INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS…..

# DRIVER DROWSINESS DETECTION USING HAAR CLASSIFIER AND TEMPLATE MATCHING

**Prof. Rajni Pamnani[1], Faizan Siddiqui[2], DharmikGajara[3], Anil Gupta[4], Kashyap Pandya[5]**

[1]K.J.Somaiya College ofEngineering, StudentofComputer EngineeringDepartment,
VidyaVihar(E),Mumbai 400077,India
*rajaniaswani@somaiya.edu*
[2]K.J.Somaiya College ofEngineering, StudentofComputer EngineeringDepartment,
VidyaVihar(E),Mumbai 400077,India
*faizanhamed.s@somaiya.edu*
[3]K.J.Somaiya College ofEngineering, StudentofComputer EngineeringDepartment,
VidyaVihar(E),Mumbai 400077,India
*dharmik.g@somaiya.edu*
[4]K.J.SomaiyaCollege ofEngineering, StudentofComputer EngineeringDepartment,
VidyaVihar(E),Mumbai 400077,India
*anil.gupta@somaiya.edu*
[5]K.J.Somaiya College ofEngineering, StudentofComputer EngineeringDepartment,
VidyaVihar(E),Mumbai 400077,India
*kashyap.pandya@somaiya.edu*

***Abstract**:Driver fatigue is one of the major causes of accidents in the world. Detecting the drowsiness of the driver is one of the surest ways of measuring driver fatigue. In this report, we discuss a method for detecting drivers' drowsiness using Haar Classifier[I] and Template Matching Algorithm. The system then subsequently alerts the driver in order to prevent accidents. It works by monitoring the eyes of the driver and sounding an alarm when he/she is drowsy with the help of a camera that points directly towards the driver's face and capture the video. Once the video is captured, monitoring of the face region and eyes is done in order to detect fatigue. The system monitors eyes and determines whether the eyes are in an open position or closed state. In such a case when drowsiness is detected, a warning signal is issued to alert the driver. If the driver's eyes are closed cumulatively more than a standard value, the system draws the conclusion that the driver is falling asleep, and then it will activate an alarm sound to alert the driver. Yawn component is detected using Template Matching Algorithm to detect whether the driver is yawning or not and subsequently the driver is alerted with an alarm.*

***Keywords:** Haar Classifier, Template Matching, OpenCV, Drowsiness.*

## 1. INTRODUCTION

The issue of distracted driving has become common especially in a society that is more susceptible to multitasking while driving. This issue has increased the urgency for driver inattentiveness and drowsiness detection system technology, to compensate for driver error. The system which is developed will run in a simulated environment with a subject (driver) sitting in front of the camera which is integrated with the software. This system will present a non-intrusive approach for drowsiness detection based on computer vision. A camera is placed in front of the driver in order to detect his face and obtain drowsiness clues form their eye closure and yawning component. If the system detects that the eyes are closed for a certain no. of frames then it will buzz an alert alarm indicating the driver to be alert.

## 2. LITERATURE REVIEW

### *2.1 Need for the system*

Industrialized countries, drowsiness has been estimated to be involved in 20% to 23% of all crashes. Increase in such fatal crashes has urged many automobile companies now-a-days to build their own software that will detect whether a driver is feeling sleepy and fatigue well in advance. We have thus build a system that will detect when drivers are becoming drowsy and promise to be a valuable aid in preventing accidents. The system uses a small monochrome camera that points directly towards the driver's face and monitors the driver's eyes & mouth in order to detect fatigue. This report describes how to find the eyes, and also how to determine if the eyes are open or closed and the mouth structure to conclude whether the driver is yawning or not. The system deals with using information obtained using Haar-based classifier to find the edges of the face, which narrows the area of where the eyes may exist. Once the face area is found, the eyes are found by looking in the area of the cropped face to determine whether the eyes are open or closed. If the eyes are found closed for specific no. of frames (for instance 8), the system draws the conclusion that the driver is falling asleep. The system keeps track of the mouth for detecting yawn using Template matching Algorithm which detects the yawning component by matching the templates already stored in the system. A combination of both algorithms helps to achieve good results.

### *2.2 Programming & software tools*

*A) C#.Net*

C# is an elegant and type-safe object-oriented language that enables developers to build a variety of secure and robust applications that run on the .NET Framework. C# can be used to create traditional Windows client applications, XML Web services, distributed components, client-server applications, database applications, and much, much more.

.Net Framework:- C# programs run on the .NET Framework, an integral component of Windows that includes a virtual execution system called the common language runtime (CLR) and a unified set of class libraries. The CLR is the commercial implementation by Microsoft of the common language infrastructure (CLI), an international standard that is the basis for creating execution and development environments in which languages and libraries work together seamlessly.

*B) OPENCV*

We used OpenCV because OpenCV [OpenCV] is an open source computer vision library. OpenCV was designed for computational efficiency and has a high focus on real-time image detection. OpenCV is coded with optimized C and can take work with multi-core processors OpenCV automatically uses the IPP library optimized for Intel-based architectures, at runtime if that library is installed.

## 2. NEW PROPOSED SCHEME

### *3.1 The system*

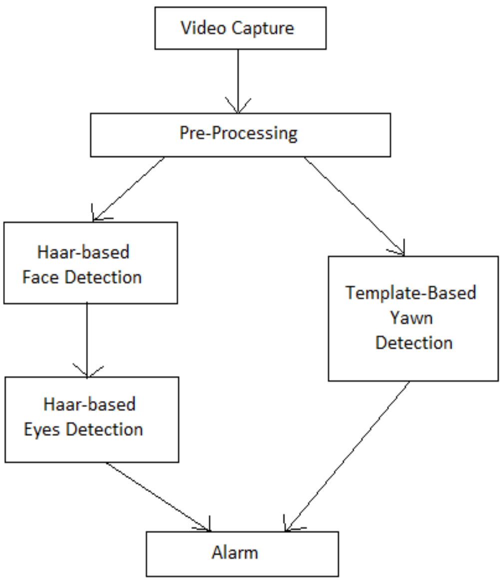


**Figure1**:The system

The flow of the system is as shown above.

The webcam captures the video and alarms the driver of his drowsiness by monitoring eyes and mouth.

### *3.2 Components of System*

*A)Video Capture*

In this component, the system continuously captures image frames from the webcam and passes it to the next component for pre-processing. The video camera is initialized using the Capture() and the frames are captured using the QuerySmallFrame() of OpenCV.

```
vidCapture = new Capture();
imgLive = vidCapture.QuerySmallFrame();        [1]
```

*B) Pre-Processing*

This component deals with the pre-processing of the captured frame before the application of the algorithms.

First, the image is resized to a particular size (in our case 320*240 pixels) by using Resize() of OpenCV. Then, the image is converted to gray-scale format as the processing of image is a little simpler in this format. Conversion of image frame is done using Covert<Gray,Byte>() function. After conversion, the image undergoes Histogram Equalization which helps to adjust the overall contrast of the image. It is basically useful when the driver is driving in low-light conditions so that the intensities can be better distributed on the histogram. As a result, areas of lower local contrast to gain a higher contrast. EqualizeHist() function of OpenCV is used to achieve image histogram equalization.

```
imgLive=vidCapture.Resize(320,240)
Image<Gray,Byte> imgGray = imgLive.Convert<Gray, Byte>();
imgGray.EqualizeHist();        [2]
```

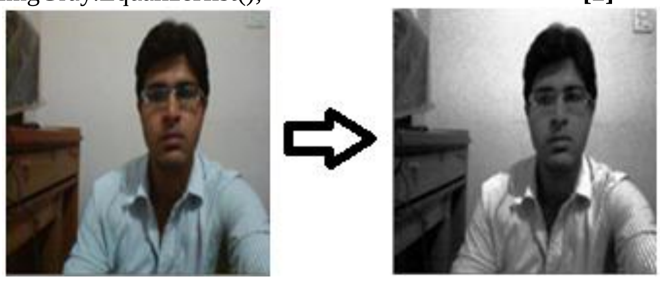

**Figure2:** Before & After Pre-Processing

*C) Haar-based Classifier*

i.) Face detection

A Haar-classifier is trained for detecting the facial region within the frame. The training of the classifier is done in OpenCV. A set of positive and negative samples are taken for training. Positive samples are the ones which contain faces while negative samples are the random images without the facial region. The OpenCV software helps to create a vector file which is a combination of cross product of all the positive and negative samples. Finally, the result of the training operation is a XML file which is used for detecting the facial region in the upcoming frames. The DetectMultiScale() helps to mark all the facial region in the upcoming frame and are stored in an array. These co-ordinates help us to mark the facial region in the image. In case of multiple faces, the area of all the facial regions are calculated and the bigger one is assumed to be the driver's face on which the further processing is done.

ii.) Eyes detection

Like face, a Haar-classifier is trained for eyes too. A set of positive and negative samples are taken for training. Positive samples contain images containing eyes while negative samples are the random images without the eye region. The result of the training operation is XML file which is used for detecting the eyes. Eyes, being a small feature is difficult to detect in the entire frame. As a result, the facial region obtained in the previous step is cropped and eyes are detected in the cropped image.

Steps involved in the above algorithm:

<u>Step 1:</u> Select a pre-processed frame from the video.

<u>Step 2</u>: Detect face from the image using Haar-classifier. If number of faces in the image>0 then crop the facial region as the Region of Interest(ROI) else go to next frame.

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS…..

Step 3: Detect eyes from Region of Image (ROI) using Gaar-classifier trained for eyes.
Step 4: If number of eyes greater than zero(i.e. if eyes are detected) set counter K=0 and go to Step 1.
Else Increment the counter K=K+1.
Step 5: If K>threshold value, system concludes driver is drowsy and sounds an alarm,
Else go to next frame.

```
Rectangle[] facesDetected = faceHaar.DetectMultiScale(imgGray, 1.3, 10, new Size(20, 20), Size.Empty);
faces.AddRange(facesDetected);
imgMain.Draw(face, new Bgr(Color.Red), 2);
```
**[3]**

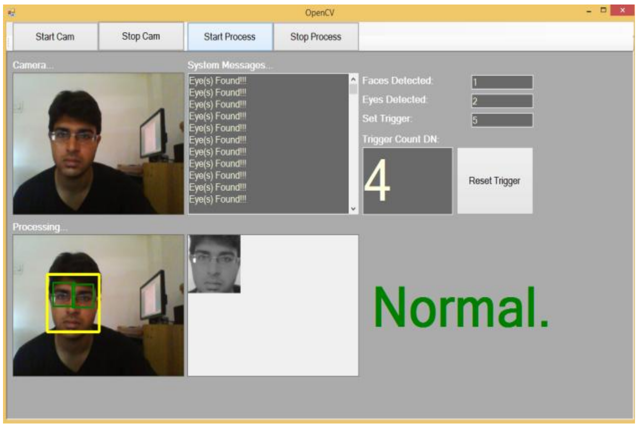


**Figure 3(a):** Face and Eyes detected

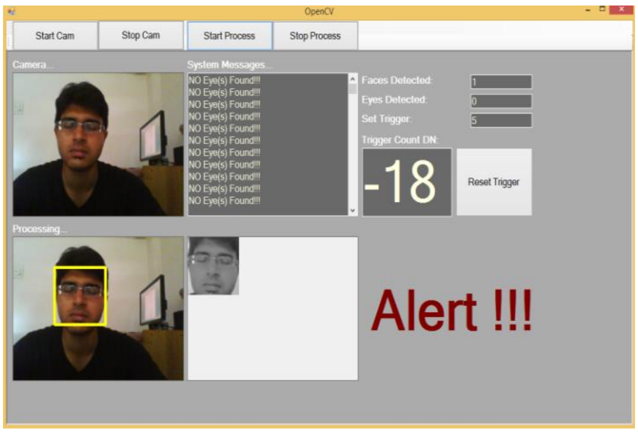


**Figure 3(b):** Alert when eyes closed

Our approach concludes that the driver is drowsy if the eyes are closed (i.e. not detected) for a specified number of frames.

*D) Template-based Yawn Detection*

This component stores the yawn template of the driver in the database and compares it with the upcoming frames to predict whether the driver is yawning or not. First, the driver needs to capture his/her yawn template from the live video and allow the application to store it in the database. The stored template image is compared with the upcoming pre-processed image. Our goal is to detect the highest matching area. To identify the matching area, we have to compare the template image against the source image by sliding it. Sliding means moving the patch one pixel at a time (left to right, up to down). At each location, a metric is calculated so it represents how "good" or "bad" the match at that location is i.e how similar the patch is to that particular area of the source image.

MatchTemplate() function of OpenCV is used for template matching.

```
imgMatch = imgMain.MatchTemplate(imgL, Emgu.CV.CvEnum.TM_TYPE.CV_TM_CCOEFF_NORMED;
imgMatch.MinMax(out Min, out Max, out pointMin, out pointMax);

imgMain.Draw(rect, new Bgr(Color.Red), 2);
```
**[4]**

Steps involved in the above algorithm:
Step1: Store the driver's yawn template in the database.
Step 2: Select a pre-processed frame from the video.
Step 3: Match the stored template in the upcoming frame using template matching algorithm.
Step 3: Select the region with the highest match.
Step 4: If the match is obtained, set counter L=0 and go to Step 1.
Else Increment the counter L=L+1.
Step 5: If If L>threshold value, system concludes that driver is yawning and sounds an alarm.
Else go to next frame.

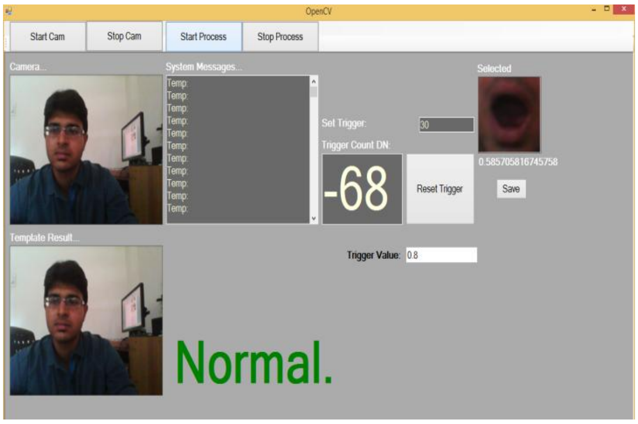


**Figure 4(a):** Template stored

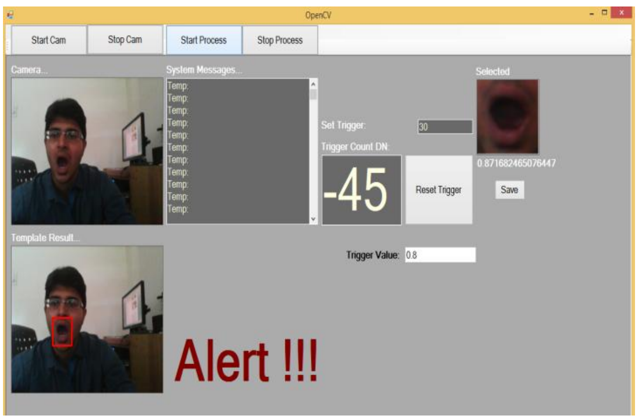


**Figure 4(b):** Alert when yawn detected

As yawning lasts for approximately 2-3 sec, we have set the threshold to a 25 frames as our application records video at 10-12 fps. On detecting yawn, the system sounds an alarm to alert the driver about his drowsiness. Our system monitors both the

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS…..

eyes and yawn components. A combination of two results helps to achieve better results and also more suitable for pratical application. The system accuracy was little down while tracking drivers wearing spectacles. Also, the system was not able to detect tilted faces.

## 3. FUTURE WORK

Training the classifier with more images so that the driver wearing spectacles can be recognized with more accuracy. Currently, the system can detect only slightly tilted face of the driver. So, we are looking for an approach to easily detect tilted faces so that the application can be practically used.

## 4. CONCLUSION

In this paper, we try to build a software product that will detect the drowsiness of the driver at an early stage and will try to warn him/her through a buzzer etc. We also see that image processing achieves highly accurate and reliable detection of drowsiness. Image processing offers a non-invasive approach to detecting drowsiness without annoyance and interference. A drowsiness detection system developed around the principle of image processing judges the driver's alertness level on the basis of continuous eye closures.

## ACKNOWLEDGEMENT

The authors would like to thank all the earlier work regarding different methods of driver drowsiness detection that contribute the work made in this paper.

## REFERENCES

**[1] Jihong Liu, Zhongfan Li ,Li Yang, Zhan Mei," Study of the Eye-tracking Methods Based on Video", 2011 Third International Conference on Computational Intelligence, Communication Systems and Networks.**

**[2] Lu Yufeng, Wang Zengcai ,"Detecting Driver Yawning in Successive Images" IEEE 2009.**

**[3] Shabnam Abtahi, Behnoosh Hariri, Shervin Shirmohammadi,"Driver Drowsiness Monitoring Based on Yawning Detection" IEEE 2011.**

**[4] Trupti Dange, Tarun Yengatiwar ,"Non-Intrusive Drowsiness Detection Using Artificial Techniques ",Journal of Computer Engineering and Technology(JCET),2013.**

**[5] Matej Cerny, Martin Dobrovolny, "Eye Tracking System on Embedded Platform".**

**[6] Xianghua Fan, Fuyou Zhang, Haixia Wang, Xiao Lu, "The System of Face Detection Based on OpenCV" , 2012 IEEE**

**[7] Sang-Joong Jung, Heung-Sub Shin, Wan-Young Chung, "Driver Fatigue and Drowsiness Monitoring System with embedded ECG Sensor on Steering Wheel" , Intelligent Transport Systems, IET (Volume:8, Issue:1),February 2013.**

**[8] T.A. Dingus, S.G. Klauer, V.L. Neale, and A. Petersen, "The 100-car naturalistic driving study," Conducted by Virginia Tech Transportation Institute, Co-Sponsors: National Highway Traffic Safety Administration and Virginia Department of Transportation, 2008.**

**[9] Weirwille, W.W. (2006), "Overview of Research on Driver Drowsiness Definition and Driver Drowsiness Detection," 14th International Technical Conference on Image Processing**

**[10] http://vision.caltech.edu/html-files/archive.html**

**[11] docs.opencv.org/doc/tutorials/tutorials.html**