

A Real-time Fatigue Driving Detection System Design and Implementation

Zhao-Bin Ma*, Yang Yang*, Fengyu Zhou**, Xu Jian-Hua***

*School of Information Science and Engineering, Shandong University, Jinan, China, 250100

**School of Control Science and Engineering, Shandong University, Jinan, China, 250014

***The 41st Research Institute of China Electronics Technology Group Corporation, Qingdao, 266555, China

yyang@sdu.edu.cn

Corresponding Author: Yang Yang

Abstract— A reliable and real-time fatigue driving detection system is very important for traffic safety. Till now, various methods have been presented to improve the accuracy and robustness, however, few papers showed the efficiency of the system or the feasibility of real-time processing in embedded system. Considering the efficiency and reliability, a real-time fatigue driving detection system is presented in this paper. Firstly, Haar-like features and AdaBoosted classifiers are adopted for real-time face detection. The selection of the optimal sampling ratio and the minimum possible object size is discussed for high detection accuracy and computational efficiency. Secondly, eye location and iris positioning are applied to detected faces. We propose to use the area proportion that the iris takes in its minimum circumscribed circle as the indicator for judging the state of the eye, which is more convenient and has good robustness to distance. Finally, the percentage of eye closure (PERCLOS) is used as the criteria to determine whether the driver is tired. The embedded platform implementation of the proposed system turns out to be efficient.

Keywords—Fatigue driving detection, Face detection, Real-time, AdaBoosted classifier, Iris positioning, PERCLOS

I. INTRODUCTION

The design of a reliable and real-time fatigue driving monitoring and early warning system is very important for drivers themselves and the public, as showed in [1]. Nowadays, the fatigue driving detecting methods can be divided into the following three categories in accordance with the detecting principle:

1) Physiology based methods. These methods judge the driver's fatigue state with some physiological indicators (EEG, ECG, etc.) which are monitored by specialized equipment (e.g. [2]). They're accurate, but not easily acceptable as they will cause some additional interference to drivers.

2) Driving behavior or vehicles' track based methods. These methods are based on the monitoring of how the drivers steer the wheel and the vehicle's behavior (e.g. [3]). The key point of these methods is whether the driver carries on the right operation when driving in a certain circumstance. However, these behaviors vary among drivers and are easily influenced by different road conditions, as in [4].

3) Computer vision based methods. When people is tired, there will be some typical features on its face, such as frequent nod, yawning, long time of eye closure, more eyelid covering the pupil, etc. The fatigue detection method based on computer vision is to judge the state of fatigue by monitoring these features, as in [5]. This kind of methods are non-contact and more easily to be accepted, although they may be influenced by light, vibration and some other disturbances.

Among all of these computer vision based methods, the analysis of eye state attracts much more attention than others. The key point of these methods is to judge whether the eye is open or close. Liu et al. [11] determined the eye state based on eyelid movement, while Tabrizi et al. [12] did this by measuring the distance between the upper and lower eyelids and comparing it with the radius of the iris. Horng et al. [13] established an edge map to locate the eyes and got the result by the number of pixels in the eyes. Other methods such as AdaBoosted LBP, as in [14], was also introduced to do the classification.

However, few of these papers showed the efficiency of the proposed systems or the feasibility of real-time processing in embedded systems. Fortunately, Cheng et al. [15] proposed an efficient system that could handle 15~16 frames per second. In this system, face detection was realized based on OpenCV library, i.e., an AdaBoosted classifiers using Haar-like features. After that, it located the eye with the face anthropometric properties presented in [16]. And an eyeglasses removal method was proposed, while the eye state was judged by the maximum number of consecutive white pixels in the binarization image.

The work in [15] is simple enough to an embedded system implementation, yet more improvement required to be achieved. First of all, the system can't reach the lowest real-time accessing rate even in high performance computer, let alone the embedded system. Secondly, in the whole process, the resolution was reduced to 320×240, while a resolution as high as possible is more preferred by iris processing. At last, a two seconds' complete closure of the eye is used as the criteria of judging drowsiness and this may be not reliable enough.

In this paper we present a simple and effective fatigue detecting algorithm on the basis of [15] and other related

achievements. Instead of using a stable low resolution through the whole process, we adjust the resolution of the input image during different processing stage to obtain a high frame rate, i.e., face detection and eye location will be carried on a much small resized image, while using the eye image with high resolution in iris position and eye state judgment. The PERCLOS parameter is introduced to improve the reliability of the results, and a corresponding method of determine the eye state with the area proportion that the iris takes in its minimum circumscribed circle is proposed.

The rest of the paper is organized as follows. The second chapter gives a brief introduction of the AdaBoosted classifier we adopted for face detection, as showed in [6]. The third chapter shows the way to locate the iris, the method to judge the state of eyes based on the area proportion that the iris takes in its minimum circumscribed circle, the algorithm to analyze drivers' fatigue state based on the value of the PERCLOS parameter, and a brief introduction of the embedded platform. Experimental results are given in chapter IV with detailed analysis. At last, chapter V summarizes this paper.

II. FACE DETECTION BASED ON ADABOOSTED CLASSIFIERS

In this paper, the classification is based on a cascade of boosted Haar-Like features classifiers. As human's face is an ordered combination of eyes, eyebrows, mouth, nose and other organs, it can be picked out by these organs' own characteristics and the positional relationship between each other. The detailed introduction is presented as follows.

A. Haar-Like Features

The Haar-Like feature is a kind of digital image feature, which reflects the grayscale variation of the image. Figure 1 shows several common templates of Haar-Like features. Each template consists both white and black rectangles, and the characteristic value is defined as the difference between the sum of pixels' value located in the white rectangle and the sum of pixels' value located in the black rectangle.

By changing the sizes and positions of the templates, we will get a very large number of Haar-Like features in any sub-window of the image, which needs a very huge amount of computation. In view of this situation, Viola and Jones proposed the use of integral image in the quick calculation of Haar-Like features, and by this way, any size of Haar-Like features can be calculated in a certain period of time.

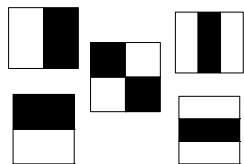


Figure 1. Common templates of Haar-Like features

B. AdaBoosted Classifiers

The classifier based on a single Haar-Like feature is just a weak classifier, which is only a little better than stochastic prediction. AdaBoost is a kind of iterative algorithm, with which a number of weak classifiers can form a stronger one.

The overall shape of an AdaBoosted Classifier is a cascaded decision tree shown in Figure 2. During detecting process, the detecting window whose size is the same with the target object will slide on the tested image, as in [8]. The Haar-Like feature of each sub-window will be calculated and compared to a threshold. After that, the qualified sub-window will be dealt with the next level of classification. Meanwhile, in order to find the target objects with different sizes, the classifier allows to change the size of the input image or directly alter the size of the templates. Therefore, multiple scans of the input image are needed to detect the target objects with unknown size.

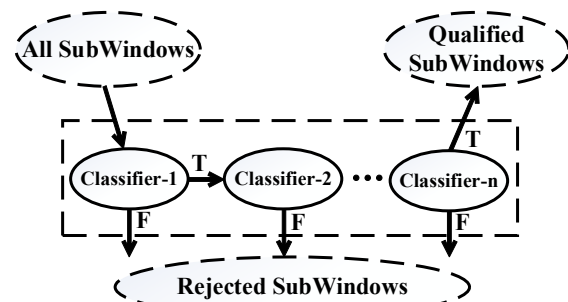


Figure 2. Structure of an AdaBoosted classifier

III. FATIGUE DETECTING BASED ON STATES OF EYES

A qualified fatigue driving detecting system should have the following characteristics:

1) The system must be simple, reliable, and efficient, as it is designed for automotive systems, in which only a few hardware resources can be accessed.

2) The system should focus on typical application scenarios, without the need of covering every possibility. For example, the driver is more likely to be tired after a long driving especially in a long road with little changes, and when driving, he/she will look forward at most time, which means that the camera is more likely to capture the frontal faces. Therefore, the fatigue detection algorithm should focus on frontal faces rather than monitor every extreme condition, which will lead to the artificially increase in complexity and cost.

3) The system could carry on some simple one-time setting parameters to reduce its complexity and increase the individual applicability. For example, based on the fact that for a certain vehicle, its driver is tend to be the same person for a very long period, this paper uses a proportion model whose parameters can be modified, inspired by the face anthropometric properties presented in [16], to pick out the eye image instead of using classifiers, and by this way, the efficiency of the algorithm is greatly improved.

In addition to the above points, the fatigue state in this paper is judged by the PERCLOS parameter, and the essence of the algorithm lies in locating the iris and determining the eye's condition on the basis of face detection. The detailed information about this fatigue driving detecting algorithm is explained as follows.

A. Proposed Fatigue Detection System

Figure 3 illustrates the flowchart of our proposed fatigue driving detecting system. Firstly, after a series of pre-treatments on the color image captured by the digital camera, face detection is carried out. And then, if there exists one face in the image, we can pick out the eye image using a proportion model. Thirdly, we locate the iris accurately with an improved iris positioning algorithm, and analyze the state of the eye by its iris' area proportion. Finally, we can calculate the value of the PERCLOS parameter after a certain time, and according to the result, the driver will be warned if he/she is fatigued.

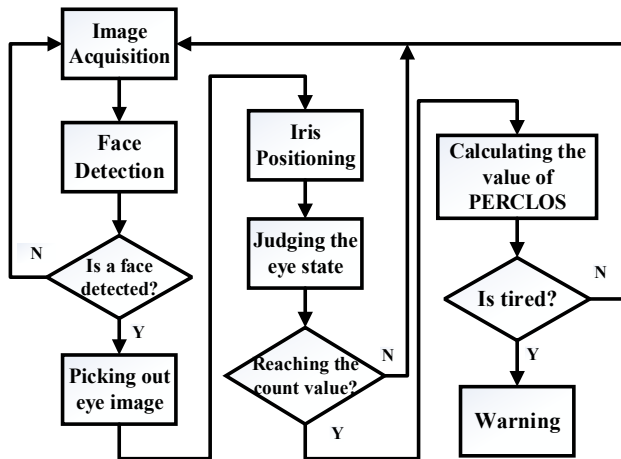


Figure 3. The flowchart of the detection system

B. Face Detection and Eye Positioning

As mentioned in Section I, with the image used in face detection should be as small as possible. To achieve this, optimal parameters of the face detection module provided by OpenCV library are tuned to meet the requirement of practical applications.

During multiple scans, the size of the input images is reduced to detect faces with varying sizes, so the size of the image directly determines the speed of detection. Liu et al. [11] and Cheng et al. [15] resized the image to a fixed low resolution to accelerate the detection, while the results were not satisfactory. So, to achieve our goal, the key point is to choose the optimal sampling rate to get a smallest image.

In the provided AdaBoosted classifier, the minimum possible object size whose default value is 30×30 in the face detection program limits the sampling rate. That is, the size of face region in the resized image must be larger than it. After testing the ORL database, which consists of 400 face photos of 40 different people, with minimum possible object size range from 0 to 45×45 , the results show that the detecting success rate remains 88.75%, which means the minimum possible object size has nothing to do with the detecting result if the object is larger than it. So this parameter may have two other functions. One is to filter the detecting results with too small size, and another is to work as the end of the iteration. The former can be proved with the fact that the size of face region in the resized image must be larger than the minimum

possible object size, but this kind of functions can be easily replaced. To find its relationship with detection efficiency, several experiments were carried out.

1) We tested the module with 8,400 images of 6 different people in various background and illustration, as shown in Figure 6(b), with the minimum possible object size set to 0, to find the relationship among frame rate, detection rate, and sample ratio.

Figure 4 and Figure 5 plot the frame rate and detection rate with respect to the sampling ratio, respectively. As shown in Figure 4, the frame rate increases greatly with the decrease of sampling ratio, which is in line with our intuition. When the ratio is smaller than $1/4$, the frame rate will be greater than 30 fps, which can satisfies the requirement of real-time processing. As illustrated in Figure 5, the detection success rate remains 100% until the sampling ratio reach the value of $1/8$, and there is a sharp decline when the ratio is smaller than $1/8$. So, a larger sampling ratio is preferred to achieve a high frame rate, while too large of it will cause a sharp decrease in detection success rate.

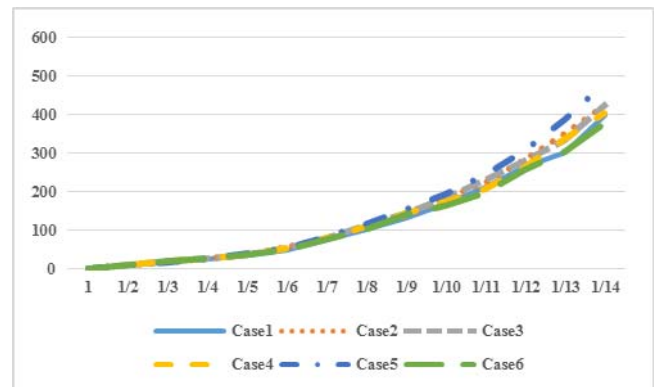


Figure 4. Variation of the frame rate with the sampling ratio

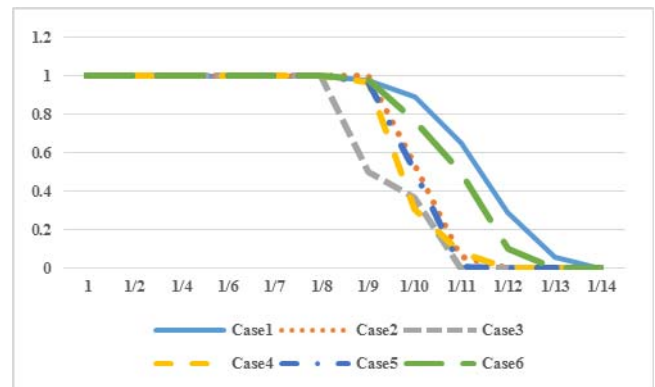


Figure 5. Variation of the detection success rate with the sampling ratio

2) We tested with 600 pictures of 6 different people to find the minimum allowed object size that could be successfully detected with minimum possible object size range from 0×0 to 45×45 . During this test, the sampling ratio was set to $1/8$. The result in Table 1 shows that the minimum detected size is 22×22 .

TABLE 1. THE MINIMUM ALLOWED OBJECT SIZE

Test Case	Case1	Case2	Case3	Case4	Case5	Case6
Min Size	27×27	29×29	26×26	31×31	22×22	28×28

3) We also analysed the relationship between frame rate and minimum possible object size. A database of 6,600 images of 6 people was tested with minimum possible object size change from 15×15 to 25×25 . And we can find in Figure 7 that the frame rate grows with the increase of minimum possible object size, and there will be a jump between 20×20 and 21×21 . Combining the results of 2) and 3), we chose the size of 21×21 to achieve a high frame rate and retain a certain margin.

Database used for these three experiments were taken under a certain simulated driving circumstance showed in Figure 6(a). The camera is 60cm away from people's frontal face and the size of the captured image is 640×480 . As we can see, images used in fatigue driving detecting system has two obvious characteristics: there usually exists only one face in the input image and the driver's face occupies a large proportion of the whole image. From these experiments, we can conclude that a sampling rate of $1/8$ with the minimum possible object size set to 21×21 will satisfy the requirement of high speed and accuracy. With this combination, we can get a frame rate of 133 fps.



Figure 6. Testing backgrounds (a) shows the simulated driving circumstance; (b) shows the tested volunteers

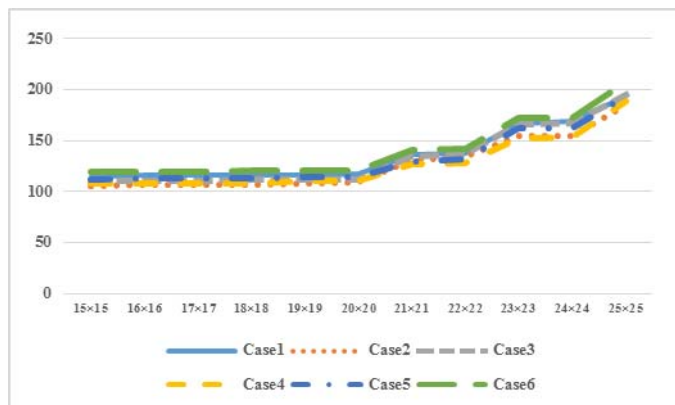


Figure 7. Variation of the detection success rate with the minimum possible object size

When a face is detected, a proportional model learned from [16] with adjustable parameters is used to pick out the eye image. This choice may result in more participation of monitored drivers, but by storing the existing models, only once adjustment is needed for every new driver, while a great save of time consuming could be achieved. An ideal eye region is the one that will always cover the whole eye but refuses other disturbances, such as eyebrows, eyeglasses, etc., and has the tolerance to small movements of heads. An example is showed in Figure 8(a).

C. PERCLOS Parameter

PERCLOS is a physical parameter proposed by Carnegie Mellon Institute after repeated experiments and demonstration. It is defined as the percentage of time during which the eyelid covering a certain proportion of the iris (e.g. [9]). The PERCLOS parameter has three common standards: P70, P80 and EM, which means that the eye state is judged to be closed if the percentage that the eyelid covers over the iris is more than 70%, 80%, and 50%, respectively. And as showed in [10], P80 is considered to have the best correlation with drowsiness.

In this paper, we also choose P80 standard, i.e., the number of positive samples will grow by 1 if the eyelid covers more than 80% of the iris. And the fatigue detecting system will warn the driver while the value of PERCLOS parameter is greater than the threshold which is set to be 0.12, which is learned from [7], in this paper.

D. Iris Positioning and Judgment of Eye State

As mentioned in Section III-C, the eye states can't be simply classified to 'completely open' and 'completely closed', as showed in [15], if using PERCLOS parameter as the judging criterion. And the Support Vector Machine (SVM) is not quite suitable to distinguish the semi-closed state as both the difficulty in acquiring qualified samples and the samples' huge amount. So, to combine this system with PERCLOS parameter, this paper presents a simple, effective algorithm to locate the iris and judge the eye state, as shown in Figure 8(b).

To locate the iris, the eye image should go through an operation flow of grayscale conversion, histogram equalization, binarization, erosion, dilation and Hough circle detection. Among them, the grayscale conversion aims to provide a sample for further process. Histogram equalization can increase the image's contrast, which provides convenience for threshold processing and enhance the robustness to the change of overall light intensity. Binarization is a threshold operation performed on the basis of the fact that the pupil is the darkest in the grayscale image of the eye. Erosion and dilation is carried out to remove the interference of eyelids and eyelashes, while making the iris' characteristics more obvious. Hough circle detection is used to locate the position of the iris.

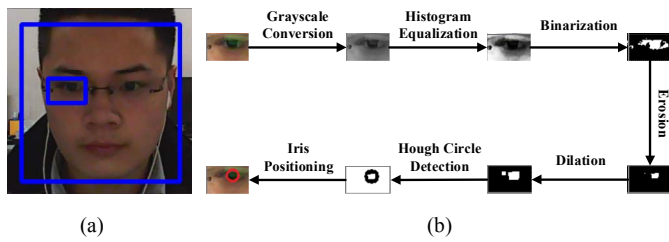


Figure 8. (a) Proportional model; (b) Flow of iris positioning

The directly use of eyeball pixels, as showed in [13], may be a possible method of computing PERCLOS parameter, but the number may varies with the distance between the tested face and the camera. What's more, this method need to calculate the total pixels of the iris with the whole iris exposed to the camera, which may be inconvenient and uncomfortable. Inspired by this method and based on the fact the iris is almost circular, we used the area proportion that the detected iris takes in the Hough circle to do the judgment instead. It is easy to prove that the proportion has good robustness to distance as the iris shape remains unchanged with various distance. The eye state is judged as 'closed' if the proportion is less than 20%.

E. Embedded Platform for Implementation

To make sure the technical practicability, we test our algorithm with embedded platforms which are widely used in automotive systems, such as the GPS guided system.

In this paper, we transplanted our system to the Tiny4412 developing platform, as showed in Figure 9(a), with its corresponding Ubuntu operation system. This board is a charming platform with an attractive price and enough general interfaces such as USB, HDMI, and so on. What's more, as presented in Figure 9(b), this board is equipped with an Exynos4412 processor and two DDR3 memory. And all these elements make it a high-performance and easy-to-use system.



Figure 9. Tiny4412 platform (a) Outlook of the Tiny4412 platform; (b) The core board of Tiny4412 platform

IV. EXPERIMENTAL RESULTS AND ANALYSIS

The algorithm in this paper is developed on VS2013 platform using C++ as programming language. The program is tested on a computer which has an Intel i5-4570 CPU and 8GB DDR3 memory, and the webcam tested is Logitech C170.

We test with six volunteers in the simulated driving circumstance showed in Figure 6(a) using this fatigue detecting system. A total of three hours long videos were catch, and we picked out 12 typical short videos, which were two minutes long. The results is presented in Figure 10. The

tests are ranked in accordance with the actual eye closure ratio, as showed in blue line, while the actual monitored results are given in red line. The yellow region in this figure, of which the middle value is 0.12, represents the critical state, while the red and green region means the driver is fatigue or vigorous, respectively. The test shows that the algorithm works well with naked eyes, and it can locate the iris accurately if there's no strong reflection on eyeglasses.

However, if there exists obvious reflection on the eyeglasses, as showed in test 8~12, the judgment may fail, for we can't determine whether the eye is closed only from the captured image in this circumstance. But we should also notice the fact that the sun visor is the one designed for this situation. What's more, even without glasses, the eyes will reflect to the strong light, which always means the slightly narrowed eyes. In this case, the drivers are easily fatigued and driving may be very dangerous. So, what the detecting system need to do is alarming the drivers.

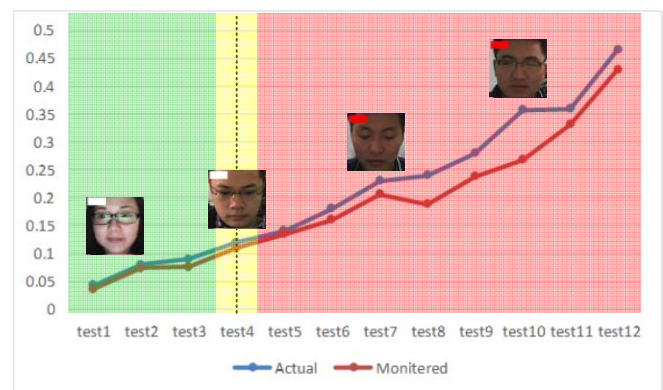


Figure 10. Performance of the proposed system

After the experiment with computers, the system is transplanted to the embedded platform as introduced in Section III-E. The execution time of both system is showed in Table 2 comparing with method in [15]. From it we can find that the frame rate that the proposed algorithm can achieve is almost three times the one of method in [15] and the system is efficient enough to satisfied real-time processing on embedded system.

TABLE 2. THE EXECUTION TIME

	PC	Tiny4412	[15]
Face Detection (ms)	7.496	13.728	12.867
Fatigue Detection (ms)	8.281	15.166	50.216
Total Time(ms)	15.777	28.894	63.083
Frame Rate (fps)	63.384	34.609	19.914

V. CONCLUSIONS

A computer vision based fatigue driving detecting system is presented in this paper. Face detection is accomplished by

Haar-Like features and an AdaBoosted Classifier. The optimal sampling ratio and minimum possible object size are determined by experimental work with different people in various background for both high detection effectiveness and computational efficiency. Moreover, in order to enhance the robustness to the jolt during driving, the state of the eye is judged by the area proportion that the iris takes in its minimum circumscribed circle instead of using the number of pixels in iris. The final result whether the driver is fatigued or not is given by the value of the PERCLOS parameter which is proved to have great relevance with drowsiness. A frame rate up to 34 fps on embedded platform indicates that it fully meets the requirements of real-time processing.

ACKNOWLEDGMENT

Project supported by the National Nature Science Foundation of China (No. 61203269, No. 61305015, No. 11474185, No. 61375084), Postdoctoral Science Foundation of China (No. 2013M541913), the Natural Science Foundation of Shandong Province (No. ZR2011FQ021) and Key Laboratory of Evidence-Identifying in Universities of Shandong (Shandong University of Political and Law) –KFKT (SDUPSL)-201410.

REFERENCES

- [1] M. Peden, et al., *World Report: On Road Traffic Injury Prevention*. World Health Organization, March 1, 2004.
- [2] M.R.Yousefi Zoshk and M.Azarnooosh, "The Measurement and Processing of EEG Signals to Evaluate Fatigue," *Proceedings of 26th Southern Biomedical Engineering Conference SBEC 2010*, 2010, pp. 258–261.
- [3] Y. Takei and Y. Furukawa, "Estimate of driver's fatigue through steering motion," *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, 2005, pp. 1765–1770.
- [4] L.M. Bergasa, et al., "Real-Time System for Monitoring Driver Vigilance," *IEEE Trans. Intelligent Transportation Systems*. 7(1). pp. 63–77, 2006.
- [5] P. Smith, M. Shah and N. da Vitoria Lobo, "Determining driver visual attention with one camera," *IEEE Trans. Intelligent Transportation Systems*. 4(4). pp. 205–218, 2003.
- [6] Songyan Ma and Tiancang Du, "Improved Adaboost Face Detection," *Proceedings of International Conference on Measuring Technology and Mechatronics Automation*, 2010, pp. 434–437.
- [7] Q. Zhu, "Driving Fatigue Detection Based on The Extraction of Images' Detailed Information," M. Eng. thesis, Nanjing University of Science & Technology, Nanjing, China, Jun. 2009.
- [8] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001, pp. 511–518.
- [9] D. Sommer and M. Golz, "Evaluation of PERCLOS based Current Fatigue Monitoring Technologies," *Proceedings of 32nd Annual International Conference of the IEEE EMBS Buenos Aires*, 2010, pp. 4456–4459.
- [10] Wierwille W, et al., *Research on vehicle-based driver status / performance monitoring; development, validation, and refinement of algorithms for detection of driver drowsiness*, Report No DOT HS 808 247, NHTSA, Washington D.C., 1994.
- [11] D. Liu, et al., "Drowsiness Detection Based on Eyelid Movement," *Proceedings of the Second International Workshop on ETCS*, Wuhan, China, 2010, pp. 49–52.
- [12] P. R. Tabrizi and R. A. Zoroofi, "Open/Closed Eye Analysis for Drowsiness Detection," *Proceedings of the First Workshops on Image Processing Theory, Tools and Applications*, Sousse, Tunisia 2008, pp. 1–7.
- [13] W. B. Horng, et al., "Driver Fatigue Detection Based on Eye Tracking and Dynamic Template Matching," *Proceedings of International Conference on Networking, Sensing & Control*, Taipei, Taiwan, 2004, pp. 7–12.
- [14] X. Fan, B. Y and Y. Sun, "Driver Fatigue Detection Using AdaBoosted LBP," *Journal of Information & Computational Science*, vol. 5, no. 6, December 2008, pp. 2501–2509.
- [15] W. Cheng, et al., "A Fatigue Detection System with Eyeglasses Removal," *Proceedings of International Conference on ICACT*, 2013, pp. 331–335.
- [16] P. C. Yuen and C. H. Man, "Human Face Image Searching System Using Sketches," *IEEE Trans. System, Man, and Cybernetics Part A: Systems and Humans*, vol. 37, no. 4, 2007, pp. 493–504.