International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

Implementation of Real Time Driver Drowsiness Detection System

Snehal S. Bharambe[1], P. M. Mahajan[2]

[1, 2] ME Electronics and Telecommunication Department, North Maharashtra University, Jalgaon
J. T. Mahajan College of Engineering, Faizpur, India

Abstract: ***Today, number of accidents happen during drowsy driving on roads and are increasing day by day. It is a known fact that many accidents occur due to driver's fatigue and sometimes due to inattention factor. This research mainly engages on maximizing the effort in identifying the drowsiness state of driver in real driving conditions. The goal of driver drowsiness detection systems is an attempt to contribute in reducing these road accidents. The secondary data collected focuses on past research on drowsiness detection systems and various methods have been used earlier for detection of drowsiness or inattention while driving. However, in this paper, a real time vision-based method is proposed to monitor driver fatigue. This research approach adopts the Viola-Jones classifier to detect the driver's facial features. Firstly, the face is located by a Haar like feature based object detection algorithm. The face area is detected using the functions in the OpenCV library with C#.net. Secondly, eye is detected. Also the eye areas are detected by using the functions in the OpenCV library and tracking by using a template matching method. Then, the open/close state of eyes is determined, and then fatigue is determined based on the series state of eyes. The correlation coefficient template matching method is applied to derive the state of each feature on a frame by frame basis. Vision- based driver fatigue detection method is a natural, non-intrusive and convenient technique to monitor driver's vigilance.***

Keywords: Driver Drowsiness Detection, Face Detection, Eye Detection, Eye Tracking, Haar Classifier, Template Matching.

1. Introduction

Generally, there are many reasons behind highway traffic accidents. Driver drowsiness is one of the major causes of serious traffic accidents. According to the National Highway Traffic Safety Administration (NHTSA) [1], there are about 56,000 crashes caused by drowsy drivers every year in US, which results in about 1,550 fatalities and 40,000 nonfatal injuries annually. The National Sleep Foundation also reported that 60% of adult drivers have driven while felling drowsy in the past year, and 37% have ever actually fallen asleep at the wheel [2]. Many of the road accidents are occur due to driver fatigue/ driver drowsiness or driver sleepiness. Sleepiness reduces the concentration, activeness, alertness and vigilance of the driver and it makes the driver to take slow decisions and sometimes no decision. Drowsiness affects the mental alertness and decreasing the driver ability to operate a vehicle safely and increasing the risk of human error that could lead to fatalities and injuries. The reasons for the fatigue related crashes are long journeys on monotonous roads, driving after eating or taking an alcoholic drink, having less sleep than normal, after taking medicines that cause drowsiness, driving after long working hours and journeys after night shifts etc. Hence there is a need to address this problem to avoid accidents by alerting the driver so that road safety can be increased.

The aim of this paper 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.

2. Related Work

In-vehicle camera is commonly installed to realize the possible reasons of car accidents. Such a camera can also be used to detect the fatigue of the driver. Several studies related to the fatigue detection are described as follows.

Sharma et al. [3] utilized the number of pixels in the eye image to determine the eye state, open or close. Hornget at. [4] established an edge map to locate the eyes locations and the eye state is determined based on the HSL color space of the eye image. Its accuracy is dependent on the location of the eyes. Sharma and Banga [3] converted the face image to YCbCr color space. The average and standard deviation of the pixel number in thebinarization image is computed. Then, fuzzy rules [5] are used to determine the eye state. Liu et al. [6] and Tabrizi et al. [7] proposed methods to detect the upper and lower eyelids based on the edge map. The distance between the upper and lower eyelids is then used to analyze the eye state. Besides, Dong et al. [8] and Li et al. [9] proposed methods by utilizing AAM (Active Appearance Model) to locate the eyes. Then, a PERCLO (PERcentage of eye CLOsure) was computed to detect the fatigue. For the above methods, the locating of eye areas was easily influenced by the change of brightness. Circular Hough transform is popular method to overcome the influence of brightness. Several studies [10, 11, 12] proposed methods to locate the pupil of eyes by using circular Hough transform. Then,the eye state was analyzed according to the locations of pupils. Zhengpei calculated the ratio of eye closing during a period of time. The ratio can reflect driver's vigilance level [13]. Wenhui Dong proposed a method to detect the distance of eyelid, and then judged the driver's status by this kind of information [14]. Nikolaos P used front view and side view images to precisely locate eyes [15]. Edge detection and gray-level projection methods were also applied for the eyes location by Wen-Bing Horng [16]. Zutao Zhang located the face by using Haar algorithm and proposed an eye tracking method based on Unscented Kalman Filter [17]. Abdel Fattah Fawky presented a combination of algorithms, namely wavelets transform, edge detection and YCrCb transform in the eye detection [18]. QiangJi depended on IR illumination

to locate eyes [19]. Eyes always contain two kinds of information: size of opening and duration of the different states. By analyzing the change rules of eyes in fatigue, we propose an efficient approach for driver fatigue detection.

3. Proposed System

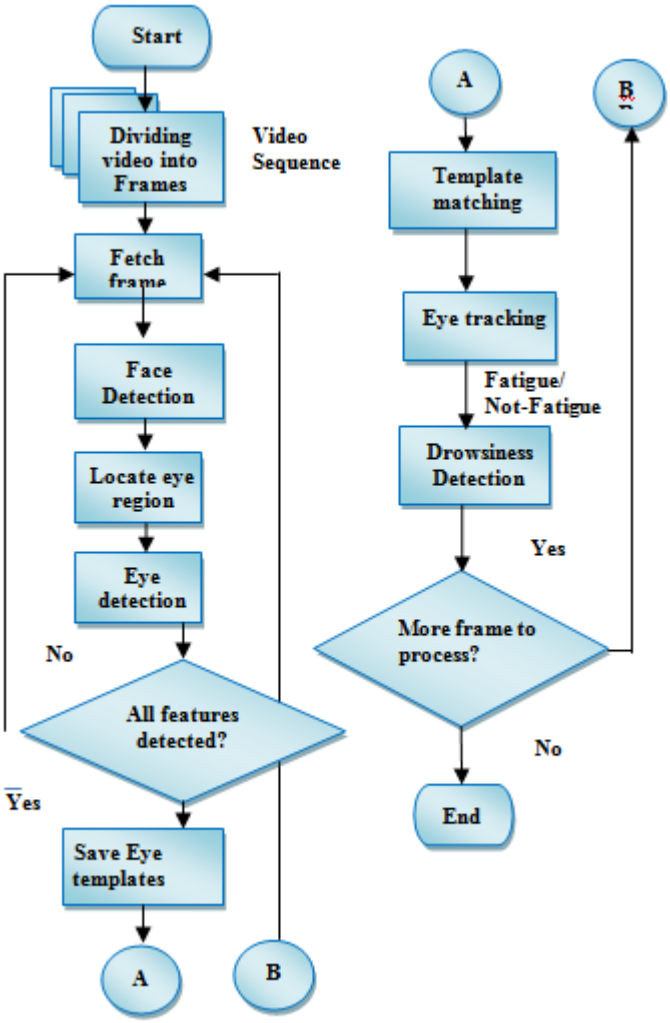

Figure 1: Flowchart of Driver Drowsiness Detection System

In proposed method, first the image is acquired by the webcam for processing. The images of the driver are captured from the camera which is installed in front of the driver on the car dashboard. It will be passed to preprocessing which prepares the image for further processing by the system. Its main operations are to eliminate noises caused by the image acquisition subsystem and image enhancement using Histogram Equalization. Then we 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. If an eye is detected then there is no blink and the blink counter is set to '20'. If the eyes are closed in a particular frame, then the blink counter is decremented and a blink is detected. When the eyes are closed for more than 4 frames then it is deducible that the driver is feeling drowsy. Hence drowsiness is detected and an alarm sounded. After that the whole process is repeated as long as the driver is driving the car. The overall flowchart for drowsiness detection system is shown in Figure 1.

3.1 Face Detection

Face detection is accomplished by the Haar algorithm [20, 21] proposed by Paul Viola and Michael Jones in 2001. Due to the complex background, it is not a good choice to locate or detect both the eyes in the original image, for this we will take much more time on searching the whole window with poor results. So firstly, we will locate the face, and reduce the range in which we will detect both the eyes. After doing this we can improve the tracking speed and correct rate, reduce the affect of the complex background. Besides, we propose a very simple but powerful method to reduce the computing complexity.

i) Haar-like features:
The simple features used are suggestive approaches of Haar basis functions which have been used by Papageorgiou et al. [22]. A Haar-like feature considers affixed rectangular regions at a specific part in a detection window; each Haar-like feature expressed by two or three jointed black and white rectangles shown in figure 2. The value of a Haar like feature is the difference between the sums of the pixel values within the black and white rectangular regions. These sums are used to find the difference between regions. Then the differences can be used to classify the sub region of an image. These differences are compared against learned threshold values to determine whether or not the object appears in the region.

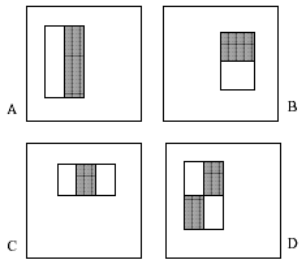

Figure 2: Rectangle features example: (A) and (B) shows two-rectangle features, (C) Shows three-rectangle feature, and (D) shows four-rectangle feature

ii) Integral image
The simple rectangular features of an image are calculated using an intermediate representation of an image, called the integral image as in (1) .The integral images are an array which consists of sums of the pixels' intensity values located directly to the left of a pixel and directly above the pixel at location (x,y) inclusive. Here, A[x,y] is the original image and Ai[x,y] is the integral image[21].

$$Ai[x, y] = \Sigma A[x', y'] \quad x' \leq x, y' \leq y \quad (1)$$

iii) AdaBoost
Adaboost, nothing but "Adaptive Boosting ", It can be used with many other types of learning algorithms to improve their performance. Adaboost takes a number of positive and negative images features and training sets, The machine creates a set of weak classifiers of Haar-like features. It selects a set of weak classifiers to combine and that assigns lesser weights to good features whereas larger weights to

poor features. This weighted combination gives strong classifier.

iv) Cascaded classifier

The cascade classifier consists of number of stages, where each stage is a collection of weak learners. The weak learners are simple classifiers known as decision stumps. Boosting is used to train the classifiers. It provides the ability to train a highly accurate classifier by taking a weighted average of the decisions made by the weak learners.

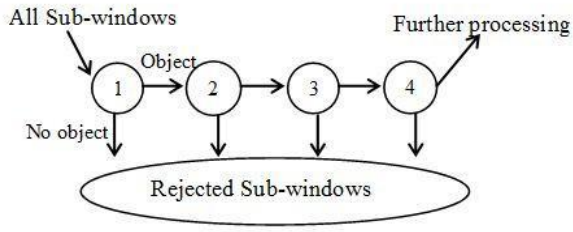

Figure 3: Cascade of Classifiers

Each stage of the classifier shows the region defined by the current location of the sliding window as either positive or negative. Positive indicates an object was found and negative indicates no object. If the label is negative, the classification of this region is complete, and the detector shifts the window to the next location. If the label is positive, the classifier passes the region to the next stage. The detector reports an object found at the current window location when the final stage classifies the region as positive. It is used to eliminate less likely regions quickly so that no more processing is needed. Hence, the speed of overall algorithm is increased.

3.2Eye Detection

Images or the real time video is captured from the camera installed in front of the driver's face. This video is converted into number of frames. OpenCV face Haar-classifier is loaded. Each frame is compared with the pre-defined features of the Haar-classifiers. When the features are matched the face is detected and a rectangle is drawn around the face. Using feature extraction we estimate the position of the eyes. By comparing with the OpenCV eye-Haar classifier, the eyes are detected and rectangles are drawn around left and right eye.

i) Template Matching

Template matching is basically the two-dimensional cross-correlation of a grayscale image with a grayscale template; hence it can be used to estimate the degree of similarity between the two images. Template matching is sensitive to variation of poses. Template matching is necessary for the desired accuracy in analyzing the user's blinking since it allows the user some freedom to move around slightly. Template matching is a technique in digital image processing for finding small parts of an image which match a template image. The normalized correlation coefficient is used to accomplish the tracking. This measure is computed at each frame using the following formula as in (2) below:

$$\frac{\sum_{x,y}[f(x,y)-fu,v][t(x-u,y-v)-t]}{\sqrt{\sum_{x,y}[f(x,y)-fu,v]2\sum_{x,y}[t(x-u,y-v)-t]2}} \quad (2)$$

Where, f (x, y) is the brightness of the video frame at the point (x, y), f (u, v) is the average value of the video frame in the current search region, t(x, y) is the brightness of the template image at the point (x, y), and t is the average value of the template image. The result of this computation is a correlation score between -1 and 1 that indicates the similarity between the open eye template and all points in the search region of the video frame. Scores closer to 0 indicate a low level of similarity, and to 1 indicate a probable match for the open eye template.

4. Software Implementation

The implementation is done by using EmguCV [23] which is an open-source image processing library for C#.NET. Emgu CV is a cross platform .Net wrapper to the OpenCV image processing library. It permit OpenCV functions to be called from .NET compatible languages such as C#, VB, VC++, Iron, Python etc..To detect human facial feature, Intel developed an Open source library used for the implementation of computer vision related programs called OpenCV(Open source computer vision). OpenCV library is used for implementation of Haartraining[13]. We have used the Haar training applications in OpenCV to detect the face and eyes. This creates a classifier given a set of positive and negative samples. 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 vision applications quickly and easily. OpenCV satisfies the low processing power and high speed requirements of our application.

5. Hardware Implementation

The basic block diagram of the system is shown in figure 1.

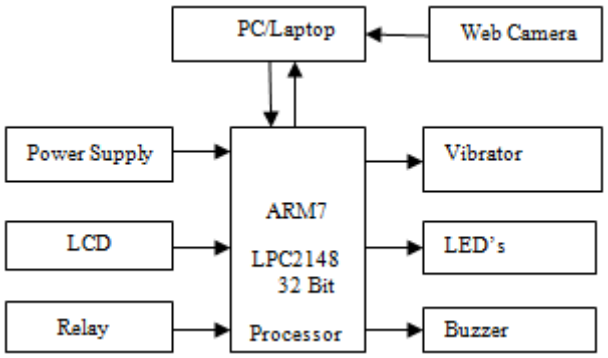

Figure 4: Block Diagram of Driver Drowsiness Detection System.

The system basically consists of webcam, microcontroller ARM7, buzzer, vibrator, LED's, relay. Microcontroller is used to perform various function and operations is ARM7 i.e. LPC2148. This paper presents a non-intrusive approach for drowsiness detection, based on computer vision. The webcamera is installed in a car and it is able to work under real operation conditions.

5.1 ARM 7 LPC 2148:

ARM stands for Advanced RISC Machines. It is a 32-bit processor core used for high end applications. The LPC2148

microcontrollers are based on a 16-bit/32-bit ARM7TDMI-S CPU with real-time emulation and embedded trace support, that combine the microcontroller with embedded high speed flash memory ranging from 32KB to 512KB. ARM (Advanced RISC Machine) T–The Thumb 16 bit instruction set. A 128-bit wide memory interface and unique accelerator architecture enable 32-bit code execution at the maximum clock rate [4].

5.2Relay driver-Single channel

This unit provides actual switching of external device connected to the pin of relay. The voltage of the coil of the relay is 12V. That means it will energies at minimum 12 voltage on across it. After tenderization of the coil the mechanical key present inside the relay switches to its other position and vice versa. This gives a heavy induced e.m.f. which can cost the rest of circuit to burn out but this is prevented by using a diode in parallel with the coil in opposite direction.

5.2 Webcam

A webcam is a video camera that feeds its image in real time to a computer or computer network. Unlike an IP camera (which uses a direct connection using Ethernet or Wi-Fi), a webcam is generally connected by a USB cable, FireWire cable, or similar cable. Their most popular use is the establishment of video links, permitting computers to act as videophones. The common use as a video camera for the World Wide Web gave the webcam its name. Other popular uses include security surveillance, computer vision, video broadcasting, and for recording social videos.

6. Experimental Results

We have used Open CV as a platform to develop a code for eye detection in real time. The code is then implemented on system installed with Open CV software. To detect human eyes, face has to be detected initially. This is done by OpenCV face haar cascade classifier. Once the face is detected, the location of the eyes is estimated and eye detection is done using eye Haar-cascade classifier. Hence using the open CV, face and eyes are detected accurately and displayed on the monitor as shown in the Figure 5 (a). The larger yellow square indicates the face while smallerred squares indicate the eyes.

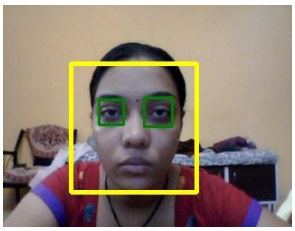

Figure 5 (a): Photograph showing the detected face and eyes.

Once face and eyes are detected, it is checking status of eyes i.e. open or closed state of the eyes. If both eyes remain closed for successive frames, it indicates that the driver is drowsy and gives the warning signal as shown in figure 5 (b).

Figure 5 (b): When both eyes are closed gives warning signal.

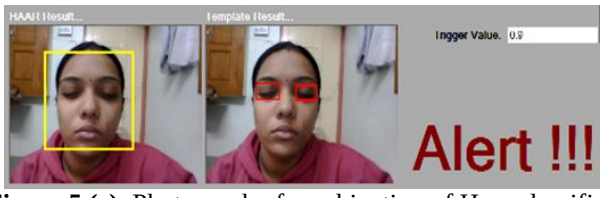

Figure 5 (c): Photograph of combination of Haar classifier and Template matching method.

A fatigue detection system based on the above method was implemented by using Visual C++. At first, we fix a camera on a car in front of the driver. Then we capture some videos from 8 drivers in normal conditions. The whole input image format is 320×240 and they are in RGB color space. We have also found that the optimum distance from camera which obtained about 30cm-50cm that is very suitable for our method.

Table1, Table 2, Table 3 shows the experimental results for each method, where the accuracy is the number of frames of open eyes divided by the total number of frames multiplied by 100. The system proposed at the average of 15 frames per second. It surely met the needs of real time. For the proposed method, the correct ratio of open eyes is higher than that of eye close. The average accuracy of our combination method is 90.873%. Thus our eye detection method is robust and irrelevant with different sizes and more accurate. According to obtained results, our system can determine the eye states with a high rate of correct decision.

Table 1: Eye state detection for Haar Classifier method

Videos	*HaarClassifer*		
	Total frames	*Open*	*Ratio (%)*
V1	189	158	83.598
V2	224	186	83.036
V3	182	150	82.410
V4	211	178	84.360
V5	252	206	81.746
V6	176	144	81.818
V7	192	157	81.771
V8	222	184	82.883

Table 2: Eye state detection for Template Matching

Videos	*Template Matching*		
	Total frames	*Close*	*Ratio (%)*
V1	277	28	89.892
V2	256	22	89.06
V3	204	25	87.741
V4	298	36	87.919
V5	348	44	87.35
V6	364	47	87.088
V7	312	30	90.385
V8	302	32	89.404

Table 3: Eye state detection for our combination method

Videos	*Combination of Haar and Template matching*			
	Total frames	*Open*	*Close*	*Ratio (%)*
V1	363	327	18	90.083
V2	405	369	27	91.111
V3	383	345	25	90.078
V4	412	378	29	91.748
V5	420	384	11	91.429
V6	392	353	20	90.051
V7	357	323	24	90.476
V8	375	345	20	92.017

For every test videos, the total frames are marked in the parentheses. The number of frames with eye open and close area also marked below. The correct ratio for open eyes is computed for every test videos.

7. Conclusions

The proposed system in this analysis provides accurate detection of driver fatigue. The analysis and design of driver drowsiness detection system is presented. The proposed system is used to avoid various road accidents caused by drowsy driving and it can also help drivers to stay awake when driving by giving a warning when the driver is sleepy. And also this system used for security purpose of a driver. During the monitoring, the system is able to decide if the eyes are opened or closed. When the eyes have been closed for too long, a warning signal is issued. Image processing achieves highly accurate and reliable detection of drowsiness. This was achieved by interfacing a webcam to a PC and recording test videos and frame database under different lighting condition. The calculation speed, accuracy and robustness will be influenced by using combined algorithm.

References

[1] NCSDR/NHTSA Expert Panel on Driver Fatigue and Sleepiness, “Drowsy driving and automobile crashes,” National Highway Traffic Safety Administration (NHTSA) report.
[2] M. R. Rosekind, “Underestimating the societal costs of impaired alertness: safety, health and productivity risks,” Sleep Medicine, vol. 6, pp. S21-S25, 2005.
[3] N. Sharma, V. K. Banga, "Development of a Drowsiness Warning System based on the Fuzzy Logic ," International Journal of Computer Applications, vol. 8, no. 9, pp.1-6, 2010.
[4] W. B. Horng, C. Y. Chen, Y. Chang and C. H. Fan, "Driver Fatigue Detection Based on Eye Tracking and Dynamic Template Matching," Proceedings of International Conference on Networking, Sensing & Control, Taipei, Taiwan, pp. 7-12, 2004.
[5] K. P. Yao, W. H. Lin, C. Y. Fang, 1. M. Wang, S. 1. Chang, and S. W. Chen "Real-Time Vision-Based Driver Drowsiness/Fatigue Detection System," Proceedings of IEEE 71st Vehicular Technology Conference, Taipei, Taiwan, pp. 1-5, 2010.
[6] D. Liu, P. Sun, Y. Q. Xiao, Y. Yin, "Drowsiness Detection Based on Eyelid Movement," Proceedings of the Second International Workshop on ETCS, Wuhan, China, pp. 49-52, 2010.
[7] P. R. Tabrizi and R. A. Zoroofi, "OpenlClosed Eye Analysis for Drowsiness Detection," Proceedings of the First Workshops on Image Processing Theory, Tools and Applications, Sousse, Tunisia, pp. 1-7, 2008.
[8] H. Z. Dong, M. Xie, "Real-Time Driver Fatigue Detection Based on Simplified Landmarks of AAM," Proceedings of the ICACIA, Chengdu, China, pp. 363-366, 2010.
[9] L. Li, M. Xie, H. Dong, "A Method of Driving Fatigue Detection Based on Eye Location," Proceedings of the IEEE 3rd ICCSN, Xian, China, 20 II, pp. 480-484.
[10] M. I. Khan and A. B. Mansoor, "Real Time Eyes Tracking and Classification for Driver Fatigue Detection," Lecture Notes in Computer Science, vol. 5112, pp. 729-738, 2008.
[11] I. Garcia, S. Bronte, L. M. Bergasa, N. Hernandez, B. Delgado, M. Sevillano "Vision-based Drowsiness Detector for a Realistic Driving Simulator," Proceedings of the 13th International IEEE ITSC, Funchal, Portugal, pp. 887-894, 2010.
[12] N. A1ioua, A. Amine, M. Rziza and D. Aboutajdine, "Driver's Fatigue and Drowsiness Detection to Reduce Traffic Accidents on Road," Proceedings of the 14th International conference on CAlP, Seville, Spain, pp. 397-407, 2011.
[13] Z. Pei, S. Zhenghe, and Z. Yiming, “Perclos-based recognition algorithms of motor driver fatigue,” Journal of China Agricultural University, pp. 104-109, 2004.
[14] W. Dong, X. Wu, “Driver Fatigue Detection Based on the Distance of Eyelid,” Procceeding IEEEInternational Workshop VLSI Design & Video, Technology, pp. 28-30, 2005.
[15] P. Nikolaos , “Vision-based Detection of Driver Fatigue,” Proc. IEEE InternetionalConference on Intelligent Transportation, 2000.
[16] W. Horng, C. Chen, Y. Chang, “Driver Fatigue Detection Based on Eye Tracking and Dynamic Template Matching,” Proc.of the 2004 IEEE International Conferenceon Networking, Sensing &Control, pp. 7-12, 2004
[17] Z. Zhang, J. Zhang, “A New Real-Time Eye Tracking for Driver Fatigue Detection,” Proc.2006 6th InternationalConference on ITS Telecommunications, pp 8-11, 2006.
[18] A. Fawky, S. Khalil, and M. Elsabrouty, “Eye Detection to Assist Drowsy Drivers,” IEEE, pp. 131-134, 2007.
[19] Q. Ji, Z. Zhu, and P. Lan, “Real-Time Nonintrusive Monitoring and Prediction of Driver Fatigue,” IEEE Transaction on vehicular technology, vol. 53, no. 4, pp. 657-662, 2004.
[20] P. Viola, M. J. Jones (2004). "Robust Real-Time Face Detection." International Journal of Computer Vision vol. 57(2), pp. 137-154.
[21] P. Viola, M. Jones, “Rapid Object Detection Using a Boosted Cascade of Simple Features,” In Proceeding IEEE Conference Computer Vision and Pattern Recognition.
[22] C. Papageorgiou, A General Framework For Object Detection, In International Conference On Computer Vision, 1989.
[23] Wiki, E. (n.d.). EmguCV, Retrieved January 20, 2012, from http://www.emgu.com/wiki/index.php/Main_Page.