Drowsiness Detection system

Submitted in partial fulfilment of the requirements

Of the degree of

Bachelor of Engineering

By

MUHAMMAD FAIZ KHAN 14DET84

ASAD KHAN 14DET80

MUSADDIQUE DAKHANI 14DET76

SUPERVISOR

Asst. Prof. Abrar Sayyid

UNIVERSITY OF MUMBAI

ANJUMAN I ISLAM KALSEKAR TECHNICAL CAMPUS

2016-2017

CERTIFICATE

This is to certify that the project entitled "**Drowsiness Detection system"** is the bonafide work carried out by:

MUHAMMAD FAIZ KHAN	14DET84
ASAD KHAN	14DET80
MUSADDIQUE DAKHANI	14DET76

B.E EXTC students of Anjuman-I-Islam Kalsekar Technical Campus, Panvel, during the year 2016-17, in partial fulfilment of the requirements for the Bachelor of engineering in Electronics and telecommunication engineering and is submitted to the Mumbai University. The project report has been approved.

_____	_____	_____
(H.O.D)	**(Examiner)**	**(Guide)**
Prof. Mujib Tamboli		**Asst.Prof. Abrar Sayyid**

Project Report Approval for B.E.

This project report entitled Drowsiness Detection system is approved for the degree of **ELECTRONICS AND TELECOMMUNICATION ENGINEERING.**

EXAMINERS:

1) ----------

2) ----------

SUPERVISOR:

1) ----------

HOD

1) ----------

Date: 24th APRIL 2017.

Place: New Panvel

DECLARATION

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Muhammad Faiz Khan – 14DET84

Asad Khan – 14DET80

Musaddique Dakhani – 14DET76

Date: 24th APRIL 2017

ACKNOWLEDGEMENT

We appreciate the beauty of a rainbow, but never do we think that we need both the sun and the rain to make its colours appear. Similarly, this project work is the fruit of many such unseen hands. It's those small inputs from different people that have lent a helping to our project.

I also take this opportunity to express a deep sense of gratitude to **Prof. Mr. Mujib Tamboli** (HOD of EXTC department) for his cordial support, valuable information and guidance, which helped us in completing this task through various stages.

I take this opportunity to express my profound gratitude and deep regards to our guide **Asst. Prof.Abrar Sayyid** for his exemplary guidance, monitoring and constant encouragement throughout the course of this project work. We also take this opportunity to thank our Lab Assistants for providing access to the lab and support.

I am obliged to the staff member of AIKTC, for the valuable information provided by them in the respective fields. I am grateful for their cooperation during the period of my project work.

ABSTRACT

Various investigations show that driver's drowsiness is one of the main causes of road accidents. The current technology in digital computer system allows researchers around the world to study the fatigue behaviour. The purpose of this study is to detect the drowsiness in drivers to prevent the accidents and to improve the safety on the highways. Real time face detection is implemented to locate driver's face region. In this project the eye blink of the driver is detected. If the drivers eyes remain closed for more than a certain period of time, the driver is said to be drowsy and an alarm is sounded. The programming is done in python language and Open CV using the Haarcascade library for the detection of facial features. In this project we aim to develop drowsiness system.

Table Content

CONTENT **PAGE NO.**

CHAPTER 1
Introduction

1.1 Drowsiness

Drowsy means sleepy and having low energy. Drowsiness is position of near to sleep, a strong desire for sleep. Drowsiness refers to being unable to keep your eyes open, or feeling sleepy or tired. Drowsiness, also called excess sleepiness. Drowsiness may lead to forgetfulness or falling asleep at inappropriate times. It can be accompanied by weakness, lethargy, and lack of mental alertness. Depression, sorrow and stress are also associated with compromised sleep. Now drowsiness of person driving vehicle is very important. It may not be due to some medical disorders but long driving by a tired driver. This may cause drowsiness so there is a need to detect this to avoid miss happening.

1.2 Driver Fatigue and Road Accidents

Driver fatigue sometimes results in road accidents every year. It is not easy to estimate the exact amount of sleep related accidents but research presents that driver fatigue may be a contributing reason in up to 20% in road accidents. These types of accidents are about 50% more expected to result in death or serious hurt. They happen mainly at higher speed impacts. And the driver who has fallen asleep cannot brake. Drowsiness reduces response time which is a serious element of secure driving. It also reduces alertness, vigilance, and concentration so that the capacity to perform attention-based activities i.e. driving is impaired. The speed at which information is processed is also reduced by drowsiness. The quality of decision-making may also be affected. It is clear that drivers are aware when they are feeling sleepy, and so make a conscious decision about whether to continue driving or to stop for a rest. It may be that those who persist in driving underestimate the risk of actually falling asleep while driving. Or it may be that some drivers choose to ignore the risks in the way drivers drink. Crashes caused by tired drivers are most likely to happen on long journeys on monotonous roads, such as motorways, between 2pm and 4pm especially after eating or taking an alcoholic drink, between 2am and 6am, after having less sleep than normal, after drinking alcohol, it driver takes medicines that cause drowsiness and after long working hours or on journeys home after long shifts, especially night shifts.

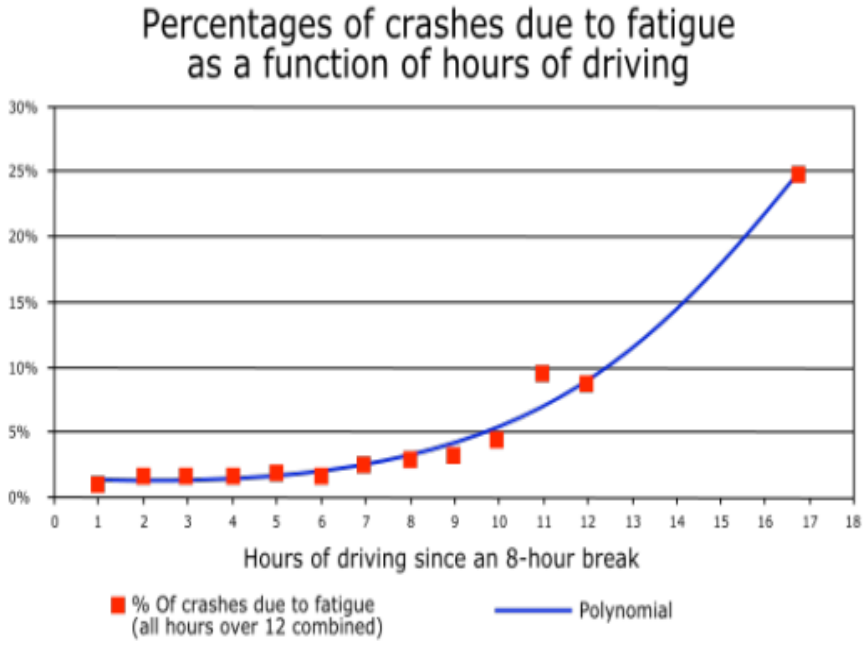


Fig 1.1: Graph between number of hours driven and % of crashes due to drowsy.

Tiredness and fatigue can often affect a person's driving ability long before he/she even notices that he/she is getting tired. Fatigue related crashes are often more severe than others because driver's reaction times are delayed or the drivers have failed to make any manoeuvres to avoid a crash. The number of hours spent driving has a strong correlation to the number of fatigue related accidents.

1.3 Motive of Detection of Problem

Driver drowsiness is a serious hazard in transportation systems. It has been identified as a direct or contributing cause of road accident. Driver drowsiness is one of the major causes of road accident. Drowsiness can seriously slow reaction time, decrease awareness and impair a driver's judgment. It is concluded that driving while drowsy is similar to driving under the influence of alcohol or drugs. In industrialized countries, drowsiness has been estimated to be involved in 2% to 23% of all crashes. Systems that detect when drivers are becoming drowsy and sound a warning promise to be a valuable aid in preventing accidents. Possible techniques

for detecting drowsiness in drivers can be generally divided into the following categories: sensing of physiological characteristics, sensing of driver operation, sensing of vehicle response, monitoring the response of driver. Driver fatigue is a significant factor in a large number of vehicle accidents. Recent statistics estimate that annually 1,200 deaths and 76,000 injuries can be attributed to fatigue related crashes. The development of technologies for detecting or preventing drowsiness at the wheel is a major challenge in the field of accident avoidance systems. Because of the hazard that drowsiness presents on the road, methods need to be developed for counteracting its affects.

1.4 Solution Approach

Two kind of monitoring systems are named as vehicle oriented system and driver oriented system.

1.4.1 Vehicle oriented system

Drowsiness is detected by analyzing the driver's behaving using information measured by sensors located in the vehicle, such as its position on the road, steering wheel movements, pressure on the driving pedals or the variability of the vehicle's speed. The main disadvantages of this approach are that driving behaviour may be very different from driver to driver. This makes it difficult to construct a "correct driving" model that can be used to detect variations in driving behaviour. This model has to be learnt for each driver.

1.4.2 Driver oriented system

There are two types of driver oriented system which are named as-

(A) Instructive monitoring system based on biological indicators

Drowsiness is detected using physiological information. It is measured by sensors located on or around the driver. The physiological information is eye activity, cerebral activity, Yawns, facial expressions, or gaze direction. These systems are more reliable because physiological drowsiness signs are known and are similar to one driver or another driver. There is one drawback with placements of sensors on the driver's body. They may not bother driver while driving. Another drawback is that measurements may be difficult because the driver is constantly moving. This system is bulky to implement.

(B) Non-instructive monitoring system based on face analysis

The human face is dynamic and has a high degree of variability. Face detection is considered to be a difficult problem in computer vision research. As one of the most important features of the human face, human eyes play an important role in face recognition and facial expression analysis. In actuality, the eyes can be considered salient and relatively stable feature on the face in comparison with other facial features. Therefore, when detecting facial features, it is advantageous to detect eyes before the detection of other facial features. The position of other facial features can be estimated using the eye position. In addition, the size, the location and the image-plane rotation of face in the image can be normalized by only the position of both eyes.

1.5 Objective

The objectives of this project are to develop a drowsiness detection system that can detect drowsy or fatigue in drivers to prevent accidents and to improve safety on the roads. This system able accurately monitors the open or closed state of the driver's eye.

CHAPTER 2

LITERATURE REVIEW

Tiesheng Wang et. al., (2005) had developed a system based on yawning detection for determining driver drowsiness. A system had an aim to detect driver drowsiness or fatigue on the base of video analysis which was presented. The main object of this study was on how to extract driver yawning. A real face detector was implemented to trace driver's face region. In this study, mouth window was traced. In which face region and degree of mouth openness was extracted to find driver yawning in video. This method was computationally capable because it ran at real-time on average. When the driver moved his head away by lack of concentration, the eyes and mouth might be occluded and might be detected. There was another situation should be reminded of the driver. For this other methods must be found to deal with it.

Jian-Da Wu et. al., (2007), developed and investigated a warning system while driving using image processing technique with fuzzy logic interface. This system was based on facial images analysis for warning the driver of drowsiness or inattention to prevent traffic accidents. The facial images of driver were taken by a CCD camera which was installed on the dashboard in front of the driver. A fuzzy logic algorithm and an interface were proposed to determine the level of fatigue by measuring the blinding duration and its frequency, and warn the driver accordingly. The experimental works were carried to evaluate the effect of the proposed system for drowsiness warning under various operation conditions. The experimental results indicated that the proposed expect system was effective for increasing safe in drive. This study proved the feasibility of applies image processing technique to safety of vehicle. In this system, besides judging the driver's level of fatigue, it also allowed the head of driver moving within an acceptable region.

Pooneh. R. Tabrizi et. al., (2008) had proposed an easy algorithm for pupil centre and iris boundary localization and a new algorithm for eye state analysis, which there was incorporation into a four step system for drowsiness detection: face detection, eye detection, eye state analysis, and drowsy decision. This new system required no training data at any step or special cameras. Their eye detection algorithm used Eye Map, thus achieving excellent pupil centre and iris boundary localization results on the IMM database. Novel eye state

analysis algorithm detected eye state using the saturation (S) channel of the HSV colour space. Analysis algorithm of eye state using five video sequences and show superior results compared to the common technique based on distance between eyelids. An easy algorithm for pupil centre and iris boundary localization based on Eye Map and a new algorithm for eye state analysis, which they incorporated into a four step system for drowsiness detection: face detection, eye detection, eye state analysis, and drowsy decision by PERCLOS parameter. Pupil centre and iris boundary localization algorithm responded in a wide range of lighting conditions with high accuracy. It also required no training data. For eye state analysis, they proposed a chromatic-based algorithm which had better detection rate for closed eye than the eyelids distance based technique and did not use training data. Proposed system for drowsiness detection was simple, non-intrusive, without the need for training data at any step or special cameras and was safe in comparison with IR illuminators. The main limitation of this system was that it was applicable only when the eyes were visible in the image that means with daylight and without dark sunglasses.

Mandalapu Sarada Devi et. al., (2008) had developed a system that can detect oncoming driver fatigue and issue timely warning could help in preventing many accidents, and consequently save money and reduced personal suffering. The authors had made an attempt to design a system that used video camera that points directly towards the driver's face in order to detect fatigue. If the fatigue was detected a warning signal was issued to alert the driver. The authors had worked on the video files recorded by the camera. Video file was converted into frames. Once the eyes are located from each frame, by measuring the distances between the intensity changes in the eye area one can determine whether the eyes were open or closed. If the eyes were found closed for 5 consecutive frames, the system draws the conclusion that the driver was falling asleep and issued a warning signal. The algorithm was proposed, implemented, tested, and found working satisfactorily. A driver monitoring system was implemented which detected the fatigued state of the driver through continuously monitoring the eyes of the driver. The basis of the method used by authors was the horizontal intensity variation on the face. One similarity among all faces was that eyebrows were significantly different from the skin in intensity, and that the next significant change in intensity, in the y-direction, was the eyes. This facial characteristic was the centre of finding the eyes on the face, which will allow the system to monitor the eyes and detect long periods of eye closure.

Esra Vural et. al., (2007) had employed machine learning to data mine actual human behaviour during drowsiness episodes. Automatic classifiers for 30 facial actions from the facial action coding system were developed using machine learning on a separate database of spontaneous expressions. These facial actions include blinking and yawn motions, as well as a number of other facial movements. Head motion was collected through automatic eye tracking and an accelerometer. These measures were passed to learning-based classifiers such as Ad boost and multinomial ridge regression. The system was able to predict sleep and crash episodes during computer game 96% accuracy within subjects and above 90% accuracy across subjects. The analysis revealed new information about human behaviour during drowsy driving.

K. Dwivedi et. al., (2014) had proposed a vision based intelligent algorithm to detect driver drowsiness. This algorithm made use of features learnt using convolution neural network so as to openly capture various latent facial features and the complex non-linear features interactions. A soft max layer was used to classify the driver as drowsy or non-drowsy. This system was used foe warning the driver of drowsiness or in attention to prevent traffic accidents. This study presents both qualitative and quantitative results.

Hyungseob Han et. al., (2014) had proposed a method of drowsiness detection with eyes open using EEG-based power spectrum analysis. In experiments, all electronic devices were turned off to reduce the artifacts, and a noiseless environment was created to cause drowsiness. After the EEG experiment was complete, drowsy periods are classified according to alpha power spectrum changes that were induced by eyes being closed in a drowsy state. Although the subject's eyes were opened for a long time, drowsiness patterns can be detected. Consequently, detection of drowsiness with eyes open was possible by using EEG-based power spectrum analysis and the proper feature vectors by LPC (linear predictive coding) coefficients.

Hireshi Ueno et. al., (1994) had developed technologies to prevent sleepiness at the wheel. This method for accurately detecting a decline in driver alertness and a method for alerting and refreshing the driver. A system that used image processing technology to analyze images of the driver's face taken with a video camera. Diminished alertness was detected on the basis of the degree to which the driver's eyes are open or closed. This system provided a noncontact technique for judging various levels of driver alertness and facilitates early detection in alertness during driving.

Richard Grace et. al., (1998) had done on efforts performed at the Carnegie Mellon Driving Research Center to develop such in vehicle driver monitoring systems. Commercial motor vehicle truck drivers were studied in actual fleet operations. The drivers operated vehicles that were equipped to measure vehicle performance and driver psycho physiological data. There were two drowsiness detection methods were being considered. First is a video- based system that measures PERCLOS, a scientifically supported measure of drowsiness associated with slow eye closure. The second detection method is based on a model to estimate PERCLOS based on vehicle performance data. A non-parametric i.e., neural network model was used to estimate PERCLOS using measures associated with lane keeping, steering wheel movements and lateral acceleration of the vehicle.

CHAPTER 3

Why OpenCV?

3.1 Eye blink detection using Matlab:-

First we input the facial image using a webcam. Pre-processing was first performed by binarizing the image. The top and sides of the face were detected to narrow down the area where the eyes exist. Using the sides of the face, the center of the face was found which will be used as a reference when computing the left and right eyes. Moving down from the top of the face, horizontal averages of the face area were calculated. Large changes in the averages were used to define the eye area. There was little change in the horizontal average when the eyes were closed which was used to detect a blink.

However Matlab had some serious limitations. The processing capacities required by Matlab were very high. Also there were some problems with speed in real time processing. Matlab was capable of processing only 4-5 frames per second. On a system with a low RAM this was even lower. As we all know an eye blink is a matter of milliseconds. Also a drivers head movements can be pretty fast. Though the Matlab program designed by us detected an eye blink, the performance was found severely wanting.

3.2 What Is OpenCV?

OpenCV [OpenCV] is an open source (see *http://opensource.org*) computer vision library available from *http://SourceForge.net/projects/opencvlibrary*.

OpenCV was designed for computational efficiency and having a high focus on real-time image detection. OpenCV is coded with optimized C and can take work with multicore processors. If we desire more automatic optimization using Intel architectures

[Intel], you can buy Intel's Integrated Performance Primitives (IPP) libraries [IPP]. These consist of low-level routines in various algorithmic areas which are optimized. OpenCV automatically uses the IPP library, at runtime if that library is installed.

One of OpenCVs goals is to provide a simple-to-use computer vision infrastructure which helps people to build highly sophisticated vision applications fast. The OpenCV library, containing over 500 functions, spans many areas in vision. Because computer vision and

machine learning oft en goes hand-in-hand, OpenCV also has a complete, general-purpose, Machine Learning Library (MLL).

This sub library is focused on statistical pattern recognition and clustering. The MLL is very useful for the vision functions that are the basis of OpenCV's usefulness, but is general enough to be used for any machine learning problem.

We have used the Haartraining applications in OpenCV to detect the face and eyes. This creates a classifier given a set of positive and negative samples. The steps were as follows:-

- Gather a data set of face and eye. These should be stored in one or more directories indexed by a text file. A lot of high quality data is required for the classifier to work well.
- The utility application create samples () is used to build a vector output file. Using this file we can repeat the training procedure. It extracts the positive samples from images before normalizing and resizing to specified width and height.
- The Viola Jones cascade decides whether or not the object in an image is similar to the training set. Any image that doesn't contain the object of interest can be turned into negative sample. So in order to learn any object it is required to take a sample of negative background image. All these negative images are put in one file and then it's indexed.
- Training of the image is done using boosting. In training we learn the group of classifiers one at a time. Each classifier in the group is a weak classifier. These weak classifiers are typically composed of a single variable decision tree called stumps. In training the decision stump learns its classification decisions from its data and also learns a weight for its vote from its accuracy on the data. Between training each classifier one by one, the data points are reweighted so that more attention is paid to the data points where errors were made. This process continues until the total error over the dataset arising from the combined weighted vote of the decision trees falls below a certain threshold.

This algorithm is effective when a large number of training data are available.
For our project face and eye classifiers are required. So we used the learning objects method to create our own haarclassifier .xml files. Around 2000 positive and 3000 negative samples

are taken. Training them is a time intensive process. Finally face.xml and haarcascade-eye.xml files are created.

These xml files are directly used for object detection. It detects a sequence of objects (in our case face and eyes). Haarcascade-eye.xml is designed only for open eyes. So when eyes are closed the system doesn't detect anything. This is a blink. When a blink lasts for more than 5 frames, the driver is judged to be drowsy and an alarm is sounded.

3.3 What Is Computer Vision?

Computer vision is the transforming of data from a still, or video camera into either a representation or a new decision. All such transformations are performed to achieve a particular goal. A computer obtains a grid of numbers from a camera or from the disk, and that's that. Usually, there is no built in pattern recognition or automatic control of focus and aperture, no cross-associations with years of experience. For the most part, vision systems are still fairly naïve.

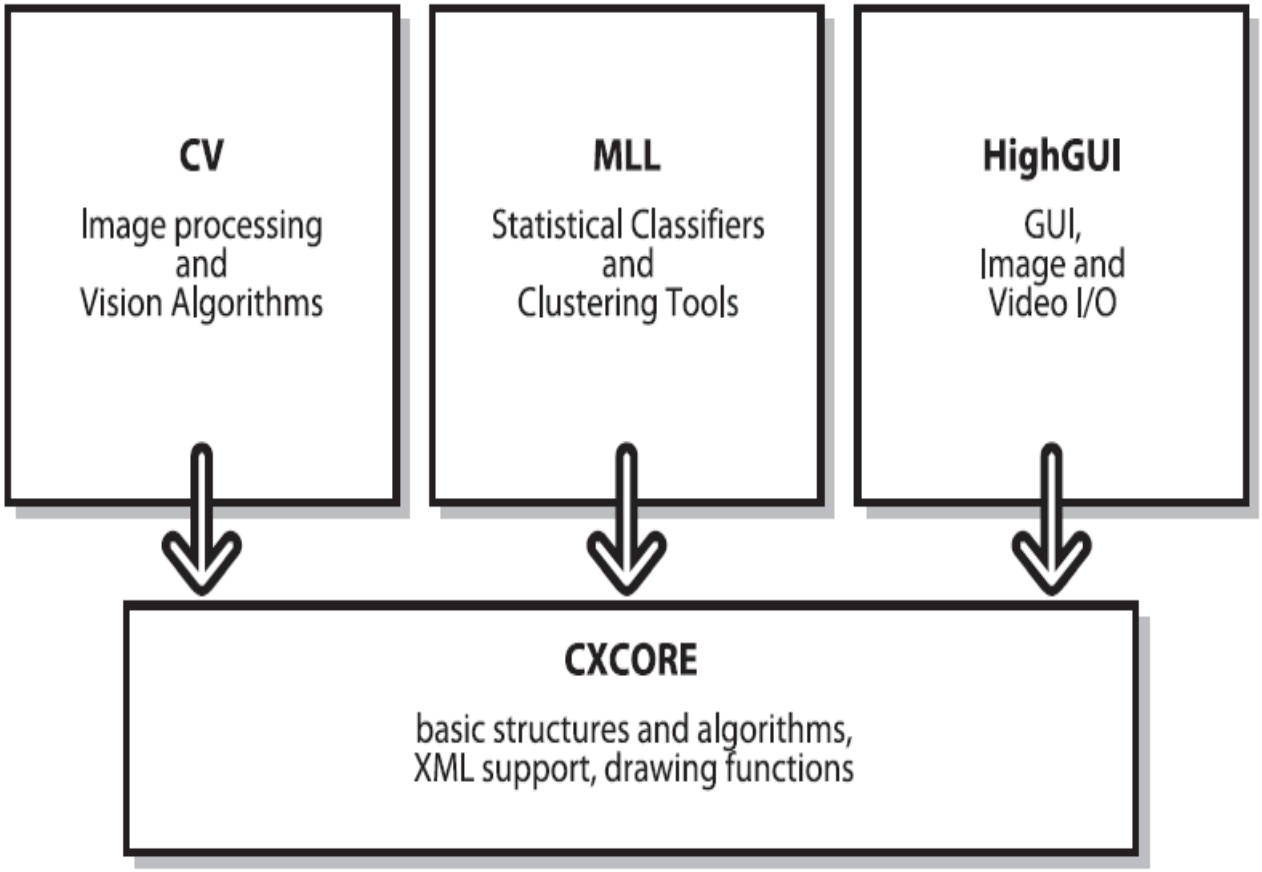
3.4 The Origin of OpenCV

OpenCV came out of an Intel Research initiative meant to advance CPU-intensive applications. Toward this end, Intel launched various projects that included real-time ray tracing and also 3D display walls. One of the programmers working for Intel at the time was visiting universities. He noticed that a few top university groups, like the MIT Media Lab, used to have well-developed as well as internally open computer vision infrastructures—code that was passed from one student to another and which gave each subsequent student a valuable foundation while developing his own vision application. Instead of having to reinvent the basic functions from beginning, a new student may start by adding to that which came before.

3.5 OpenCV Structure and Content

OpenCV can be broadly structured into five primary components, four of which are shown in the figure. The CV component contains mainly the basic image processing and higher-level computer vision algorithms; MLL the machine learning library includes many statistical classifiers as well as clustering tools. High GUI component contains I/O routines with

functions for storing, loading video & images, while CXCore contains all the basic data structures and content.



3.6 Why OpenCV?

Specific

OpenCV was designed for image processing. Every function and data structure has been designed with an Image Processing application in mind. Meanwhile, Matlab, is quite generic. You can get almost everything in the world by means of toolboxes. It may be financial toolboxes or specialized DNA toolboxes.

Efficient

Matlab uses just way too much system resources. With OpenCV, we can get away with as little as 10mb RAM for a real-time application. Although with today's computers, the RAM factor isn't a big thing to be worried about. However, our drowsiness detection system is to be used inside a car in a way that is non-intrusive and small; so a low processing requirement is vital.

Thus we can see how OpenCV is a better choice than Matlab for a real-time drowsiness detection system.

Speedy

Matlab is just way too slow. Matlab itself was built upon Java. Also Java was built upon C. So when we run a Matlab program, our computer gets busy trying to interpret and compile all that complicated Matlab code. Then it is turned into Java, and finally executes the code.

If we use C/C++, we don't waste all that time. We directly provide machine language code to the computer, and it gets executed. So ultimately we get more image processing, and not more interpreting.

After doing some real time image processing with both Matlab and OpenCV, we usually got very low speeds, a maximum of about 4-5 frames being processed per second with Matlab. With OpenCV however, we get actual real time processing at around 30 frames being processed per second.

CHAPTER 4

Python

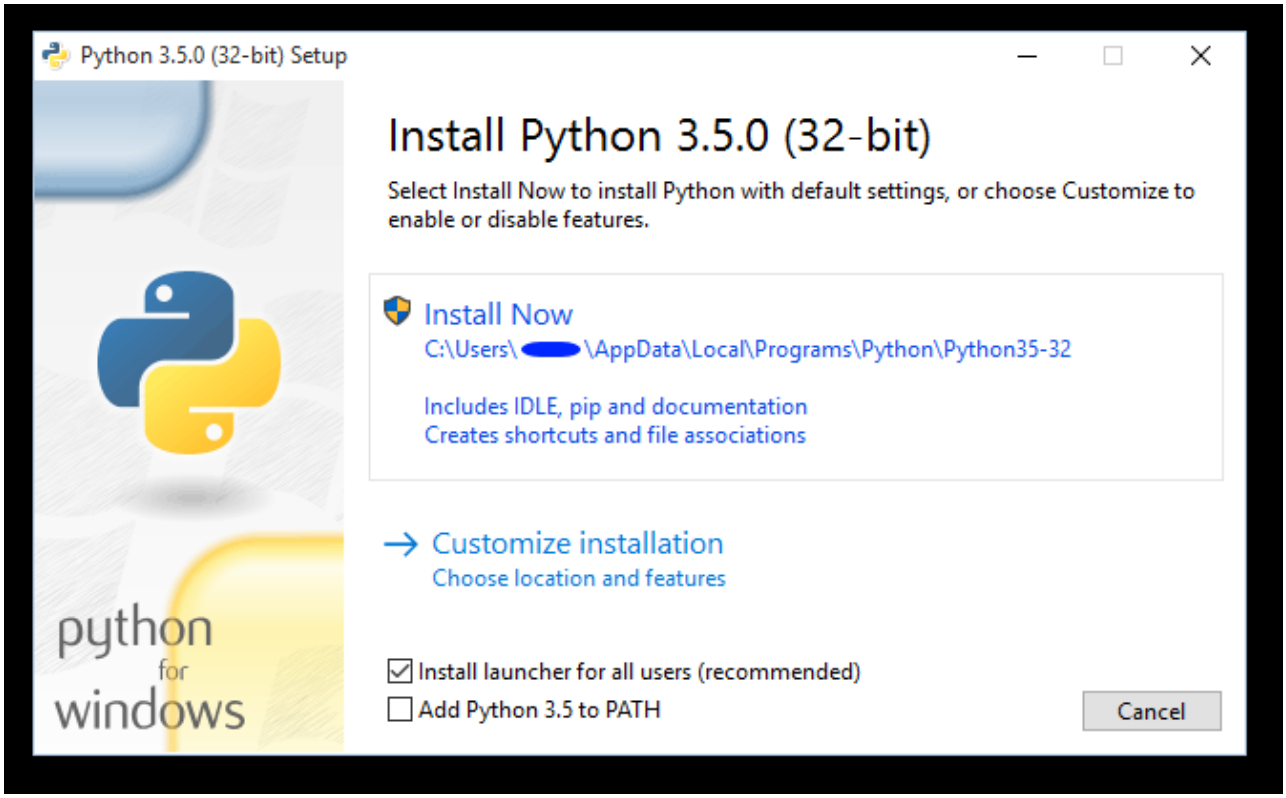
4.1 What is Python?

Python is a powerful modern computer programming language. It bears some similarities to Fortran, one of the earliest programming languages, but it is much more powerful than Fortran. Python allows you to use variables without declaring them (i.e., it determines types implicitly), and it relies on indentation as a control structure. You are not forced to define classes in Python (unlike Java) but you are free to do so when convenient. Python was developed by Guido van Rossum, and it is free software. Free as in "free beer," in that you can obtain Python without spending any money. But Python is also free in other important ways, for example you are free to copy it as many times as you like, and free to study the source code, and make changes to it. There is a worldwide movement behind the idea of free software, initiated in 1983 by Richard Stallman.

This document focuses on learning Python for the purpose of doing mathematical calculations. We assume the reader has some knowledge of basic mathematics, but we try not to assume any previous exposure to computer programming, although some such exposure would certainly be helpful. Python is a good choice for mathematical calculations, since we can write code quickly, test it easily, and its syntax is similar to the way mathematical ideas are expressed in the mathematical literature. By learning Python you will also be learning a major tool used by many web developers.

4.2 Installation and documentation

Mac OS X or Linux, then Python should already be installed on your computer by default. If not, you can download the latest version by visiting the Python home page, at http://www.python.org where you will also find loads of documentation and other useful information. Windows users can also download Python at this website.



4.3 Features of Python

Simple

Python is a simple and minimalistic language. Reading a good Python program feels almost like reading English, although very strict English! This pseudo-code nature of Python is one of its greatest strengths. It allows you to concentrate on the solution to the problem rather than the language itself.

Easy to Learn

As you will see, Python is extremely easy to get started with. Python has an extraordinarily simple syntax, as already mentioned.

Free and Open Source

Python is an example of FLOSS (Free/Libré and Open Source Software). In simple terms, you can freely distribute copies of this software, read its source code, make changes to it, and use pieces of it in new free programs. FLOSS is based on the concept of a community which

shares knowledge. This is one of the reasons why Python is so good. It has been created and is constantly improved by a community who just want to see a better Python.

High-level Language

When you write programs in Python, you never need to bother about the low-level details such as managing the memory used by your program, etc.

Object Oriented

Python supports procedure-oriented programming as well as object-oriented programming. In procedure-oriented languages, the program is built around procedures or functions which are nothing but reusable pieces of programs. In object-oriented languages, the program is built around objects which combine data and functionality. Python has a very powerful but simplistic way of doing OOP, especially when compared to big languages like C++ or Java.

Extensive Libraries

The Python Standard Library is huge indeed. It can help you do various things involving regular expressions, documentation generation, unit testing, threading, databases, web browsers, CGI, FTP, email, XML, XML-RPC, HTML, WAV files, cryptography, GUI (graphical user interfaces), and other system-dependent stuff.

4.4 NumPy

NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.

At the core of the NumPy package, is the *ndarray* object. This encapsulates *n*-dimensional arrays of homogeneous data types, with many operations being performed in compiled code for performance.

There are several important differences between NumPy arrays and the standard Python sequences:

- NumPy arrays have a fixed size at creation, unlike Python lists (which can grow dynamically). Changing the size of an *ndarray*will create a new array and delete the original.
- The elements in a NumPy array are all required to be of the same data type, and thus will be the same size in memory. The exception: one can have arrays of (Python, including NumPy) objects, thereby allowing for arrays of different sized elements.
- NumPy arrays facilitate advanced mathematical and other types of operations on large numbers of data. Typically, such operations are executed more efficiently and with less code than is possible using Python's built-in sequences.
- A growing plethora of scientific and mathematical Python-based packages are using NumPy arrays; though these typically support Python-sequence input, they convert such input to NumPy arrays prior to processing, and they often output NumPy arrays. In other words, in order to efficiently use much (perhaps even most) of today's scientific/mathematical Python-based software, just knowing how to use Python's built-in sequence types is insufficient - one also needs to know how to use NumPy arrays.

4.5 Matplotlib

Matplotlib is a library for making 2D plots of arrays in Python. Although it has its origins in emulating the MATLAB graphics commands, it is independent of MATLAB, and can be used in a Pythonic, object oriented way. Although Matplotlib is written primarily in pure Python, it makes heavy use of NumPy and other extension code to provide good performance even for large arrays. Matplotlib is designed with the philosophy that you should be able to create simple plots with just a few commands, or just one! If you want to see a histogram of your data, you shouldn't need to instantiate objects, call methods, set properties, and so on; it should just work.

For years, I used to use MATLAB exclusively for data analysis and visualization. MATLAB excels at making nice looking plots easy. When I began working with EEG data, I found that I needed to write applications to interact with my data, and developed an EEG analysis application in MATLAB. As the application grew in complexity, interacting with databases, http servers, manipulating complex data structures, I began to strain against the limitations of

MATLAB as a programming language, and decided to start over in Python. Python more than makes up for all of MATLAB's deficiencies as a programming language, but I was having difficulty finding a 2D plotting package.

Matplotlib is used by many people in many different contexts. Some people want to automatically generate PostScript files to send to a printer or publishers. Others deploy Matplotlib on a web application server to generate PNG output for inclusion in dynamically-generated web pages. Some use Matplotlib interactively from the Python shell in Tkinter on Windows.

CHAPTER 5

Machine Learning

5.1 What Is Machine Learning

The goal of **machine learning** is to turn data into information. After having learned from a gathering of data, we want a machine that is able to answer any question about the data:

- What are the other data that are similar to given data? Is there a face in the image?
- What kind of ad will influence the user?

5.2 OpenCV ML Algorithms

The machine learning algorithms that are included in OpenCV are given as follows. All the algorithms are present in the *ML* library apart from Mahalanobis and K-means, which are present in CVCORE, and the algorithm of face detection, which is present in CV.

Mahalanobis: It is a measure of distance that is responsible for the stretchiness of the data. We can divide out the covariance of the given data to find this out. In case of the covariance being the identity matrix (i.e. identical variance), this measure will be identical to the Euclidean distance.

K-means: It is an unsupervised clustering algorithm which signifies a distribution of data w.r.t. *K* centers, *K* being chosen by the coder. The difference between K-means and expectation maximization is that in K-means the centers aren't Gaussian. Also the clusters formed look somewhat like soap bubbles, as centers compete to occupy the closest data points. All these cluster areas are usually used as a form of sparse histogram bin for representing the data.

Normal/Naïve Bayes classifier: It is a generative classifier where features are often assumed to be of Gaussian distribution and also statistically independent from one another. This

assumption is usually false. That's why it's usually known as a ―naïve Bayes‖ classifier. That said, this method usually works surprisingly well.

Decision trees: It is a discriminative classifier. The tree simply finds a single data feature and determines a threshold value of the current node which best divides the data into different classes. The data is broken into parts and the procedure is recursively repeated through the left as well as the right branches of the decision tree. Even if it is not the top performer, it's usually the first thing we try as it is fast and has a very high functionality.

Boosting: It is a discriminative group of classifiers. In boosting, the final classification decision is made by taking into account the combined weighted classification decisions of the group of classifiers. We learn in training the group of classifiers one after the other. Each classifier present in the group is called a weak classifier. These weak classifiers are usually composed of single-variable decision trees known as stumps. Learning its classification decisions from the given data and also learning a weight for its vote based on its accuracy on the data are things the decision tree learns during training. While each classifier is trained one after the other, the data points are re-weighted to make more attention be paid to the data points in which errors were made. This continues until the net error over the entire data set, obtained from the combined weighted vote of all the decision trees present, falls below a certain threshold. This algorithm is usually effective when a very large quantity of training data is available.

Random trees: It is a discriminative forest of a lot of decision trees, each of which is built down to a maximal splitting depth. At the time of learning, every node of every tree is allowed a choice 20 of splitting variables, but only from a randomly generated subset of all the data features. This ensures that all the trees become statistically independent and a decision maker. In the run mode, all the trees get an unweighted vote. Random trees are usually quite effective. They can also perform regression by taking the average of the output numbers from every tree.

Face detector (Haar classifier): It is an object detection application. It is based on a smart use of boosting. A trained frontal face detector is available with the OpenCV distribution. This works remarkably well. We can train the algorithm for other objects by using the software provided. This works wonderfully for rigid objects with characteristic views.

Expectation maximization (EM): It is used for clustering. It is a generative unsupervised algorithm it fits *N* multidimensional Gaussians to the data, N being chosen by the user. It can act as an efficient way for representing a more complex distribution using only a few parameters (i.e. means and variances). Usually used in segmentation, it can be compared with K-means.

5.3 Variable Importance

This is the importance of a particular variable in a dataset. One of the uses of variable importance is for reducing the number of features the classifier needs to consider. After starting with a number of features, we train our classifier to find the importance of each feature in relation to all the other features. We then get rid of unimportant features. Eliminating unimportant features helps in improving speed performance (since it can eliminate the processing taken for computing those features) and also makes training and testing faster. When we don't have sufficient data, which is regularly the case, then eliminating unimportant variables helps in increasing classification accuracy, which in turn yields faster processing and better results.

Breiman's algorithm for variable importance is as follows

1. A classifier is trained on the training set.

2. A validation or test set is used for determining the accuracy of the classifier.

3. For each data point and a chosen feature, a new value for that feature is randomly chosen from among the values that the feature has in the remainder of the data set (known as ―sampling with replacement). This helps in ensuring that the distribution of that feature remains the same as in the original data set, however, the actual structure or meaning of that feature is removed (as its value is chosen at random from the remainder of the data).

4. The classifier is trained on the altered set of the training data and then the accuracy of classification measured on the changed test or validation data set. When randomizing of a feature hurt accuracy a lot, then it is implied that the feature is vital. When randomizing of a feature does not hurt accuracy much, then the feature is of little importance and is a suitable candidate for removal.

5. The original test or validation data set is restored and the next feature is tried until we are finished. The result obtained orders each feature by its importance. This procedure used above is built into random trees and decision trees and boosting. Thus, we can use these algorithms to decide which variables we will finally use as features; then we can use the optimized feature vectors to train the classifier.

5.4 Object detection:-

The Viola-Jones cascade is a binary classifier: It simply decides whether or not the object in an image is similar to the training set. Any image that doesn't contain the object of interest can be turned into a negative sample. It is useful to take the negative images from the same type of data. That is, if we want to learn faces in online videos, for best results we should take our negative samples from comparable frames .However, comparable results can still be obtained using negative samples taken from any other source. Again we put the images into one or more directories and then make an index file consisting of a list of image filenames, one per line. For example, an image index file called background.idx.

5.5 Creation of .xml files:-

For our project face and eye classifiers are required. So we used the learning objects method to create our own haarclassifier .xml files. The Haarcascade files are one for the face and one for the eyes.

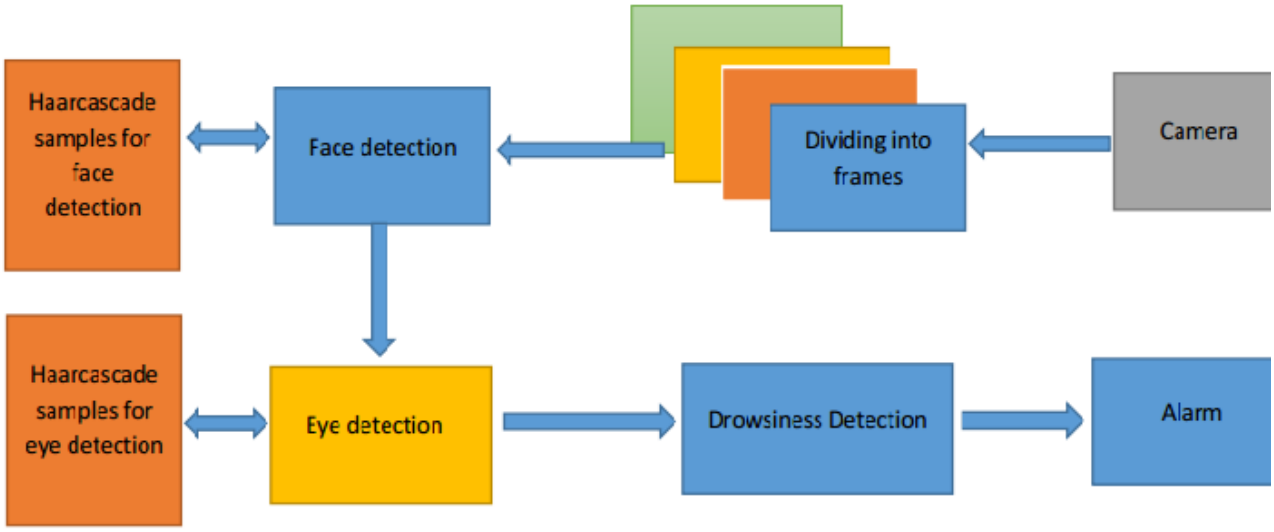
Around 1000 positive and 1500 negative samples were taken. It took a long time to train them. Finally face.xml and Haarcascade_eye.xml files are created.

These xml files are directly used for object detection using haardetectobjects() function. It detects a sequence of objects (in our case our face and eyes). Haarcascade-eye.xml is designed only for open eyes. So when eyes are closed the system doesn't detect anything. This is a blink. When a blink lasts for more than 5 frames, the driver is judged to be drowsy and an alarm is sounded.

Chapter 6

Algorithm and Implementation

6.1 Block Diagram



Drowsiness of a person can be measured by the extended period of time for which his/her eyes are in closed state. In our system, primary attention is given to the faster detection and processing of data. The number of frames for which eyes are closed is monitored. If the number of frames exceeds a certain value, then a warning message is generated on the display showing that the driver is feeling drowsy.

In our algorithm, first the image is acquired by the webcam for processing. Then we use the Haarcascade file face.xml to search and detect the faces in each individual frame. If no face is detected then another frame is acquired. If a face is detected, then a region of interest in marked within the face. This region of interest contains the eyes. Defining a region of interest significantly reduces the computational requirements of the system. After that the eyes are detected from the region of interest by using Haarcascade_eye.xml.

6.2 Image acquisition:-

The function cvCaptureFromCAM allocates and initialized the CvCapture structure for reading a video stream from the camera.

CvCapture* cvCaptureFromCAM(int index);

Index of the camera to be used. If there is only one camera or it does not matter what camera to use , -1 may be passed.

cvSetCaptureProperty Sets camera properties For example

cvSetCaptureProperty(capture, CV_CAP_PROP_FRAME_WIDTH, 280);

cvSetCaptureProperty(capture, CV_CAP_PROP_FRAME_HEIGHT, 220);

The function cvQueryFrame() grabs a frame from a camera or video file,

The function cvQueryFrame() grabs a frame from a camera or video file, decompresses it and returns it. This function is just a combination of *GrabFrame* and *RetrieveFrame*, but in one call. The returned image should not be released or modified by the user. In the event of an error, the return value may be NULL.

6.3 Dividing into Frames:

We are dealing with real time situation where video is recorded and has to be processed. But the processing or application of algorithm can be done only on an image. Hence the captured video has to be divided into frames for analysing.

.

6.4 Face detection:-

In this stage we detect the region containing the face of the driver. A specified algorithm is for detection of face in every frame. By face detection we means that locating the face in a frame or in other words finding location of facial characters through a type of technology with the use of computer. The frame may be any random frame. Only facial related structures or features are detected and all others types of objects like buildings, tree, bodies are ignored.

We know that face is also a type of object. So we can consider detection of face as a particular case of object detection. In this type of object type of class detection, we try to know where the objects in the interest image are located and what is their size which may belongs to a particular class. The work of algorithm that is made for face detection is mostly concentrated on finding the front side of the face. But the algorithms that are developed recently focus on more general cases. For our case it may be face in the tilted position or any other portion of the faces and also it finds the possibility of multiple faces. Which means the rotation axis with respect to the present

observer from the reference of face in a particular? Or even if there is vertical rotation plane then also it is able to solve the purpose. In new type of algorithm it is considered that the picture or video is a variable which means that different condition in them like hue contrast may change its variance. The amount of light may also affect. Also the position of the input may vary the output. Many calculations actualize the face-detection assignment as a two way pattern-differentiation task.

Cascade is loaded by:

```
cascade = (CvHaarClassifierCascade*)cvLoad( cascade_name,
0, 0, 0 );
```

storage is allocated:

```
CvMemStorage* storage = cvCreateMemStorage(0);
CvSeq* cvHaarDetectObjects(
const CvArr* image,
CvHaarClassifierCascade* cascade,
CvMemStorage* storage,
double scale_factor = 1.1,
int min_neighbors = 3,
int flags = 0,
CvSize min_size = cvSize(40,40)
);
cvRectangle(
img,
cvPoint(face->x, face->y),
cvPoint(
face->x + face->width,
face->y + face->height
),
CV_RGB(0, 255, 0),
1, 8, 0
)
```

The above function draws a rectangle in the image for the corresponding corner points .the other parameter are for drawing colour and thickness and type of lines in the rectangle.

6.5 Set Image Region of interest:-

```
cvSetImageROI(
img, /* the source image */
cvRect(
face->x, /* x = start from leftmost */
face->y + (face->height)/5, /* y = a few pixels from the top */
face->width, /* width = same width with the face */
(face->height)/3 /* height = 1/2 of face height */
)
```

It sets the ROI for the corresponding image. We have taken from the left most point in the face to a few pixels from the top to half of face height. Width of the region is same as that of the face. The rectangular region is now used to get eyes.

6.6 Recognition of face region

To detect the region of face after minimizing the background extra portion from the image, labelling method is used. In the labelling method, components are connected in 2-D binary image. This function will return a matrix of the same size as binary image. It contains labels for the connected objects in binary image. it can have a value of either 4 where 4 specifies 4-connected objects or 8 where 8 specifies 8-connected objects. If the argument is omitted, it defaults to 8. The elements of matrix are integer values greater than or equal to 0. The pixels labeled 0 are the background. The pixels labelled 1 make up one object; the pixels labelled 2 make up a second object and so forth.

After that it is required to measure the properties of image regions. The region properties function is used. This function measures a set of properties for each labelled region in the label matrix. Positive integer elements of matrix correspond to different regions. For example, the set of elements of matrix equal to 1 corresponds to region 1; the set of elements of matrix equal to 2 corresponds to region 2; and so on. The return value is a structure array of length matrix. The fields of the structure array denote different measurements for each region, as specified by properties. Properties can be a comma-separated list of strings, a cell array containing strings, the single string 'all', or the string 'basic'. There are set of valid property strings. Property strings are case insensitive. This rectangle box is used to store different operated images. And these operated images are three kind of. First operated image which is stored as matrix in rectangle box is

cropped cb-cr image. Second image is gray image which is converted from colour image. Third image is histogram equalization image which is stored as matrix in rectangle box.

6.7 Eye Detection:-

In our method eye is the decision parameter for finding the state of driver. Though detection of eye may be easier to locate, but it's really quite complicated. At this point it performs the detection of eye in the required particular region with the use of detection of several features. Generally Eigen approach is used for this process. It is a time taking process. When eye detection is done then the result is matched with the reference or threshold value for deciding the state of the driver.

Poor contrast of eyes generally creates lots of problems in its detection. After successful detection of face eye needs to be detected for further processing. In our method eye is the decision parameter for finding the state of driver. Though detection of eye does not look complex but the actual process is quite hectic. In this case it performs the detection of eye in the specified region with the use of feature detection. Generally Eigen approach is used for this process. It is a time taking process. When eye detection is done then the result is matched with the reference or threshold value for deciding the state of the driver.

6.8 Digital Image Processing

Digital image processing is the use of computer algorithms to perform image processing on digital images. In the field of digital signal processing, there are many advantages of digital image processing with compare to analog image processing. There is a much wider range of algorithms to be applied to the input data. It can avoid problems like the build-up of noise and signal deformation during processing. Since images are defined over two dimensions, even applied in the case of more dimensions also, digital image processing may be modelled in the form of multidimensional systems.

Digital image processing permits the use of much more difficult algorithms. Digital image processing can offer both more complicated performance at simple tasks, and the implementation of methods which would be impossible by analog means. Digital image processing is the only practical technology for classification, pattern recognition, projection, feature extraction and multi-scale signal analysis.

Some techniques which are used in digital image processing names as pixlation, principal components analysis, linear filtering, anisotropic diffusion, wavelets, neural networks, independent component analysis, hidden markov model, self-organizing maps and partial differential equation. There is necessary to discuss on Digital Image processing because throughout whole methodology this technique only is used.

An image may be defined as a two-dimensional function, f(x,y), where x and y are spatial coordinates, and the amplitude of f at any pair of coordinates (x,y) is called the intensity or gray level of the image at that point. When x.y, and the amplitude values of f are all finite, discrete quantities, the image is called as a digital image. Digital image processing refers to processing digital images using a digital computer. A digital image is composed of a finite number of elements, each of which has a particular location and value. These elements are referred to as picture elements, image elements, pels, and pixels. Pixel is the term used most widely to denote the elements of a digital image.

Vision is the most advanced of among all senses, so it is obvious that images participate the single most vital role in human perception. But unlike humans, who are limited to the visual band of electromagnetic spectrum, imaging machines cover almost the entire electromagnetic spectrum. It has range from gamma to radio waves. They can operate also on images generated by sources that humans do not usually relate with images. These include electron microscopy, ultrasound, and computer-generated images. So, it is easily seen that the digital image processing encompasses a large, huge and wide varied field of applications.

CHAPTER 7

RESULT

7.1 Summary:-

To obtain the result a large number of videos were taken and their accuracy in determining eye blinks and drowsiness was tested.

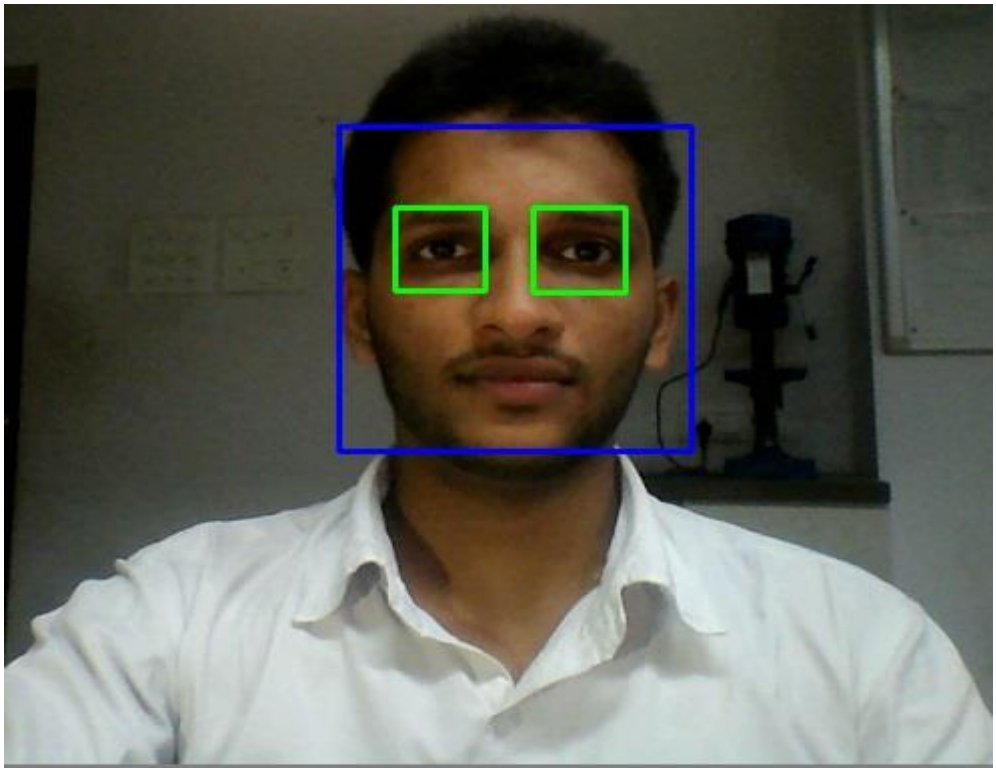
For this project we used a 0.3 megapixel webcam connected to the computer. The webcam required white LEDs attached to it for providing better illumination. In real time scenario, infrared LEDs should be used instead of white LEDs so that the system is non-intrusive. Buzzer is used to produce alert sound output in order to wake up the driver when drowsiness exceeds a certain threshold.

The system was tested for different people in different ambient lighting conditions (daytime and night-time). When the webcam backlight was turned ON and the face is kept at an optimum distance, then the system is able to detect blinks as well as drowsiness with more than 90% accuracy. This is a good result and can be implemented in real-time systems as well.

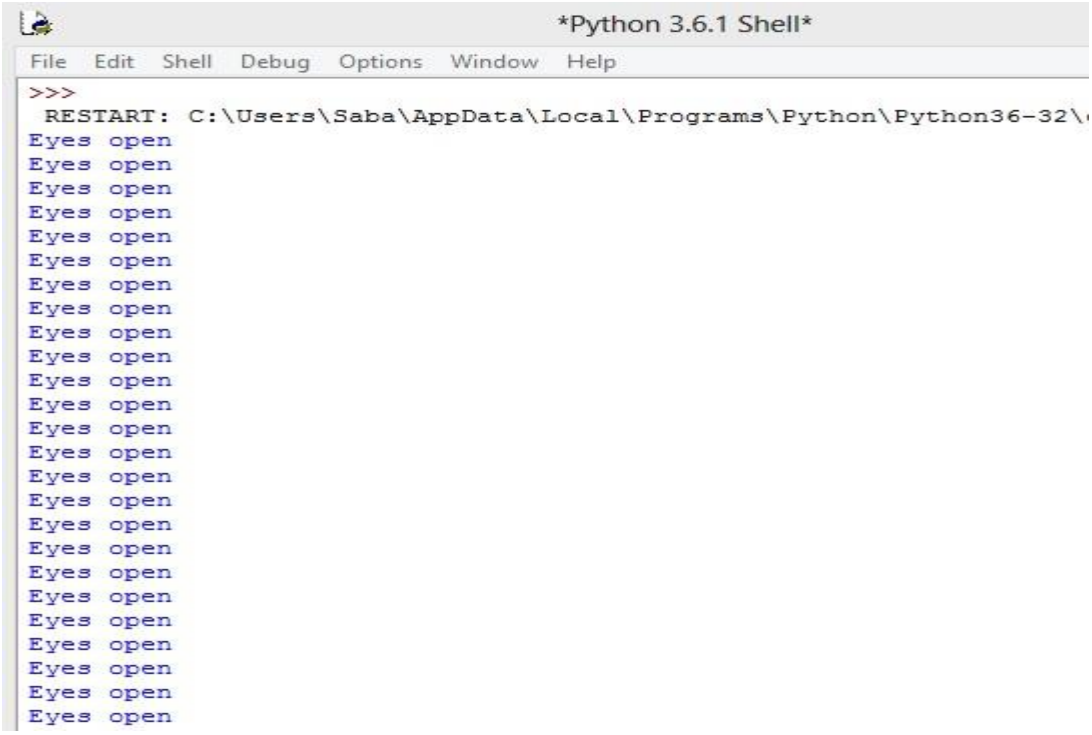
Sample outputs for various conditions in various images are given below. Videos were taken; in which both face and eyes were detected. Though the two processes have relatively equal accuracy.

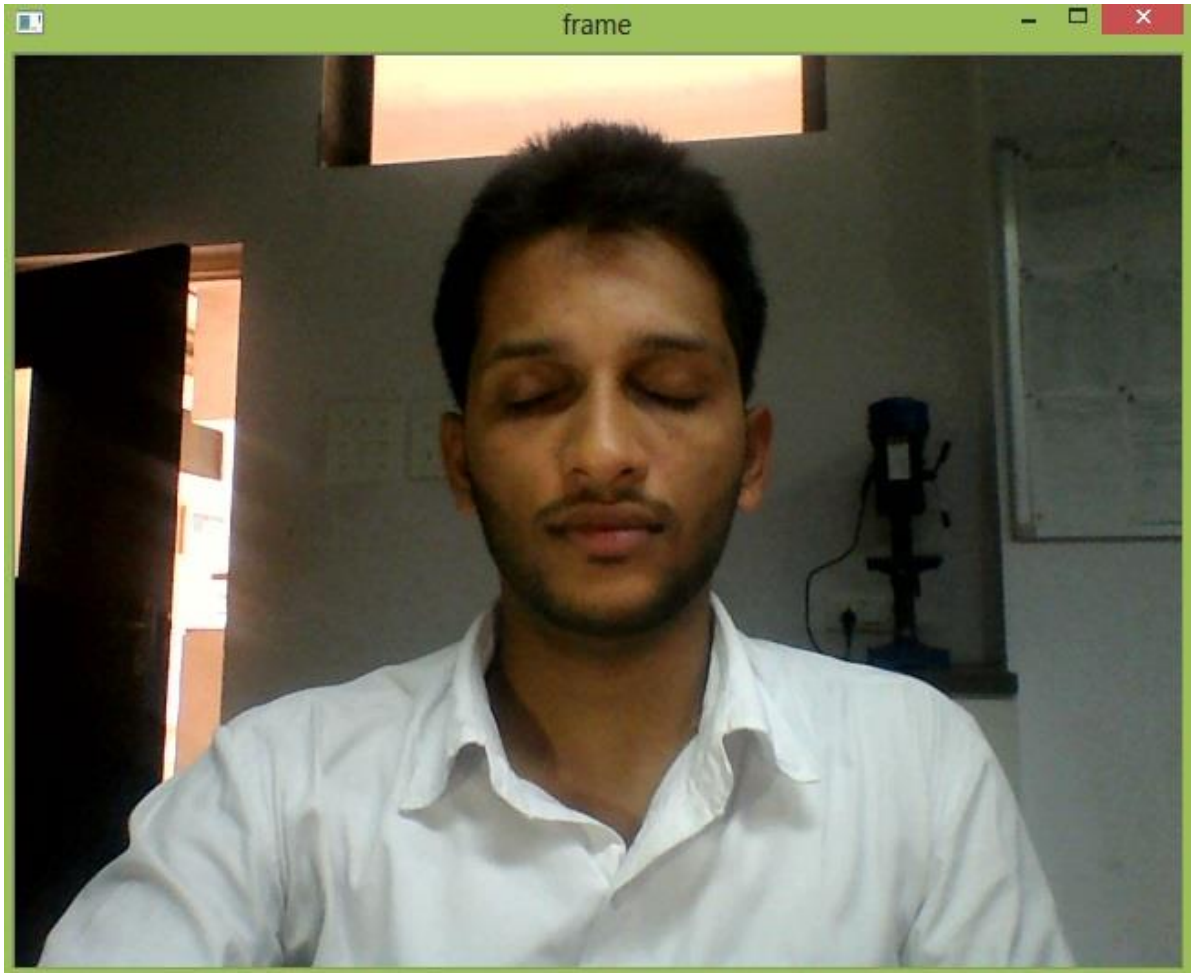
7.2 **Sample Images:-**

Sample 1



Face and eyes detected

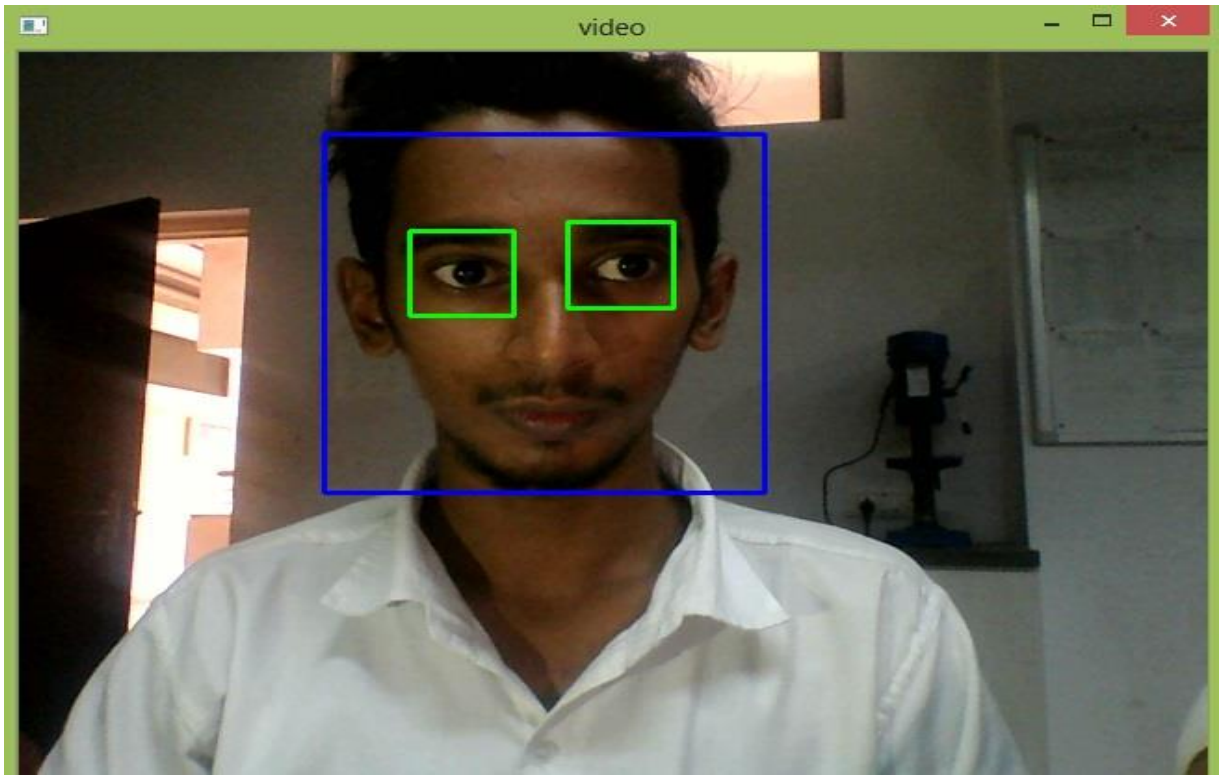




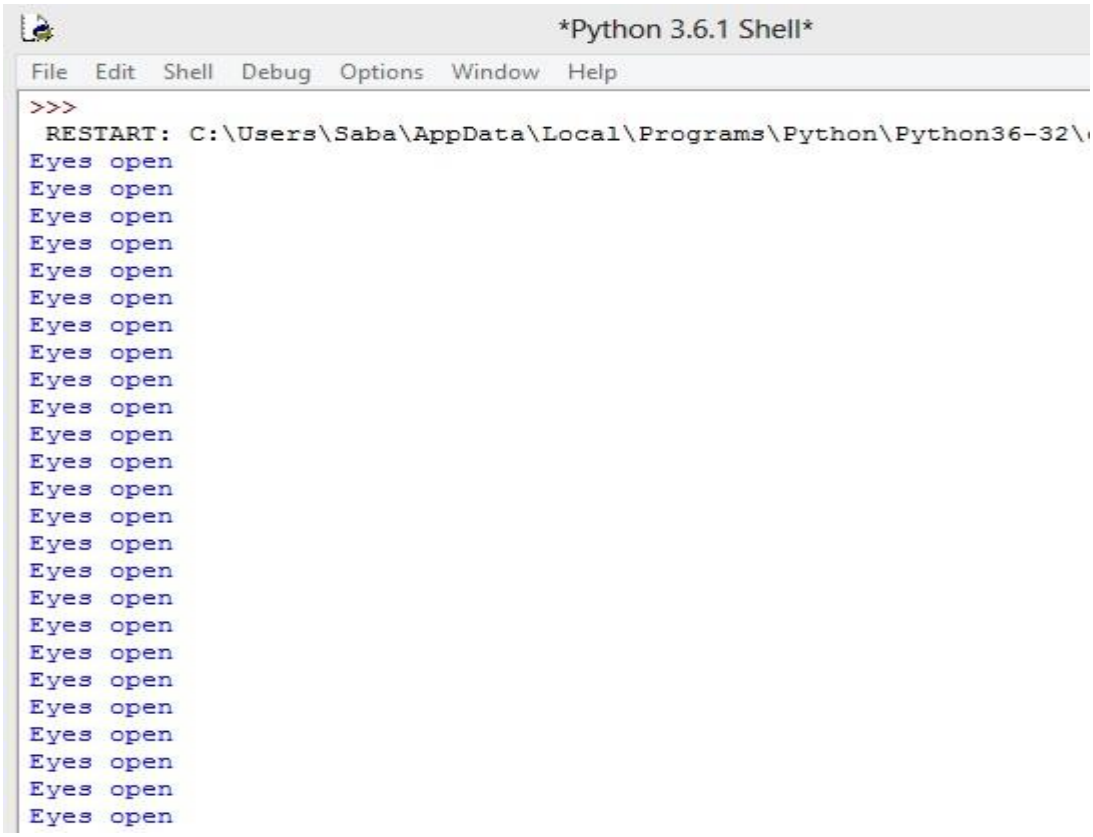
Drowsiness Detected!

```
Eyes closed
Eyes closed
Eyes closed
Eyes closed
Eyes closed
Eyes closed
Eyes closed
Eyes closed
Eyes closed
Eyes closed
Drowsiness Detected!!!
```

Sample 2



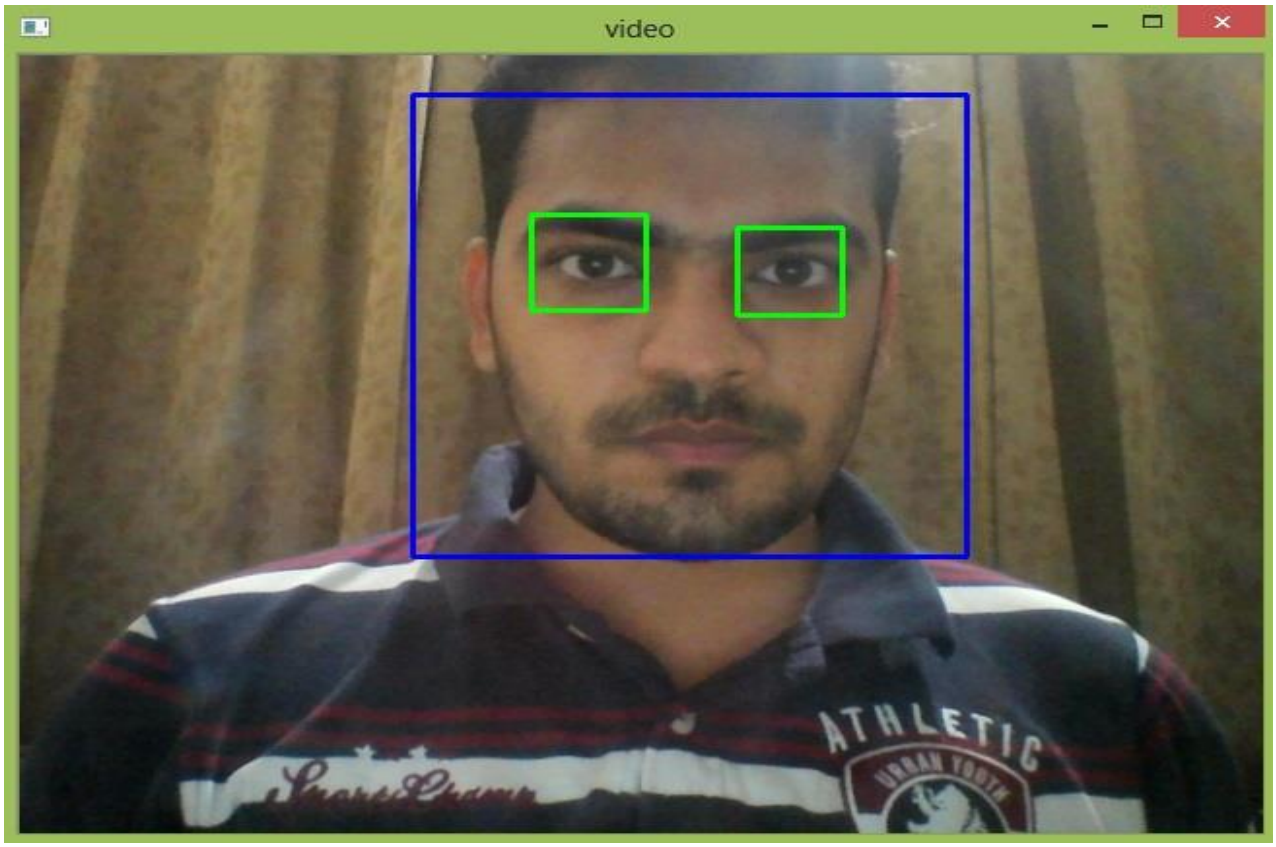
Face and eyes detected



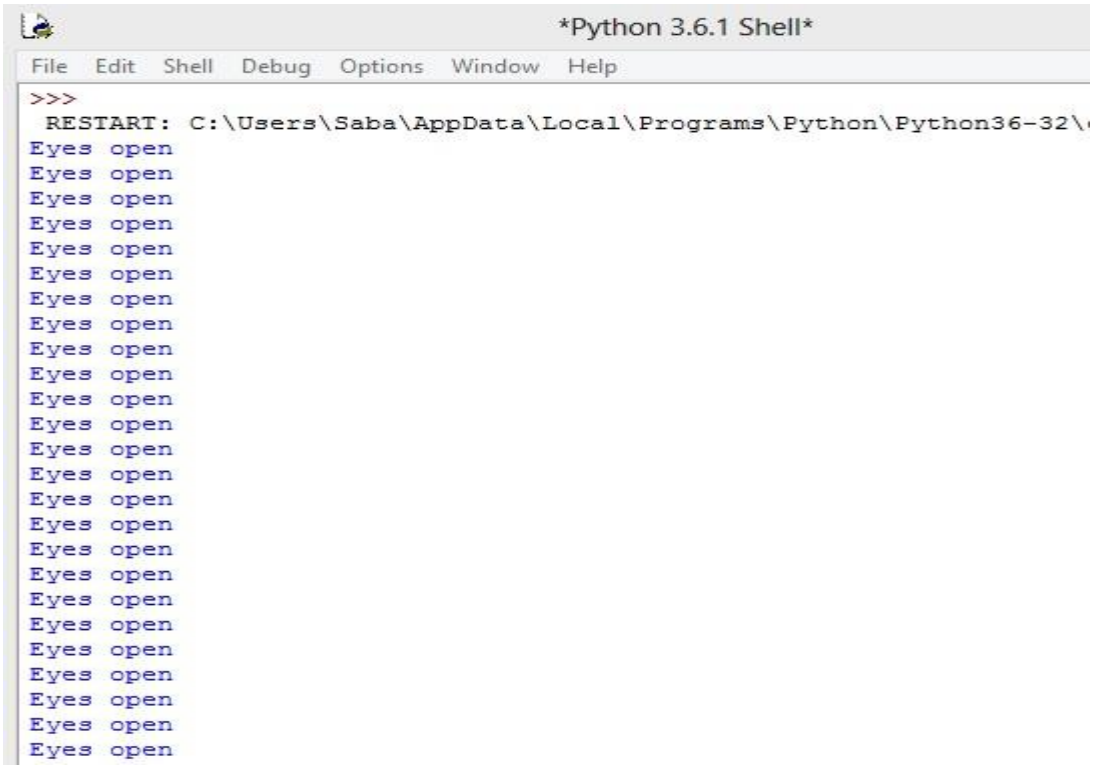
Drowsiness detected!

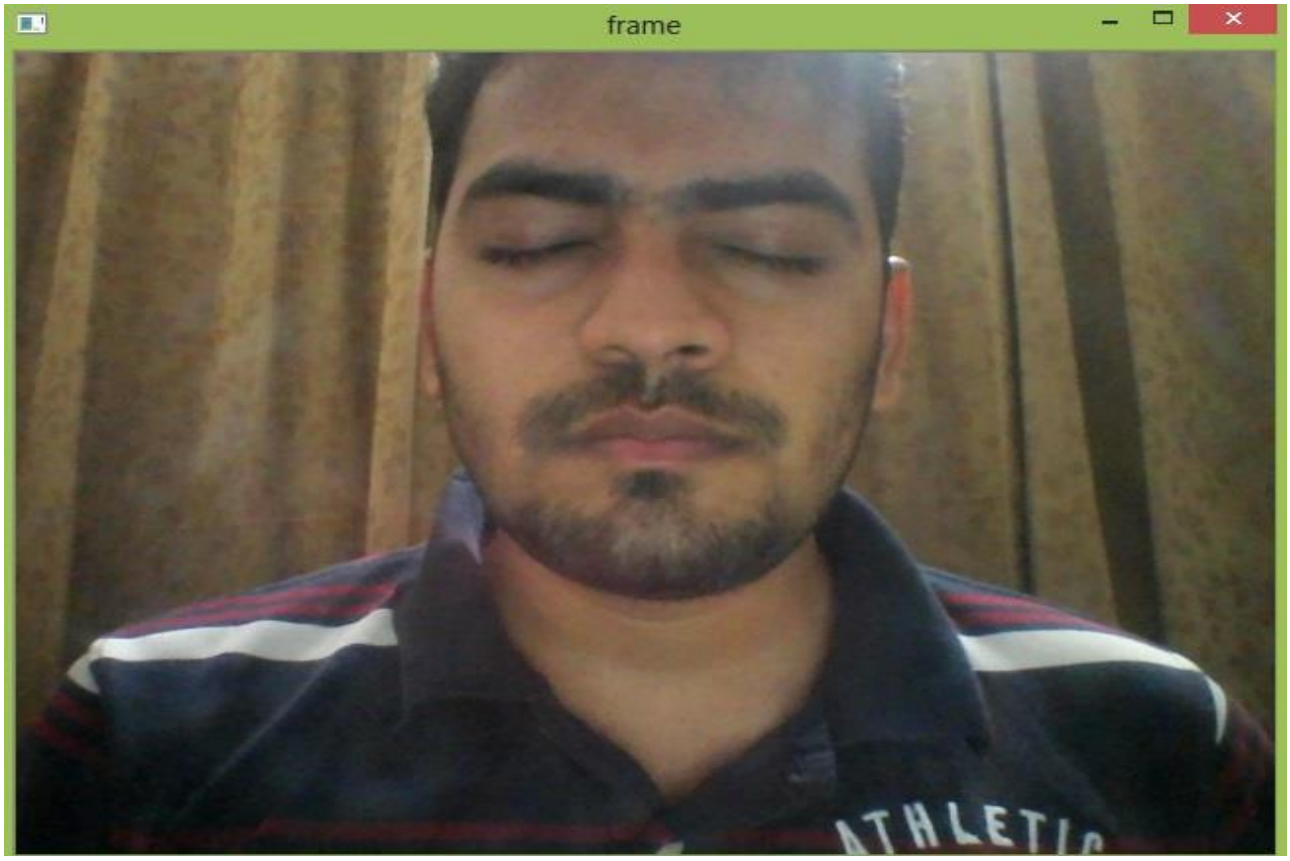
```
Eyes open
Eyes closed
Eyes closed
Eyes closed
Eyes closed
Eyes closed
Eyes closed
Eyes closed
Eyes closed
Eyes closed
Eyes closed
Drowsiness Detected!!!
```

Sample 3



Face and eyes detection





Drowsiness detected!

```
Eyes closed
Eyes closed
Eyes closed
Eyes closed
Eyes closed
Eyes closed
Eyes closed
Eyes closed
Eyes closed
Eyes closed
Drowsiness Detected!!!
```

7.3 Limitations:-

The limitations of the system are as follows.

1. Dependence on ambient light:-

With poor lighting conditions even though face is easily detected, sometimes the system is unable to detect the eyes. So it gives an erroneous result which must be taken care of. In real time scenario infrared backlights should be used to avoid poor lighting conditions.

2. Optimum range required:-

When the distance between face and webcam is not at optimum range then certain problems are arising.

When face is away from the webcam (more than 70cm) then the backlight is insufficient to illuminate the face properly. So eyes are not detected with high accuracy which shows error in detection of drowsiness.

This issue is not seriously taken into account as in real time scenario the distance between drivers face and webcam doesn't exceed 50cm. so the problem never arises.

Considering the above difficulties, the optimum distance range for drowsiness detection is set to 40-70 cm.

3. Hardware requirements:-

Our system was run in a PC with a configuration of 2.2GHz and 2GB RAM Pentium dual core processor.

Though the system runs fine on higher configurations, when a system has an inferior configuration, the system may not be smooth and drowsiness detection will be slow.

4. Delay in sounding alarm:-

When drowsiness level exceeds a certain threshold, an alarm is produced by a system speaker. There is a significant delay between when drowsiness is detected and when system generates the alarm. But in real time, drowsiness is a continuous phenomenon rather than a one off occurrence. So the delay is not that problematic.

5. Poor detection with spectacles:-

When the driver wears glasses the system fails to detect eyes which is the most significant drawback of our system. This issue has not yet been resolved and is a challenge for almost all eye detection systems designed so far.

6. Problem with multiple faces:-

If more than one face is detected by the webcam, then our system gives an erroneous result. This problem is not important as we want to detect the drowsiness of a single driver.

7.4 Future works:-

In the real time driver fatigue detection system it is required to slow down a vehicle automatically when fatigue level crosses a certain limit. Instead of threshold drowsiness level it is suggested to design a continuous scale driver fatigue detection system. It monitors the level of drowsiness continuously and when this level exceeds a certain value a signal is generated which controls the hydraulic braking system of the vehicle.

Hardware components required:-

1. Dedicated hardware for image acquisition processing and display.
2. Interface support with the hydraulic braking system which includes relay, timer, stepper motor and a linear actuator.

Function:-

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

The linear actuator converts rotational movement of stepper motor to linear motion. This linear motion is used to drive a shaft which is directly connected to the hydraulic braking system of the vehicle. When the shaft moves it applies the brake and the vehicle speed decreases.

Currently there is not adjustment in zoom or direction of the camera during operation. Future work may be to automatically zoom in on the eyes once they are localized. This would avoid the trade-off between having a wide field of view in order to locate the eyes, and a narrow view in order to detect fatigue.

This system only looks at the number of consecutive frames where the eyes are closed. At that point it may be too late to issue the warning. By studying eye movement patterns, it is possible to find a method to generate the warning sooner. Using 3D images is another possibility in finding the eyes. The eyes are the deepest part of a 3D image, and this maybe a more robust way of localizing the eyes. Adaptive binarization is an addition that can help make the system more robust. This may also eliminate the need for the noise removal function, cutting down the computations needed to find the eyes. This will also allow adaptability to changes in ambient light. The system does not work for dark skinned individuals. This can be corrected by having an adaptive light source. The adaptive light source would measure the amount of light being reflected back. If little light is being reflected, the intensity of the light is increased. Darker skinned individual need much more light, so that when the binary image is constructed, the face is white, and the background is black.

7.5 Conclusion:-

This is the study that uses a large set of spontaneous facial expressions for the detection of drowsiness. Previous approaches to drowsiness detection primarily make pre-assumptions about the relevant behaviour, focusing on blink rate, eye closure, and yawning. Here there is implementation of learning methods to determine actual human behaviour during drowsiness episodes. This study reveals that facial expressions are very reliable indicators of driver drowsiness and facial expressions can be used to do fine discrimination in the different levels of drowsiness and reliably predict the time to crash. In laboratory conditions computer vision expression recognition systems can be used to reliably detect drowsiness and predict crash with high reliability. Field studies are needed to evaluate the performance of these systems in actual driving environments. Spontaneous facial expressions under drowsiness are very different from posed expressions of drowsiness. A non-invasive system is able to localize the eyes and monitor fatigue was developed. During the monitoring, the system is able to decide if the eyes are opened or closed.

Thus we have successfully designed a drowsiness detection system using OpenCV software and Haar Classifiers. The system so developed was successfully tested,

REFERENCES

[1] Jian-Da Wu, Tuo Rung Chen, “Development of a drowsiness warning system based on the fuzzy logic images analysis”, Elsevier, Expert System with Applications, vol.34, pp.1556-1561, 2008

[2] Vijayalaxmi, D.Elizabeth Rani, “Eye state Detection Using Image Processing Technique”,ajer, vol.04, pp.44-48, 2015

[3] Nidhi Sharma, Prof. V.K. Banga, “Development of a Drowsiness Warning System based on the fuzzy logic”, International journal of Computer Applications (0975-8887) vol.8, 2010

[4] Itenderpal Singh, V.K. Banga, “DevelopmentOf A Drowsiness Warning System Using Neural Networks”, International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, vol.2, pp.36143623, August 2013

[5] K. Dwivedi, K. Biswaranjan, A. Sethi, “Drowsy driver detection using representation learning”, IEEE, International Advance Computing Conference, pp.995-999, Feb-2014

[6] Willem B. Verwey, David M. Zaidel, “Preventing drowsiness accidents by an alertness maintenance device”, Elsevier, Accident Analysis and Prevention, vol.31, pp.199-211, 1999