image, features such as PERCLOS, yawning frequency and head angle, are extracted using an efficient feature extraction technique, such as Wavelet Decomposition, Gabor Wavelets, Discrete Wavelet Transform or Condensation Algorithm. The behavior is then analyzed and classified as either normal, slightly drowsy, highly drowsy through the use of classification methods such as support vector machine, fuzzy classifier, neural classifier and linear discriminant analysis.

However, it has been found that the rate of detecting the correct feature, or the percentage of success among a number of detection attempts, varies depending on the application and number of classes. Thedetermination of drowsiness using PERCLOS and Eye Blink has a success rate of close to 100% and 98%, respectively. However, it has to be noted that, the high positive detection rate achieved by was when the subjects didn't wear glasses. Likewise, as most researchers conducted their experiments in simulated environment they achieved a higher success rate. The positive detection rate decreased significantly when the experiment was carried out in a real environment.

Another limitation of behavioral measure was brought out in an experiment conducted by Golz et al. They evaluated various drowsiness monitoring commercial products, and observed that driver state cannot be correlated to driving performance and vehicle status based on behavioral measures alone.

D. PHYSIOLOGICAL MEASURES

As drivers become drowsy, their head begins to sway and the vehicle may wander away from the center of the lane. The previously described vehicle- based and vision based measures become apparent only after the driver starts to sleep, which is often too late to prevent an accident.

However, physiological signals start to change in earlier stages of drowsiness. Hence, physiological signals are more suitable to detect drowsiness with few false positives; making it possible to alert a drowsy driver in a timely manner and thereby prevent many road accidents.

Many researchers have considered the following physiological signals to detect drowsiness: electrocardiogram (ECG), electromyogram (EMG), electroencephalogram (EEG) and electro-oculogram (EoG). Some researchers have used the EoG signal to identify driver drowsiness through eye movements. The electric potential difference between the cornea and the retina generates an electrical field that reflects the orientation of the eyes; this electrical field is the measured EoG signal. Researchers have investigated horizontal eye movement by placing a disposable Ag-Cl electrode on the outer corner of each eye and a third electrode at the center of the forehead for reference. The electrodes were placed as specified so that the parameters - Rapid eye movements (REM) and Slow Eye Movements (SEM) which occur when a subject is awake and drowsy respectively, can be detected easily.

The heart rate (HR) also varies significantly between the different stages of drowsiness, such as alertness and fatigue. Therefore, heart rate, which can be easily determined by the ECG signal, can also be used to detect drowsiness.Others have measured drowsiness using Heart Rate Variability (HRV), in which the low (LF) and high (HF) frequencies fall in the range of 0.04–0.15 Hz and 0.14–0.4 Hz, respectively. HRV is a measure of the beat-to-beat (R-R Intervals) changes in the heart rate. The ratio of LF to HF in the ECG decreases progressively as the driver progresses from an awake to a drowsy state.

The Electroencephalogram (EEG) is the physiological signal most commonly used to measure drowsiness. The EEG signal has various frequency bands, including the delta band (0.5–4 Hz), which corresponds to sleep activity, the theta band (4–8 Hz), which is related to drowsiness, the alpha band (8–13 Hz), which represents relaxation and creativity, and the beta band (13–25 Hz), which corresponds to alertness. A decrease in the power changes in the alpha frequency band and an increase in the theta frequency band indicates drowsiness.

Akin et al. observed that the success rate of using a combination of EEG and EMG signals to detect drowsiness is higher than using either signal alone. The measurement of raw physiological signals is always prone to noise and artifacts due to the movement that is involved with driving. Hence, in order to eliminate noise, various preprocessing techniques, such as low pass filter, digital differentiators, have been used. In general, an effective digital filtering technique would remove the unwanted artifacts in an optimal manner. A number of statistical features are then extracted from the processed signal using various feature extraction techniques, including Discrete Wavelet Transform (DWT) and Fast Fourier Transform (FFT). The extracted features are then classified using Artificial Neural Networks (ANN), Support Vector Machines (SVM), Linear Discriminant Analysis (LDA), or other similar methods.

The reliability and accuracy of driver drowsiness detection by using physiological signals is very high compared to other methods. However, the intrusive nature of measuring physiological signals remains an issue to be addressed. To overcome this, researchers have used wireless devices to measure physiological signals in a less intrusive manner by placing the electrodes on the body and obtaining signals using wireless technologies like Zigbee, Bluetooth. Some researchers have gone further ahead by measuring physiological signals in a non intrusive way; by placing electrodes on the steering wheel or on the driver's seat. The signals obtained were then processed in android based smart phone devices and the driver was alerted on time. The accuracy of a non-intrusive system is relatively less due to movement artifacts and errors that occur due to improper electrode contact.

3. SYSTEM ANALYSIS

3.1 - REQUIREMENTS SPECIFICATION

1. Less Response Time
2. High Data Utility
3. Better Performance
4. Less computational intensity

Requirements Model

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

Requirements analysis is critical to the success or failure of a systems or software project. The requirements should be documented, actionable, measurable, testable, traceable, related to identified business needs or opportunities, and defined to a level of detail sufficient for system design.

User requirements

1. Execution time should be fast
2. More Accurate
3. User Friendly

Software requirements

1. O/S : Windows 18
2. Language : Python3
3. Tools and Technologies : Visual Studio Code, Machine Learning

Hardware requirements

System : Pentium IV 2.4 GHz

Hard Disk : 40GB

Monitor : 15 VGA Colour

Mouse : Logitech

RAM : 512MB

3.1.1 – Functional Requirements

- System should be able to monitor the driver by capturing video and take thevideo as input.
- Packages used are tkinter, scipy, imutils, numpy, dlib, cv2, argparse,playsound.
- Each frame of video generates facial landmarks and different assessment metrics are calculated for each frame.
- System should be able to identify the whether driver is drowsy or not correctly.
- System should alert the driver when the driver is said to have drowsiness.

3.1.2 - Non-Functional Requirements

- **Reliability:** The system is highly reliable.
- **Accessibility:** It can be easily accessible i.e click & run.
- **Efficiency:** Resource consumption for given load is quite low.
- **Fault tolerance:** Our system is not fault tolerant due to insufficienthardware.
- **Robustness:** Our system is not capable to cope with errors duringexecution.

3.2 - UML DIAGRAMS FOR THE PROJECT WORK

UML is an acronym that stands for Unified Modeling Language. Simply put, UML is a modern approach to modeling and documenting software. In fact, it's one of the most popular business process modeling techniques.

It is based on diagrammatic representations of software components. As the old proverb says: "a picture is worth a thousand words". By using visual representations, we are able to better understand possible flaws or errors in software or business processes.

The elements are like components which can be associated in different ways to make a complete UML picture, which is known as diagram. Thus, it is very important to understand the different diagrams to implement the knowledgein real-life systems.

Any complex system is best understood by making some kind of diagrams or pictures. These diagrams have a better impact on our understanding. If we look around, we will realize that the diagrams are not a new concept but it is used widely in different forms in different industries.

Mainly, UML has been used as a general-purpose modeling language in the field of software engineering. However, it has now found its way into the documentation of several business processes or workflows. For example, activity diagrams, a type of UML diagram, can be used as a replacement for flowcharts. They provide both a more standardized way of modeling workflows as well as a wider range of features to improve readability and efficiency. Use cases are best discovered by examining the actors and defining what the actor will be able to do with the system. Since all the needs of a system typically cannot be covered in one use case, it is usual to have a collection of use cases. Together this use casecollection specifies all the ways the system. An association provides a pathway for communication. The communication can be between use cases, actors, classes or interfaces. Associations are the most general of all relationships and consequentially the most semantically weak. If two objects are usually considered independently, the relationship is an association. They provide both a more standardized way of modeling workflows as well as a wider range of features to improve readability and efficiency. Use cases are best discovered by examining the actors and defining what the actor will be able to do with the system.

Since all the needs of a system typically cannot be covered in one use case, it is usual to have a collection of use cases. By default, the association tool on the toolbox is unidirectional and drawn on a diagram with a single arrow at one end of the association. The end with the arrow indicates who or what is receiving the communication. A dependency is a relationship between two model elements in which a change to one model element will affect the other model element. Typically, on class diagrams, a dependency relationship indicates that the operations of the client invoke operations of the supplier. The work flow in this case begins from importing the dataset by the developer and then replacing missing values with mean value of corresponding column, model building, validating that model by generating a confusion matrix and finally predicting the test sample class label. Transitions are used to show the passing of the flow of control from activity to activity.

The various UML diagrams are:

- Use case diagram
- Activity diagram
- Sequence diagram
- Collaboration diagram
- Object Diagram
- State Chart diagram
- Class diagram
- Component diagram
- Deployment diagram

Use case Diagram

A use case diagram is a graph of actors, a set of use cases enclosed by a system boundary, communication (participation) associations between the actors and users and generalization among use cases. The use case model defines the outside (actors) and inside (use case) of the system’s behavior. Actors are not part of the system. Actors represent anyone or anything that interacts with (input to or receive output from) the system. Use-case diagrams can be used during analysis to capture the system requirements and to understand how the system should work. During the design phase, you can use use-case diagrams to specify the behavior of the system as implemented. Use case is a sequence of transactions performed by a system that yields a measurable result of values for a particular actor. The use cases are all the ways the system may be used.

Use case is a list of actions or event steps, typically defining the interactions between a role (known as an actor) and a system, to achieve a goal. In case of the use case diagram developer and the end user are the actors. Use casesare best discovered by examining the actors and defining what the actor will be able to do with the system.

An include relationship specifies how behavior in the inclusion use case is used by the base use case. An extend relationship is a stereotyped relationship that specifies how the functionality of one use case can be inserted into thefunctionality of another use case. Extend relationships between uses cases are modeled as dependencies by using the Extend stereotype. The different use cases are import dataset for importing datasets, preprocessing ,model building, validation and prediction.

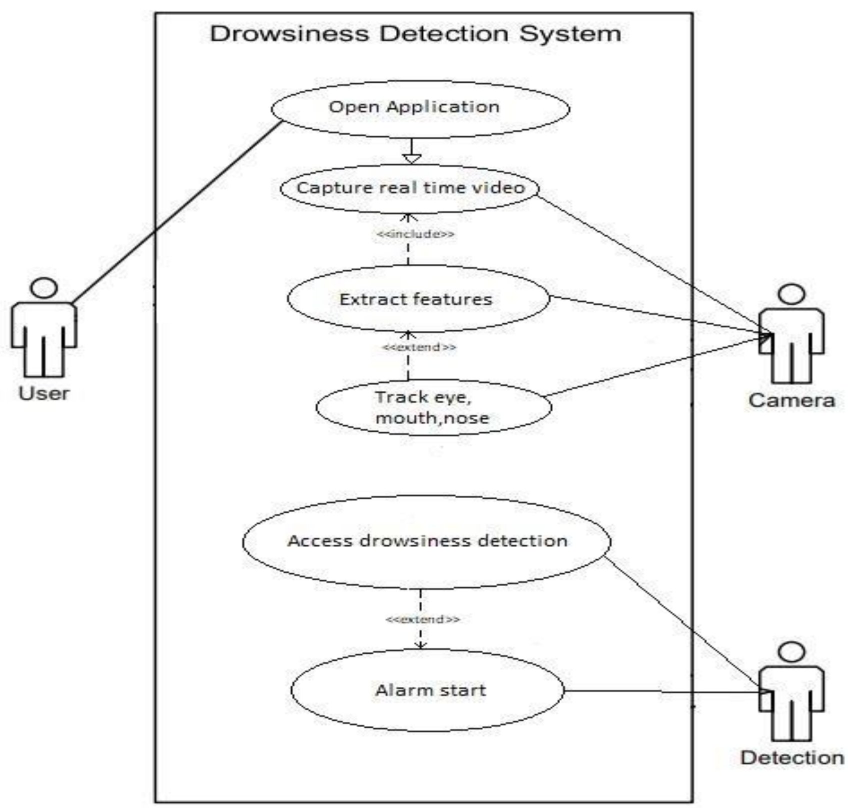

Fig - 3.2.1 - Use Case Diagram

Activity Diagram

An Activity diagram is a variation of a special case of a state machine, in which the states are activities representing the performance of operations and the transitions are triggered by the completion of the operations. The purpose of Activity diagram is to provide a view of flows and what is going on inside a use case or among several classes. Activity diagrams contain activities, transitions between the activities, decision points, and synchronization bars. An activity represents the performance of some behavior in the workflow. In the UML, activities are represented as rectangles with rounded edges, transitions are drawn as directed arrows, decision points are shown as diamonds, and synchronization bars are drawn as thick horizontal or vertical bars as shown in the following. The activity icon appears as a rectangle with rounded ends with a name and a component for actions.

Swim lanes may be used to partition an activity diagram. This typically is done to show what person or organization is responsible for the activitiescontained in the swim lane. Swim lanes are helpful when modeling a business workflow because they can represent organizational units or roles within a business model. Swim lanes are very similar to an object because they provide a way to tell who is performing a certain role. Swim lanes only appear on activity diagrams. When a swim lane is dragged onto an activity diagram, it becomes a swim lane view. Swim lanes appear as small icons in the browse while swim lane views appear between the thin, vertical lines with a header that can berenamed and relocated. An activity represents the performance of some behavior in the work flow. In the UML, activities are represented as rectangles with rounded edges, transitions are drawn as directed arrows, decision points are shown as diamonds, and synchronization bars are drawn as thick horizontal or vertical bars as shown in the following. The activity icon appears as a rectangle with rounded ends with a name and a component for actions. An Activity diagram is a variation of a special case of a state machine, in which the states are activities representing the performance of operations and the transitions are triggered by the completion of the operations. The purpose of Activity diagram is to provide a view of flows and what is going on inside a use case or among several classes.

Activity diagrams can be regarded as a form of a structured flowchart combined with a traditional data flow diagram. Typical flowchart techniques lack constructs for expressing concurrency. However, the join and split symbols in activity diagrams only resolve this for simple cases; the meaning of the model is not clear when they are arbitrarily combined with decisions or loops. Arrows run from the start towards the end and represent the order in which activities happen.

Activity diagrams are constructed from a limited number of shapes, connected with arrows. The most important shape types:

- ellipses represent actions.
- diamonds represent decisions.
- bars represent the start (split) or end (join) of concurrent activities.
- a black circle represents the start (initial node) of the work flow.
- an encircled black circle represents the end (final node).

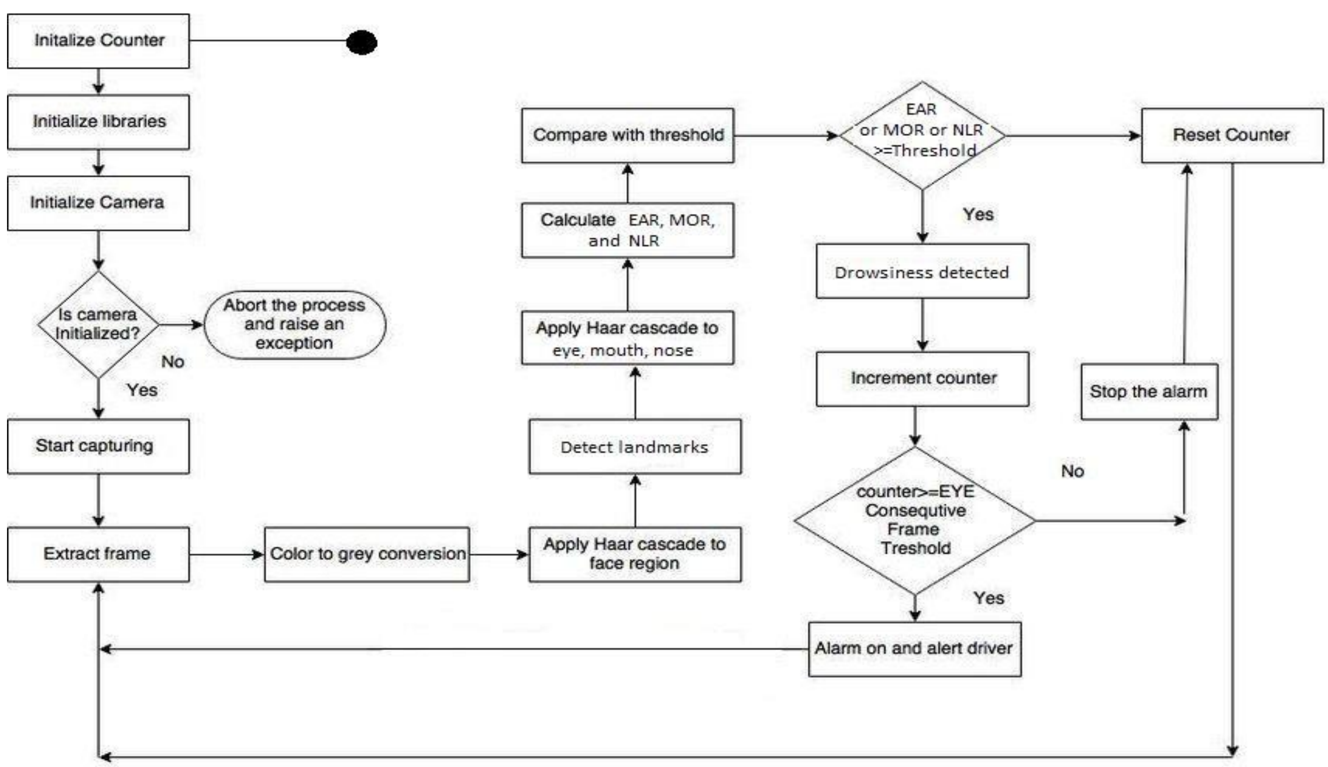

Fig - 3.2.2 Activity Diagram

Sequence Diagram

A sequence diagram is an interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. 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 the system under development. Sequence diagrams are sometimes called as event diagrams.

A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner. If the lifeline is that of an object, it demonstrates a role. Leaving the instance name blank can represent anonymous and unnamed instances.

Messages, written with horizontal arrows with the message name written above them, display interaction. Solid arrow heads represent synchronous calls, open arrow heads represent asynchronous messages, and dashed lines represent reply messages. If a caller sends a synchronous message, it must wait until the message is done, such as invoking a subroutine. If a caller sends an asynchronous message, it can continue processing and doesn’t have to wait for a response. Asynchronous calls are present in multithreaded applications, event-driven applications and in message- oriented middleware. Activation boxes, or method- call boxes, are opaque rectangles drawn on top of lifelines to represent that processes are being performed in response to the message.

Objects calling methods on themselves use messages and add new activation boxes on top of any others to indicate a further level of processing. If an object is destroyed, an X is drawn on bottom of the lifeline, and the dashed line ceases to be drawn below it. It should be the result of a message, either from the object itself, or another.

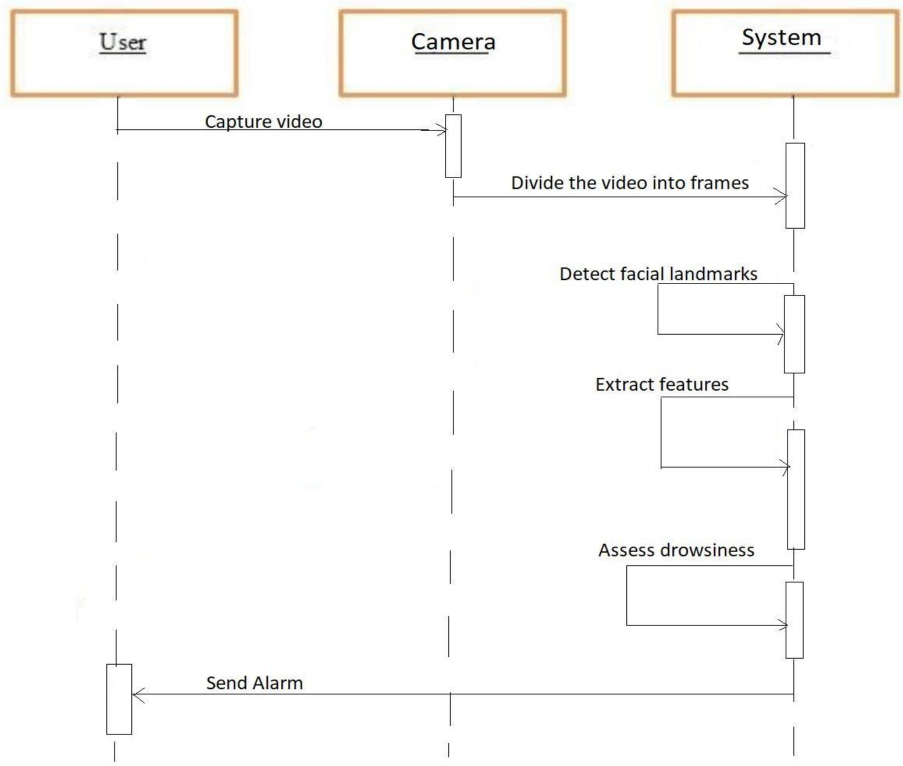

Fig - 3.2.3 Sequence Diagram

Collaboration Diagram

A collaboration diagram shows that the order of messages that implement an operation or a transaction. Collaboration diagrams show objects, their links, and their messages. They can also contain simple class instances and class utility instances. Each collaboration diagram provides a view of the interactions or structural relationships that occur between objects and object like entities in the current model. Collaboration diagrams and sequence diagrams are called interaction diagrams. A collaboration diagram shows that the order of messages that implement an operation or a transaction. Collaboration diagrams show objects, their links, and their messages. They can also contain simple class instances and class utility instances. Each collaboration diagram provides a viewof the interactions or structural relationships that occur between objects and object like entities in the current model.

The second interaction diagram is the collaboration diagram. It shows theobject organization as seen in the following diagram. In the collaboration diagram, the method call sequence is indicated by some numbering technique. The number indicates how the methods are called one after another. We have taken the same order management system to describe the collaboration diagram. Method calls are similar to that of a sequence diagram. However, difference being the sequence diagram does not describe the object organization, whereas the collaboration diagram shows the object organization. To choose between these two diagrams, emphasis is placed on the type of requirement. If the time sequence is important, then the sequence diagram is used. If organization is required, then collaboration diagram is used. interaction diagrams are used to describe the dynamic nature of a system. Now, we will look int the practical scenarios where these diagrams are used.

The main purpose of both the diagrams are similar as they are used to capture the dynamic behavior of a system. However, the specific purpose is more important to clarify and understand. Sequence diagrams are used to capture the order of messages flowing from one object to another. Collaboration diagrams are used to describe the structural organization of the objects taking part in the interaction. A single diagram is not sufficient to describe the dynamic aspect of an entire system, so a set of diagrams are used to capture it as a whole.

Interaction diagrams are used when we want to understand the message flow and the structural organization. Message flow means the sequence of control flow from one object to another. Structural organization means the visual organization of the elements in a system.

Generally, the labels on a collaboration diagram are determined by the needs of the user base. Someone creating this kind of resource may use actual file names, generic phrases representing the function of programs, or even customized icons to show how pieces of a system work together. Customized collaboration diagrams can help business leaders and others to see more about what’s going on within a complex IT system and how software interactions work.

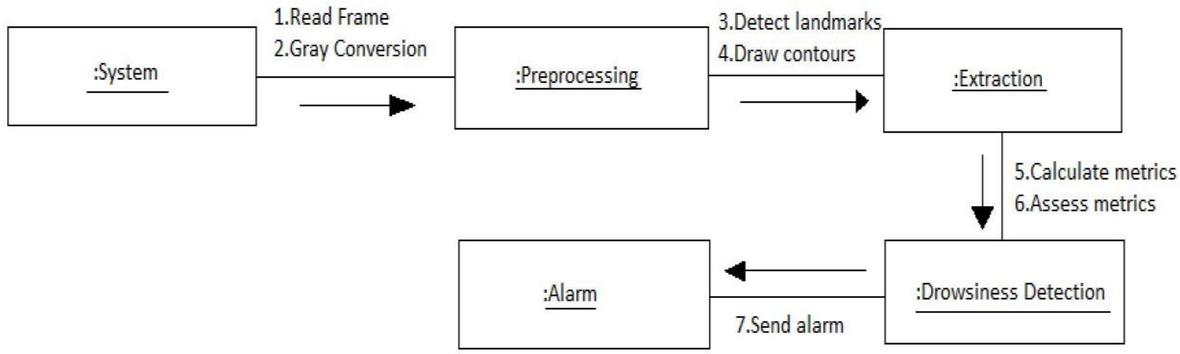

Fig- 3.2.4 Collaboration Diagram

State Chart Diagram

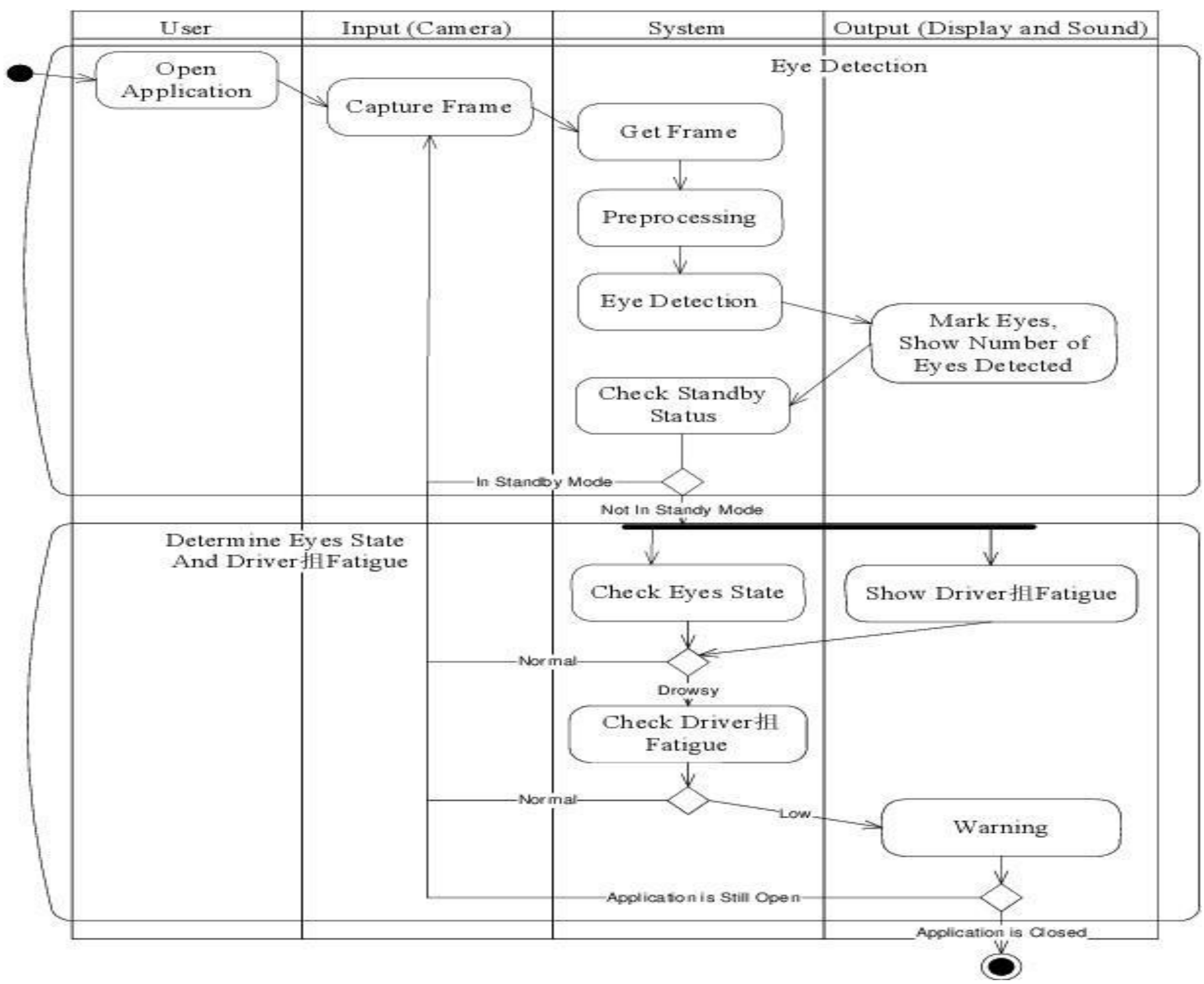

Fig - 3.2.5 State Chart Diagram

Use cases and scenarios provide a way to describe system behavior; inthe form of interaction between objects in the system. Sometime it is necessary to consider inside behavior of an object. A state chart diagram shows the states of a single objects, the events or messages that cause a transition from one state to another and the actions that result from a state change.

As I activity diagram, state chart diagram also contains special symbols for start state and stop state. State chart diagram cannot be created for every class in the system, it isonly for those class objects with significant behavior.

State transition

A state transition indicates that an object in the source state will perform certain specified actions and enter the destination state when a specified event occurs or when certain conditions are satisfied. A state transition is a relationship between two states, two activities, or between an activity and a state.

We can show one or more state transitions from a state as long as each transition is unique. Transitions originating from a state cannot have the same event, unless there are conditions on the event.

Provide a label for each state transition with the name of at least one event that causes the state transition. You do not have to use unique labels for state transitions because the same event can cause a transition to many different statesor activities.

Class Diagram

Class diagrams contain icons representing classes, interfaces, and their relationships. You can create one or more class diagrams to represent the classes atthe top level of the current model; such class diagrams are themselves contained by the top level of the current model. You can also create one or more class diagrams to represent classes contained by each package in your model; such class diagrams are themselves contained by the package enclosing the classes they represent; the icons representing logical packages and classes in class diagrams.

1. Class diagrams are created to provide a picture or view of some or all of the classes in the model.

2. The main class diagram in the logical view of the model is typically a picture of the packages in the' system. Each package also has its own main class diagram, which typically displays the public classes of the package.

A class diagram is a picture for describing generic descriptions of possible systems. Class diagrams and collaboration diagrams are alternate representations of object models.

A Class is a description of a group of objects with common properties (attributes) common behavior (operations), common relationships to other objects, and common semantics. Thus, a class is a template to create objects. Each object isan instance of some class and objects cannot be instances of more than one class.

In the UML, classes are represented as compartmentalized rectangles.

1. The top compartment contains the name of the class.
2. The middle compartment contains the structure of the class (attributes).
3. The bottom compartment contains the behavior of the class(operations).

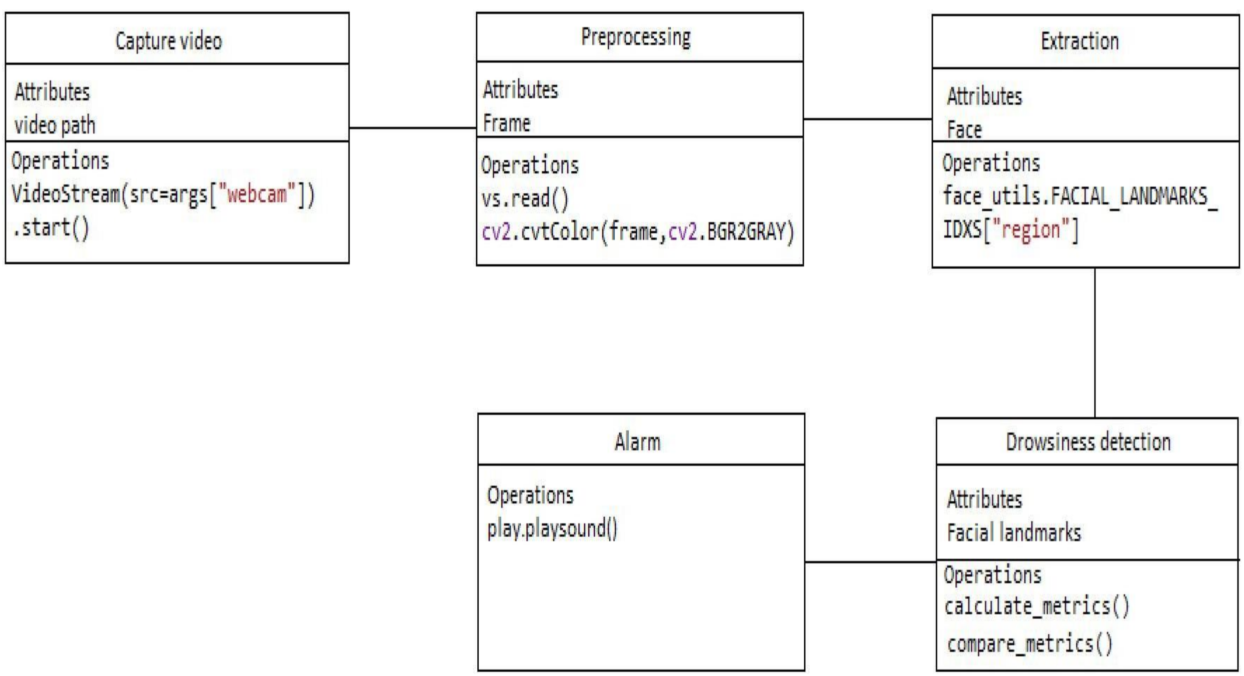

Fig - 3.2.6 Class Diagram

Component Diagram

Component diagram does not describe the functionality of the system but it describes the components used to make those functionalities. So from that point component diagrams are used to visualize the physical components in a system. These components are libraries, packages, files etc. Component diagrams can also be described as a static implementation view of a system. Static implementation represents the organization of the components at a particular moment. A single component diagram cannot represent the entire system but a collection of diagrams are used to represent the whole.

So the purpose of the component diagram can be summarized as:

1. Visualize the components of a system.
2. Construct executables by using forward and reverse engineering.
3. Describe the organization and relationships of the components.

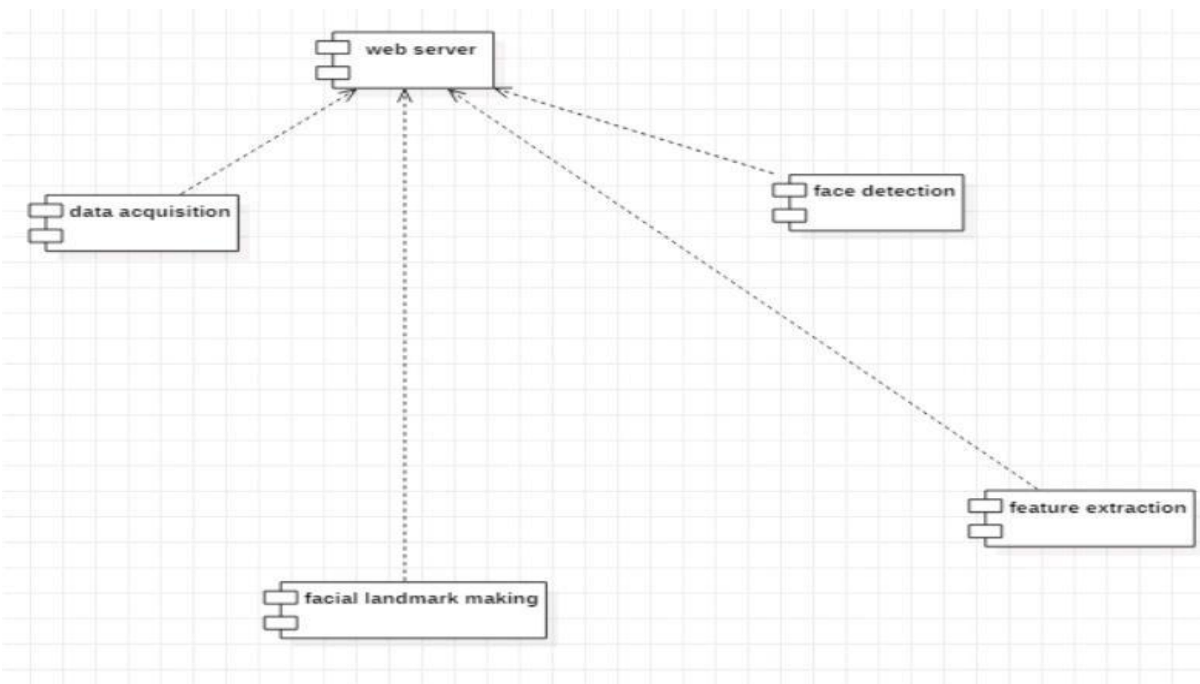

Fig - 3.2.7 Component Diagram

Deployment Diagram

The second type of implementation diagram provided by UML is the deployment diagram. Deployment diagrams are used to show the configuration of run-time processing elements and the software components and processes that are located on them.

Deployment diagrams are made up of nodes and communication associations. Nodes are typically used to show computers and the communication associations show the network and protocols that are used to communicate between nodes. Nodes can be used to show other processing resources such as people or mechanical resources.

Nodes are drawn as 3D views of cubes or rectangular prisms, and the following figure shows a simplest deployment diagram where the nodes connected by communication associations.

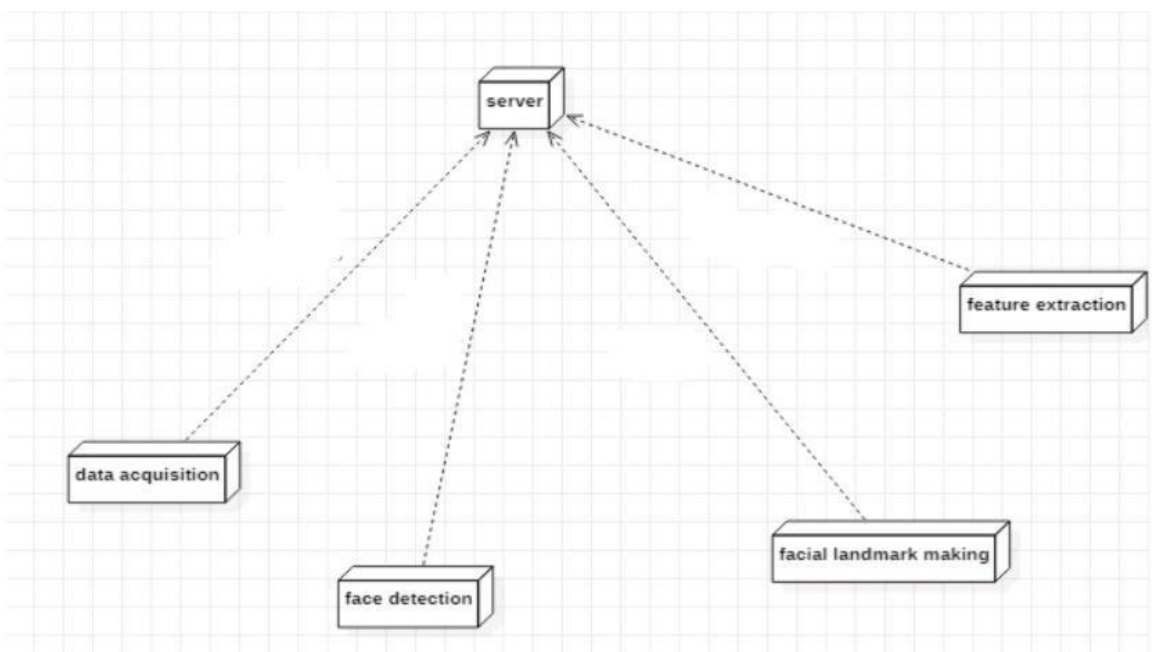

Fig - 3.2.8 Deployment Diagram

4. SYSTEM DESIGN

4.1 - ARCHITECTURE OF THE PROPOSED SYSTEM

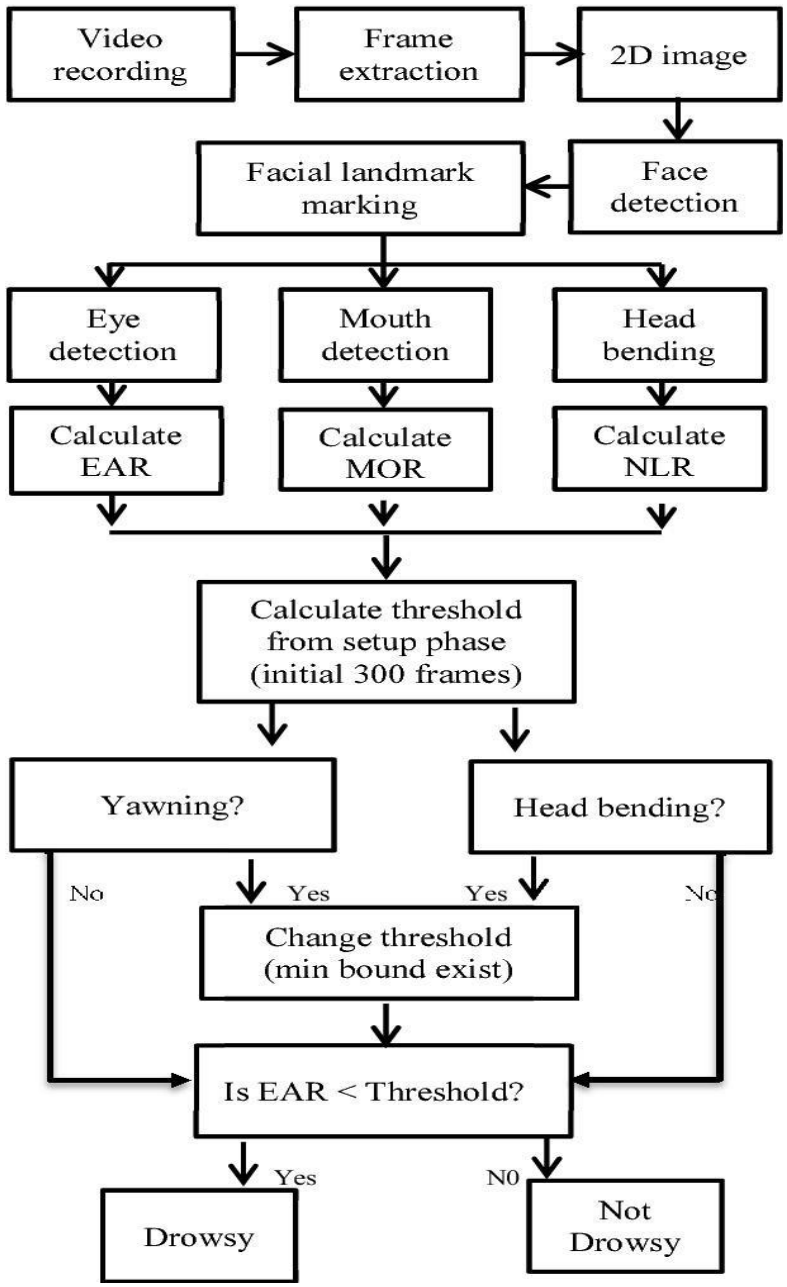

Fig - 4.1.1 Architecture

The project was developed to provide further information for traffic safety and others could use in their efforts to reduce the number of drowsy related crashes. A block diagram of the proposed driver drowsiness monitoring system has been depicted in Fig.4.1.1. 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) or using OPENCV or 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.

4.2 - WORKFLOW OF THE PROPOSED SYSTEM

The critical issue that a fatigue detection system must address is the question of how to accurately and early detect fatigue at the initial stage. Possible non intrusive techniques for detecting fatigue in drivers using computer vision, capturing video, extract frames, detect face in each frame and marking facial landmarks.

The system uses Histogram Oriented Gradient (HOG) feature descriptor for face detection and facial points recognition. Then SVM is used to checkwhether detected object is face or non-face. It further monitors the Eye Aspect Ratio (EAR) , Mouth Opening Ratio (MOR) and Nose Length Ratio (NLR) of the driver up to a fixed number of frames to check the sleepiness, yawning and head bending. If any of these is observed then drowsiness is identified and an alarm is sent.

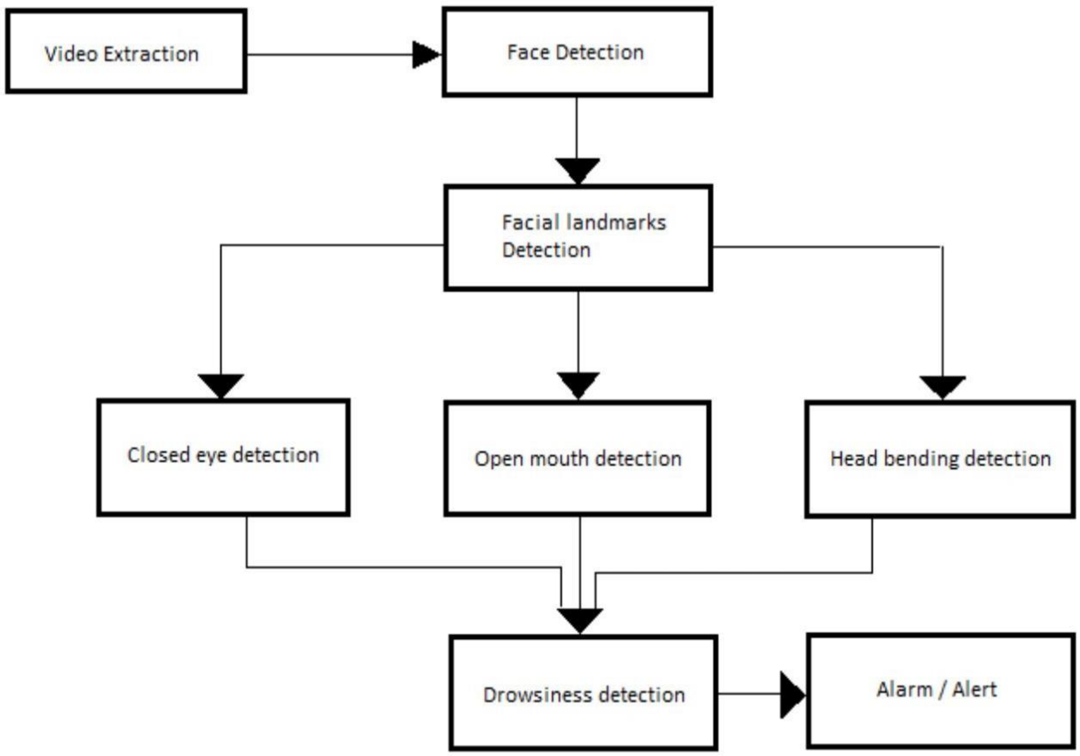

Fig - 4.2.1 Workflow

4.3 - MODULE DESCRIPTION

There are five modules to be implemented in this driver drowsiness monitoring system using visual behaviour and machine learning, they are:

4.3.1 Data Acquisition

4.3.2 Face Detection

4.3.3 Facial landmark marking

4.3.4 Feature Extraction

4.3.5 Classification

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.

B. Face Detection

After extracting the frames, first the human faces are detected in each frame. Numerous online face detection algorithms are there. Human faces are detected using histogram of oriented gradients (HOG) and linear SVM method. 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 training 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 drawnaround the face.

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 the 500 pixels and converted to gray- scale image.

After image normalization, ensemble of regression trees is used to estimate the landmark positions on the face from a sparse subset of pixel intensities. In this method, the sum of square error loss is optimized using gradient boosting learning. Different priors are used to find different structures.Using this method, the boundary points of eyes, mouth and the central line of the nose are marked and the number of points for eye, mouth and nose are given in below table.

Parts	**Landmark Points**
Mouth	[13-24]
Right eye	[1-6]
Left eye	[7-12]
Nose	[25-28]

Table 4.3.1 Facial landmark points

The facial landmarks are shown in below Figure. The red points are the detected landmarks for further processing.

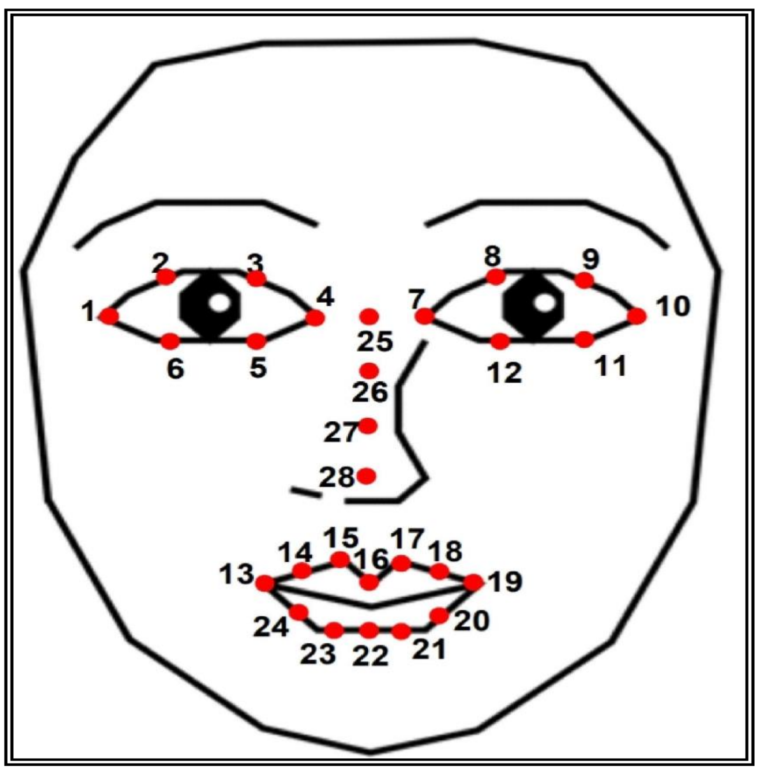

Fig - 4.3.1 Facial landmark points

D. Feature Extraction

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

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

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

where p_i represents point marked as i in facial landmark and (p_i - p_j) is the distance between points marked as i and j. Therefore, when the eyes are fully open, EAR is high value and as the eyes are closed, EAR value goes towards zero. Thus, monotonically decreasing EAR values indicate gradually closing eyes and it's almost zero for completely closed eyes (eye blink). Consequently, EAR values indicate the drowsiness of the driver as eye blinks occur due to drowsiness.

Mouth opening ratio (MOR): Mouth opening ratio is a parameter to detect yawning during drowsiness. Similar to EAR, it is calculated as

$$MOR = \frac{(p_{15} - p_{23}) + (p_{16} - p_{22}) + (p_{17} - p_{21})}{3(p_{19} - p_{13})}$$

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.

Head Bending: Due to drowsiness, usually driver's head tilts (forward or backward) with respect to vertical axis. So, from the head bending angle,driver drowsiness can be detected.

As the projected length of nose on the camera focal plane is proportional to this bending, it can be used as a measure of head bending. In normal condition, our nose makes an acute angle with respect to focal plane of the camera. This angle increases as the head moves vertically up and decreases on moving down. Therefore, the ratio of nose length to an average nose length while awake is a measure of head bending and if the value is greater or less than a particular range, it indicates head bending as well as drowsiness. From the facial landmarks, the nose length is calculated and it is defined as

$$NLR = \frac{nose\ length(\mathrm{p}_{28} - \mathrm{p}_{25})}{average\ nose\ length}$$

The average nose length is computed during the setup phase of the experiment.

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 seventy five (for 10s at 30 fps) frames are recorded. Out of these seventy five initial frames containing face, average of 75 values is considered as thehard 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.

Head bending feature is used to find the angle made by head with respect to vertical axis in terms of ratio of projected nose lengths. Normally, NLR has values from 0.9 to 1.1 for normal upright position of head and it increases or decreases when head bends down or up in the state of drowsiness. The average nose length is computed as the average of the nose lengths in the setup phase assuming that no head bending is there. After computing the threshold values, the system is used for testing.

The system detects the drowsiness if in a test frame drowsiness is detected for at least one feature. To make this thresholding more realistic, the decision for each frame depends on the last 15 frames. If at least 10 frames(out of those 15) satisfy drowsiness conditions for at least one feature, then the system gives drowsiness detection indication and the alarm. To make this thresholding adaptive, another single threshold value is computed which initially depends on EAR threshold value. The average of EAR values is computed as the average of 75 values out of 75 frames in the setup phase. Then offset is determined heuristically and the threshold is obtained as offsetsubtracted from the average value.

Driver safety is at risk when EAR is below this threshold. This EAR threshold value increases slightly with each yawning and head bending up to a certain limit. As each yawning and head bending is distributed over multiple frames, so yawning and head bending of consecutive frames are considered as single yawn and head bending and added once in the adaptive threshold. In a test frame, if EAR value is less than this adaptive threshold value, then drowsiness is detected and an alarm is given to the driver. Sometimes it may happen that when the head is too low due to bending, the system is unable to detect the face. In such situation, previous three frames are considered and if head bending was detected in the three frames, drowsiness alert will be shown.

5. IMPLEMENTATION

5.1 - ALGORITHMS

5.1.1 - HAAR FEATURES

The face detection method uses Haar features for face detection. Haar features are extracted by using a set of rectangular black and white windows. The black color has a weight of -1 and the white region has weight 0. The windows are first applied to the image and corresponding values are multiplied with the pixel intensities. Then these values are added together, and the Haar feature corresponding to the window used is obtained. But all the Haar features extracted are not required for successful detection of faces. Hence the most important features that can be used for face detection are taken.

5.1.2 - CASCADE CLASSIFIERS

Once the Haar features are obtained then individual classifiers are built based on the values of each Haar feature. These individual classifiers are then arranged intoa cascade classifier. A cascaded classifier is combination of several classifiers arranged in the different stages cascaded on after one another. The number of classifiers in each stage and their threshold values are determined by the boosting algorithm during the training of the classifiers with labeled face images. The cascade classifier used here has 22 stages and a total of 2135 features.

5.1.3 - HOG FEATURES

The HOG features method was developed by Dalal and Trigs in 2005. It is a feature descriptor that is used for various object detection applications in the field of computer vision. The key idea of the HOG features is to group gradient magnitudes into bins in a histogram based on its orientation. HOG features are extracted from the eye images obtained in the previous section. For that the image is first resizes into 24x24 pixels. Then the image is divided into 4 blocks of size 16x16 pixels with each block overlapping half of the region covered by the preceding block.

Each block has 4 cells of size 8x8 pixels. Next the gradients are computed for each pixel inside the cells using Sobel filters. These gradients magnitudes are then plotted in a histogram which has magnitudes on the y-axisand orientations in the x-axis. The x-axis is divided into 9 bins, each bin having a width of 20 degrees. The gradient magnitudes are then arranged into these 9-bin histograms based on their orientation. The value of each of these 9 bins corresponds to the feature values extracted from each cell of the image. Thus each cell of 64 pixels is represented using 9 feature values. Then these feature values of all the cells inside each block are concatenated to obtain the final feature descriptor.

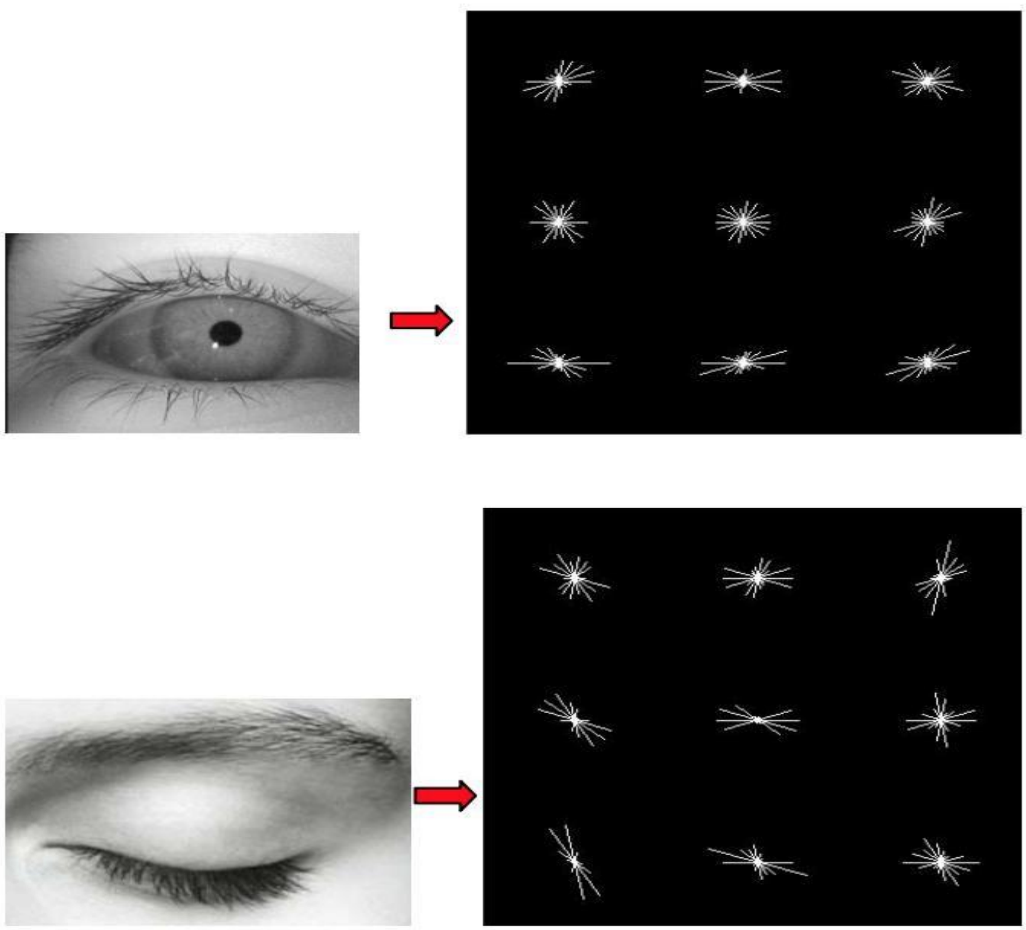

Fig - 5.1.3.1 HOG features extracted from open and closed eyes

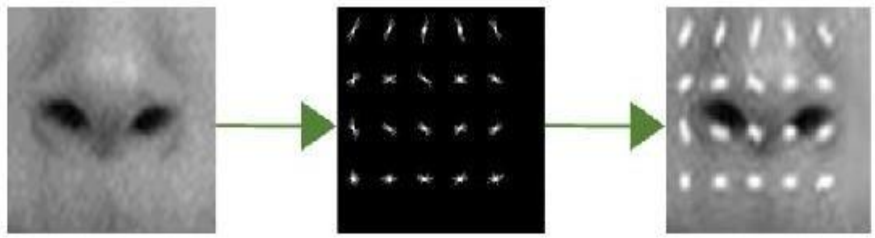

Fig - 5.1.3.2 HOG features extracted from nose

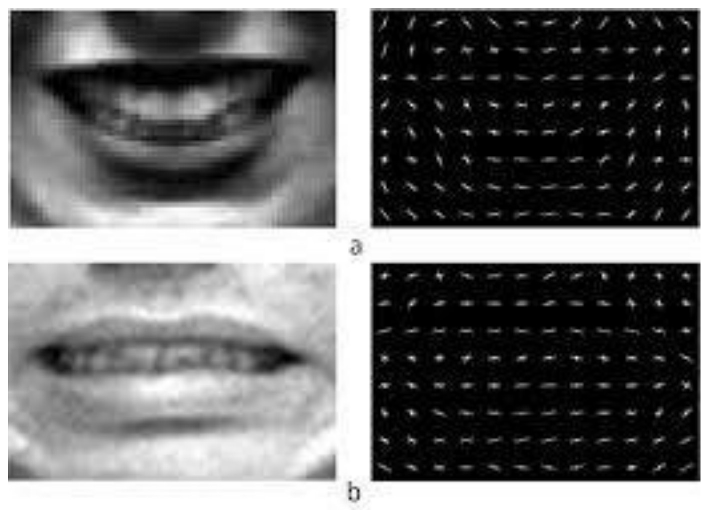

Fig - 5.1.3.3 HOG features extracted from mouth

5.1.4 - SVM CLASSIFIER

Support Vector Machines were initially developed by Vapnik and his team. It was later improved by other researchers. It is a statistical learning model used commonly for classification problems. Let the data points of the eye images be represented by: $(h1,y1)$, $(h2,y2)$ (hn,yn) were hi represent the HOG feature vector representing the nth eye image and y_n represent the class of the nth eye image. y can have two values 0 or a 1. 0 represents the closed eye image and 1 represents the open eye image. The basic idea of the HOG features is to find a hyper plane with the maximum margin that separates the two classes.

In case of linearly separable data the hyper plane in terms of support vectors is given by:

$$f(h) = \sum_{i=1}^{n} \alpha_i y_i h_i(.)h(i) + b$$

where yi denotes the class of the data point hi and h(i) represents the supportvector machines. This is a Lagrange optimization problem and it is solved using Lagrange multipliers α_i (i = 1,...l). In case the data is not linearly separable then it is first mapped into another feature space where it is linearly separable using Kernel function. Then the equation becomes:

$$f(h) = \sum_{i=1}^{n} \alpha_i y_i V(h_i, h(i)) + b$$

where V represents the Kernel function. In the presented system the SVM classifier uses a Gaussian Radial Basis function for mapping the data into another feature space. The classifiers are trained using 40 eye images each of open and closed eyes. Once the classifier is trained then the test images are classified applied to the classifier for classification.

5.2 - DATA SOURCES USED

For experimental analysis, a pre-trained HOG + Linear SVM object detector was employed mainly for the task of face detection. This pre-trained object detector is shape_predictor_68_face_landmarks.dat. This model creates predictor object. The video of the subject under test is captured using the web camera. The video captured is stored as a collection of frames (images). Each of these frames are extracted and processed separately. Each frame is sent to the pre- trained object detector to detect face. Packages like tkinter, scipy, imutils, numpy, dlib, cv2, argparse, playsound are used.

6.TESTING

6.1 – Introduction To Testing

6.1.1 - Unit Testing

Unit testing focuses verification effort on the smallest unit of Software design that is the module. Unit testing exercises specific paths in a module's control structure to ensure complete coverage and maximum error detection. This test focuses on each module individually, ensuring that it functions properly as a unit. Hence, the naming is Unit Testing. During this testing, each module is tested individually and the module interfaces are verified for the consistency with design specification. All the important processing path are tested for the expected results. All error handling paths are also tested.

6.1.2 - Integration Testing

Integration testing addresses the issues associated with the dual problems of verification and program construction. After the software has been integrated a set of high order tests are conducted. The main objective in this testing process isto take unit tested modules and builds a program structure that has been dictatedby design.

6.1.3 - User Acceptance Testing

User Acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes wherever required. The system developed provides a friendly user 50interface that can easily be understood even by a person who is new to the system.

6.1.4 - Output Testing

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in the specified format.

Asking the users about the format required by them tests the outputs generated or displayed by the system under consideration.Hence the output format is considered in 2 ways - one is on screen and another in printed format.

6.2 - Test Cases

The test cases are tabulated with the fields test case name, test case description, test steps including step, expected and actual and test status(P/F). The test cases are shown below:

Test Case 1:

TestCaseId	Test Scenario	TestCase	Pre - Condition	Test Steps	Expected Result	Post - Condition	Actual Result	Test Status (P/F)
1	Capture the Video Driver	Import the frames of the video	Run Application	Select a Frame, find face and Detect Landmarks	Eyes Closed	Driver Closed Eyes in a Frame	Displayed Eyes Closed	P

Table 6.2.1 - Test case 1

Test Case 2:

TestCaseId	Test Scenario	TestCase	Pre - Condition	Test Steps	Expected Result	Post - Condition	Actual Result	Test Status (P/F)
2	Capture	Import the frames of the video	Run Application	Select a frame, find face and detect Landmarks	Yawning	Driver opened mouth in a frame	Displayed yawning	P

Table 6.2.2 - Test case 2

Test Case 3:

Test Case Id	Test Scenario	TestCase	Pre - Condition	Test Steps	Expected Result	Post - Condition	Actual Result	Test Status (P/F)
3	Capture the video of the driver	Import the frames of the video.	Run application	Select a frame, find face and Detect Landmarks	Head bending	Driver bent head in a frame	Displayed Head Bending	P

Table 6.2.3 - Test case 3

Test Case 4:

TestCaseId	Test Scenario	TestCase	Pre - Conditio n	Test Steps	Expected Result	Post - Condition	Actual Result	Test Status (P/F)
4	Capture the video of the driver	Import the frames of the video	Run Application	Select a frame, find face and detect landmarks	Drowsiness Alert and ring alarm	Driver closed Eyes or is Yawning or bent head for More than 10 Previous frames	Displayed Drowsiness Alert and alarm is sent	P

Table 6.2.4 - Test Case 4

7. RESULTS

7.1 - Actual Results

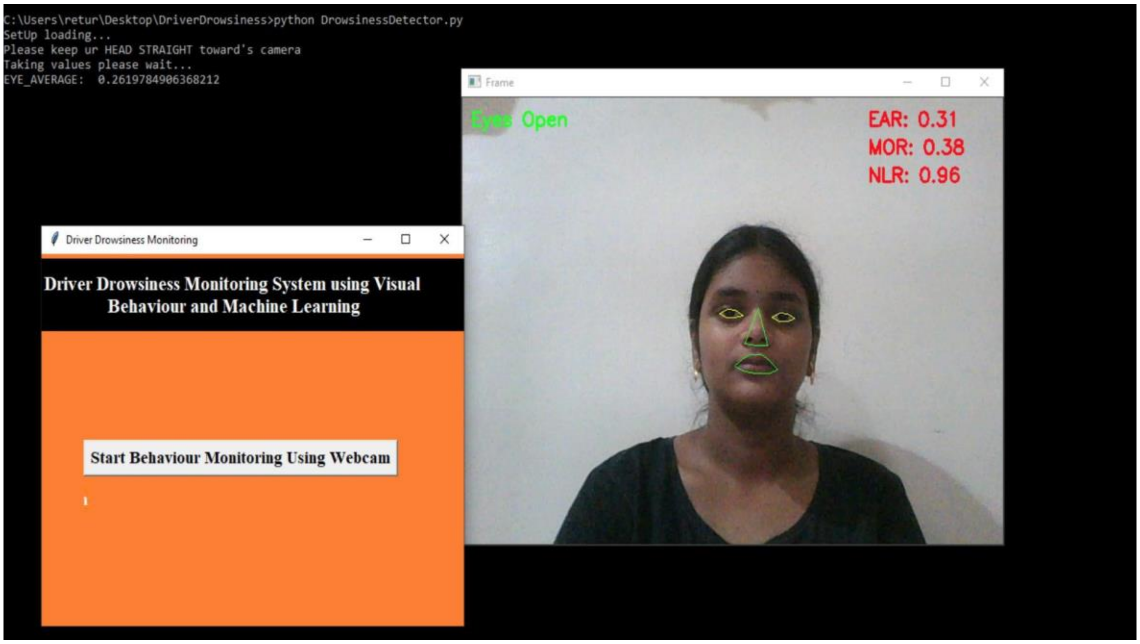

Fig 7.1.1 - Frame when driver is not drowsy

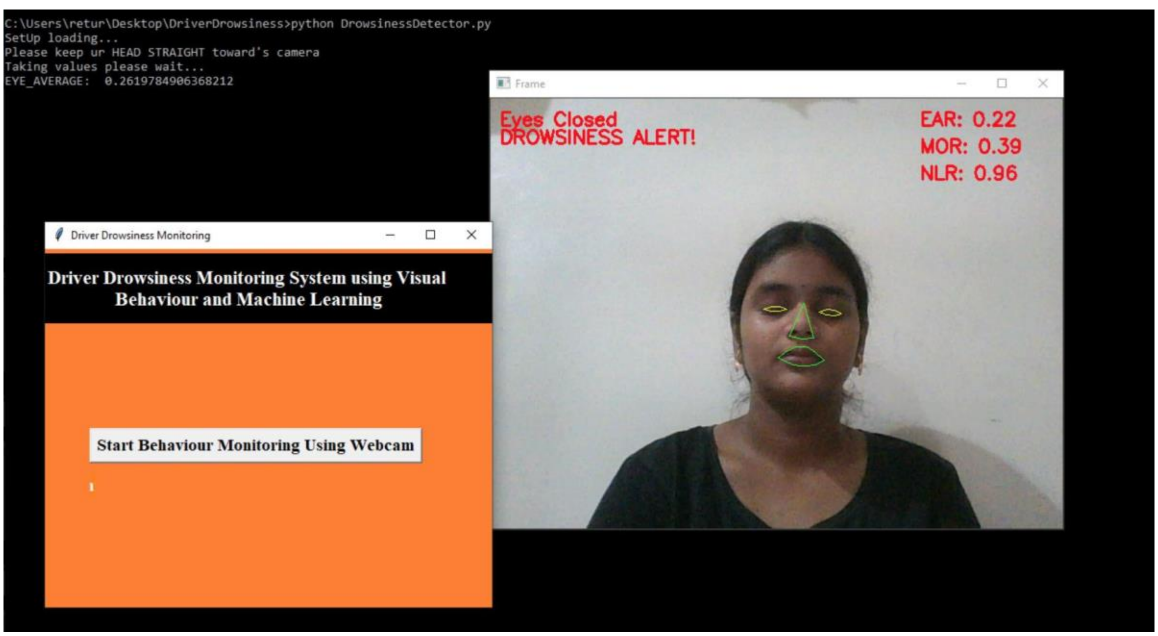

Fig 7.1.2 - Frame when driver closes eyes

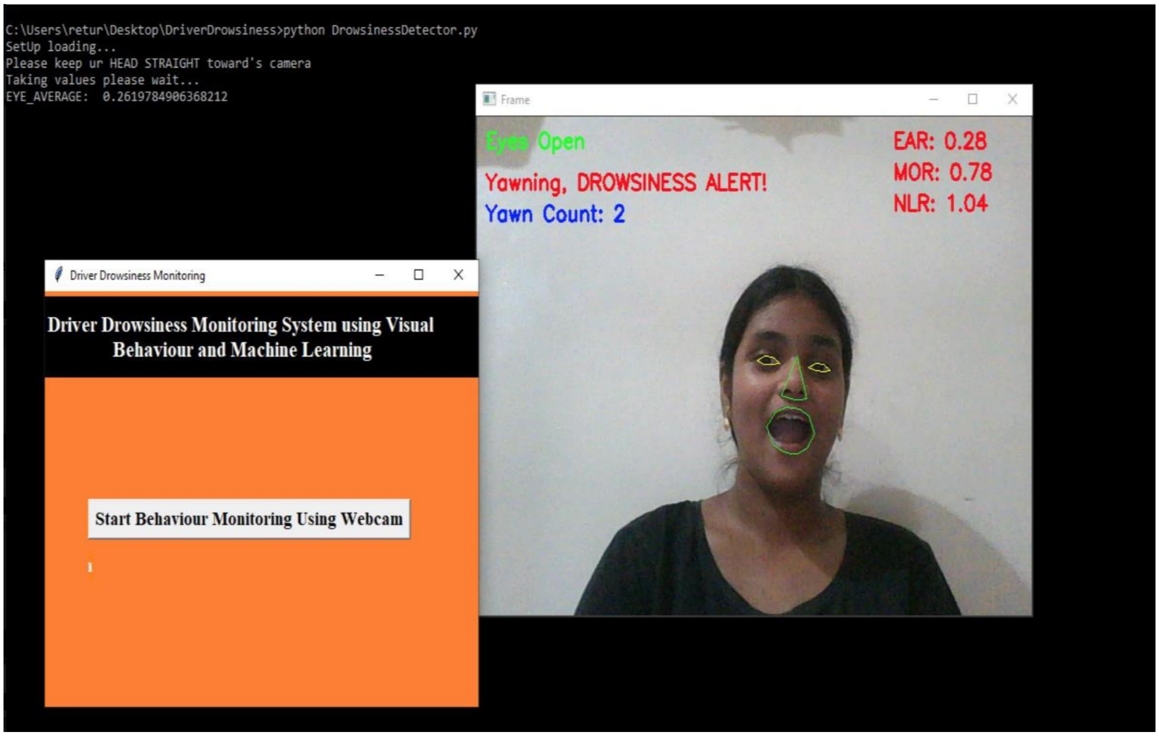

Fig 7.1.3 - Frame when driver yawns

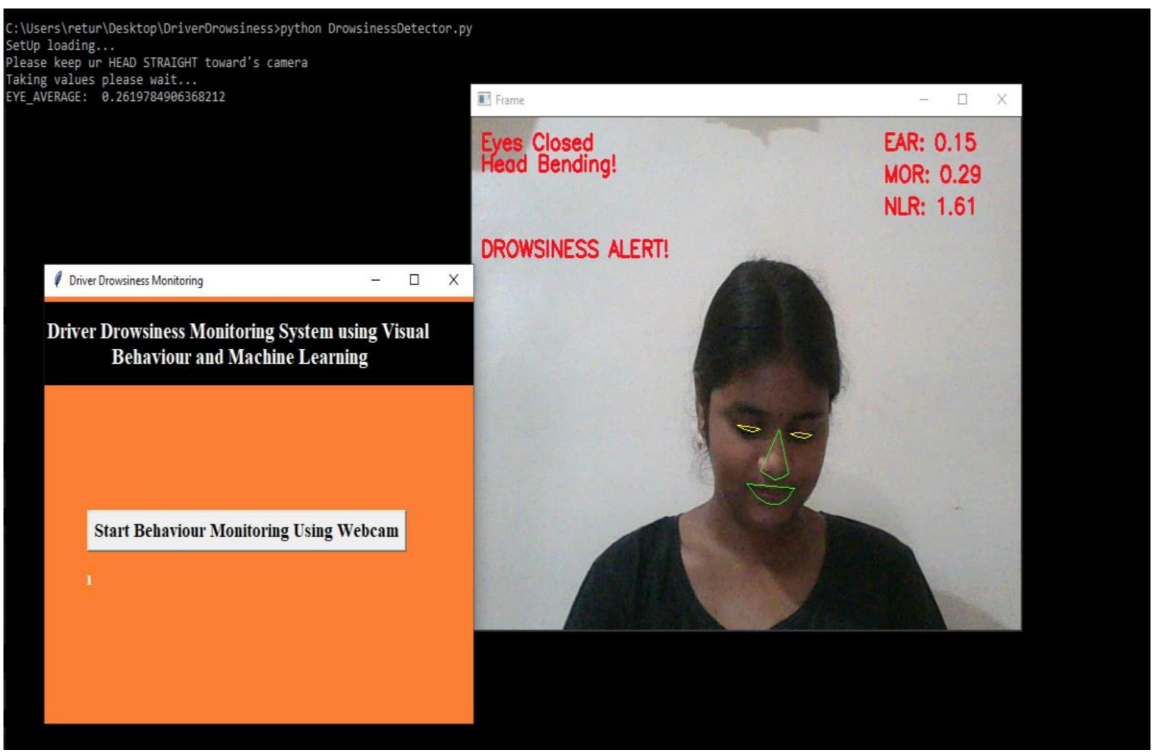

Fig 7.1.4 - Frame when driver bends head

7.2 - Analysis of obtained results

The driver gets alert while closing eyes or yawning or bending head. Alarm is sent only when this behaviour is observed for continuous 10 frames. If alarm is rung when actual drowsiness is not there, it causes disturbance to driver. This distracts the driver. Therefore we implemented in such a way that the driver doesn't get alarm when actual drowsiness is not there.

The sample values of different parameters for different states are given in the **below table.**

State	EAR	MOR	NLR
Normal	0.35	0.34	1.003
Yawning	0.22	0.77	0.76
Eye Closed	0.15	0.419	0.876
Head Bending	0.15	0.577	0.66

Table 7.2.1 Sample values of different parameters for different states

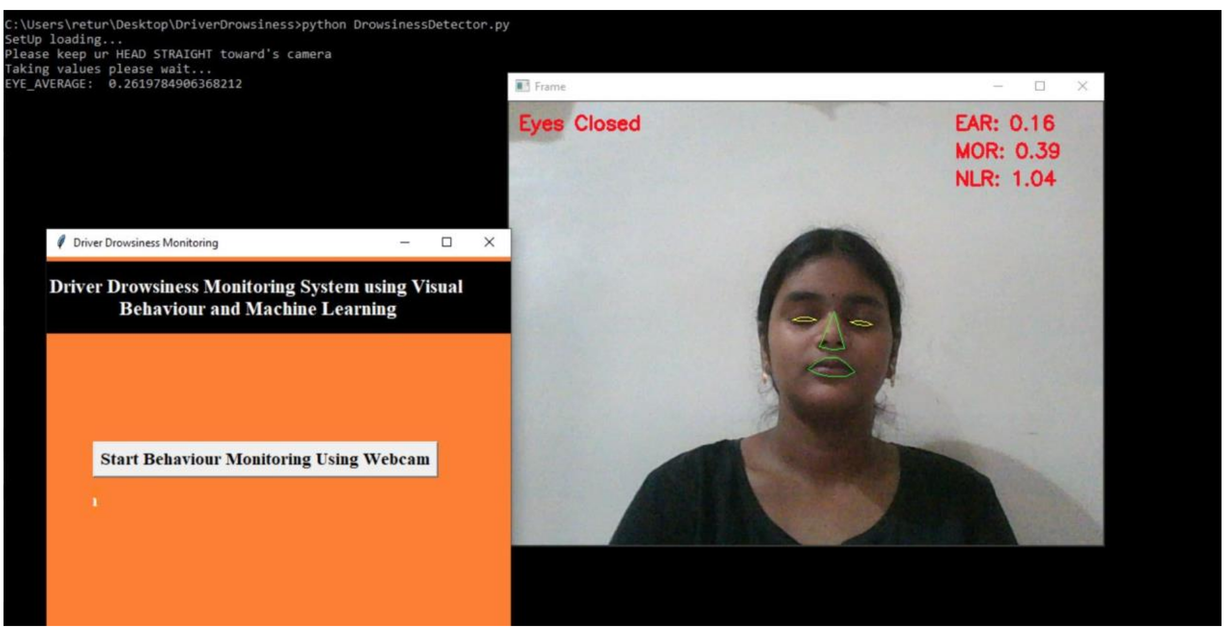

Fig 7.2.1 - Frame where driver closes eyes for a second but not continuously

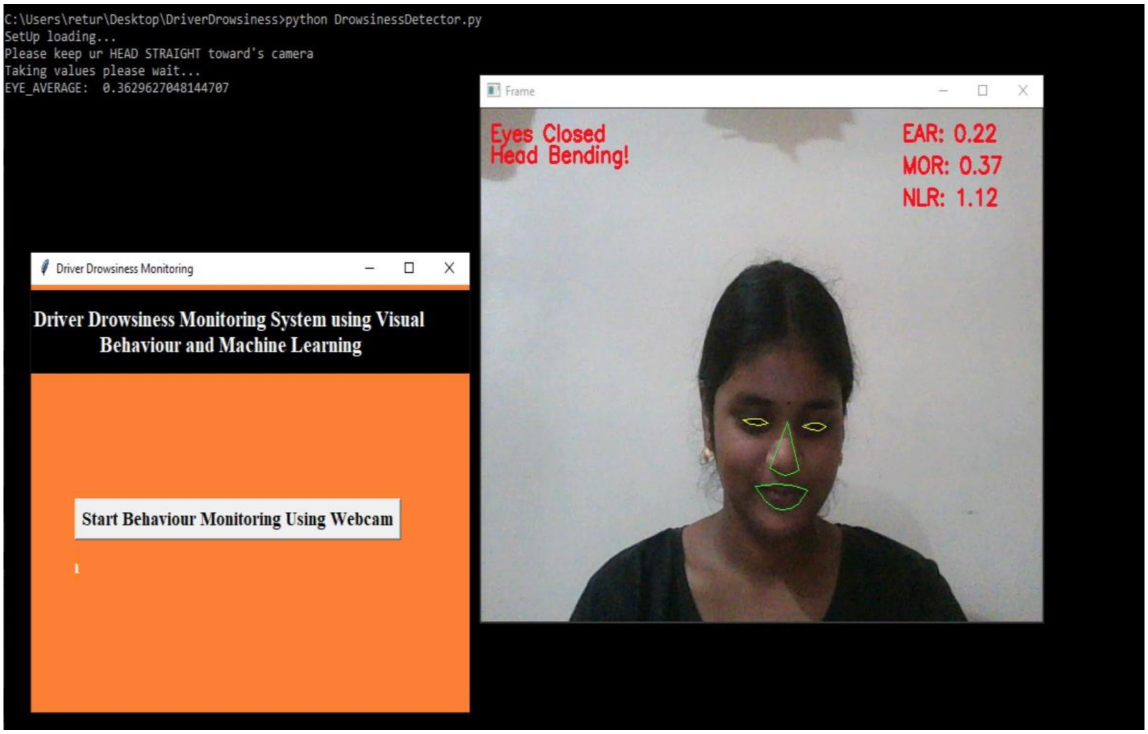

Fig 7.2.2 - Frame where driver bends head for a second but not continuously

8. CONCLUSION

In this project, a low cost, real time driver drowsiness monitoring system has been developed based on visual behavior and machine learning. The system uses the Haar based cascade classifier and HOG-SVM combination object detector for face detection. Here, visual behavior features like eye aspect ratio, mouth opening ratio and nose length ratio are computed from the streaming video, captured by a webcam. An adaptive thresholding technique has been developed to detect driver drowsiness in real time. The developed system works accurately with the generated synthetic data. The presented system performs well under the normal lighting conditions and normal resolutions. The method is non intrusive and hence user friendly. It doesn’t need any special hardware other than a normal web camera. This makes the system suitable to be implemented in desktop computers, mobile devices and so on. This method can be used in wide variety of applications like driver alertness measurement, liveliness detection, concentration measurement,measure of attentiveness etc.

9. 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.

[2] 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.

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

[4] 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.

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

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

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

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

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

[10] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection", IEEE conf. on CVPR, 2005.

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

[12] Richard O. Duda, Peter E. Hart, David G. Stork, "Pattern Classification", Wiley student edition