**INTRODUCTION**

1.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.

2. The aim of this project is to develop a prototype drowsiness detection system. The focus will be placed on designing a system that will accurately monitor the open or closed state of the driver’s eyes in real-time. By monitoring the eyes, it is believed that the symptoms of driver fatigue can be detected early enough to avoid a car accident. Detection of fatigue involves the observation of eye movements and blink patterns in a sequence of images of a face.

3.Initially, we decided to go about detecting eye blink patterns using python. The procedure used was the geometric manipulation of intensity levels. The algorithm used was as follows. First we input the facial image using a webcam. Preprocessing 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 python had some serious limitations.

4.The processing capacities required by python were very high. Also there were some problems with speed in real time processing. python was capable of processing only 4-5 frames per second. On a system with a low RAM this was even 9 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. This is where OpenCV came in. OpenCV is an open source computer vision library. It is designed for computational efficiency and with a strong focus on real time applications. It helps to build sophisticated vision applications quickly and easily. OpenCV satisfied the low processing power and high speed requirements of our application. 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.

5.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 createsamples() 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 3 negative background images.

6. 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.

7.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). Haarcascadeeye. 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.

**Edited**

1. Driver lethargy is considered as the major cause of vehicle accidents. Technological development for detecting and preventing driver’s drowsiness at the wheel is one of the major tasks for the developers. To avoid such disasters on roads, evolution of technology is required.

2. Development of prototype drowsiness detection system is the primary object of this project. This unique system is designed in such a way that it detects the open or closed state of driver’s eyes with accuracy at early sight and therefore avoids accidents. Detection of drowsiness is focused on blinking patterns and iris movements by sequential image capturing.

3. Firstly, we chose to detect eye blink patterns using python through procedure called geometric manipulation of intensity levels. The following algorithms are used.

Firstly, we capture a facial image using a webcam. Later, it is pre-processed by binarizing the image. It detects the top and sides of the face to narrow down the area around the eyes. Using the sides of the face, the center of the face is detected and is used as reference while computing the left and right eye. Moving from the top area, horizontal facial averages are calculated. Major changes in these averages are used to spot the eye area. This showed a minor change in horizontal average when eyes were closed, resulting the detection of eye blink.

However, python arrayed major limitations.

4. Python requires extreme processing capacities. It also had certain problems with real time processing speed. It is possible to process only 4-5 frames per second. It is even slow on low RAM systems. The eye blink and the head movement takes hardly seconds and though, the MATLAB program is designed to detect an eye blink, the performance can be future improved. Hence, OpenCV has been preferred.

OpenCV is an open source optimized library focuses on real time applications. It is designed for computational efficiency that builds up organized vision applications effortlessly. In this , we use Haartraining applications for eyes and face detection, thus creating a classifier providing a set of positive and negative samples.

5. Following are the steps to be followed: collect a data set of eyes and face which has to be stored in more than one directory recorded by a text file. High quality data is required for a classifier to perform better.

To build a vector output file, Utility application createsample() can be used and training procedure can be repeated. Before normalizing and resizing the image to a certain width and height, it extracts the positive samples from it. Any image that does not contain any object of interest is converted into negative sample. Thus, in order to learn an object, 3 negative background images are required.

6. These negative images are gathered in a single file and indexed. Training of an image is done through boosting. In training, we learn about various classifiers one at a time. Each classifier in this set contains a weak classifier typically composed of a single variable decision tree known as Stumps. During the training of each classifier one by one, data points are re-weighted such that the focus is mainly data points were errors are made. This is a continuous process until the total errors of the dataset emerges from the combined weighted votes of the decision trees, drops at a certain point.

7. Such algorithm is effective when a bulk of training data is available. We need a face and eye classifiers for which we utilize learning objects method to create our own haarclassifier.xml files. We take around 2000 to 3000 positive and negative samples. Training them is a time consuming process. Finally, face.xml and haarcascade-eye.xml files are created. These xml files are used directly for detecting a object. It detects sequential objects (like face and eyes). haaracascadeeye.xml is designed only for open eyes and closed eyes cannot be detected.

Thus, when a blink is captured in more than 5 frames, then the driver is considered to be drowsy and is immediately alarmed.

1.In the real time driver fatigue detection system, it's required to hamper a vehicle automatically when fatigue level crosses a particular limit. rather than threshold drowsiness level it's suggested to style endless scale driver fatigue detection system. It monitors the extent of drowsiness continuously and when this level exceeds a particular value a sign is generated which controls the hydraulic braking system of the vehicle.

1.In this detection system, it needs to be help a vehicle automatically when the drowsiness level crosses a particular limit. Rather than threshold fatigue level it is proposed to style endless scale detection system. It observes the level constantly, when the level surpass the limit then a sign is created which leads to handling of vehicle’s hydraulic braking system.

2.Each eye is represented by 6 (x, y)-coordinates in landmarks retuned Dlib predictor function, starting at the left-corner of the eye (as if you were looking at the person), and afterward working clockwise around the rest of the district. There is a connection between the width and the stature of these directions. Creator at that point infer a condition that mirrors this connection called the eye angle proportion (EAR).

2.