Research Article

A System of Driving Fatigue Detection Based on Machine Vision and Its Application on Smart Device

Wanzeng Kong,[1] Lingxiao Zhou,[1] Yizhi Wang,[1] Jianhai Zhang,[1] Jianhui Liu,[1] and Shenyong Gao[2]

[1]*College of Computer Science, Hangzhou Dianzi University, Hangzhou 310018, China*
[2]*School of Information Engineering and Art Design, Zhejiang University of Water Resources and Electric Power, Hangzhou 310018, China*

Correspondence should be addressed to Wanzeng Kong; kongwanzeng@hdu.edu.cn

Driving fatigue is one of the most important factors in traffic accidents. In this paper, we proposed an improved strategy and practical system to detect driving fatigue based on machine vision and Adaboost algorithm. Kinds of face and eye classifiers are well trained by Adaboost algorithm in advance. The proposed strategy firstly detects face efficiently by classifiers of front face and deflected face. Then, candidate region of eye is determined according to geometric distribution of facial organs. Finally, trained classifiers of open eyes and closed eyes are used to detect eyes in the candidate region quickly and accurately. The indexes which consist of PERCLOS and duration of closed-state are extracted in video frames real time. Moreover, the system is transplanted into smart device, that is, smartphone or tablet, due to its own camera and powerful calculation performance. Practical tests demonstrated that the proposed system can detect driver fatigue with real time and high accuracy. As the system has been planted into portable smart device, it could be widely used for driving fatigue detection in daily life.

1. Introduction

Driving fatigue is a common phenomenon due to long time of driving or lack of sleep. It is a significant potential hazard in traffic safety. As many as 100,000 traffic accidents caused by driving fatigue which led to 400,000 people injured and 1550 people killed happened in the United States each year [1]. Research on driving fatigue detection is becoming a popular issue all over the world.

Currently, detection methods of driving fatigue can be mainly divided into four categories [2]. (1) First category is methods based on drivers' physiological signal [3, 4]. Such kind of physiological signal includes electroencephalograph (EEG), electrocardiograph (ECG), and electrooculogram (EOG). These methods usually result in good performance of fatigue detection. However, how to conveniently obtain clean signals is a problem to be solved in practical applications [5–7]. (2) Second category is methods based on drivers' operation behavior. Literatures reported that driving fatigue can be detected through drivers' operation such as steering

wheel operation [8, 9]. When the driver is falling into fatigue, he will reduce the grip strength of steering wheel or decrease the ability of controlling the steering wheel [10]. (3) Third category is methods based on vehicle state. The trail of vehicle and lane departure information are also extra useful information for detecting fatigue [11]. Both trail and lane information are correlated with controlling steering wheel; hence, they also reflect the driver's operation but are noncontacted. (4) Fourth category is methods based on drivers' physiological reaction. Fatigue can be detected by physiological behavior such as blinking and yawning, among which the most effective method is based on detection of eye states [12–15]. Generally speaking, the frequency and duration of eye closed-state will increase and those of eye open state will decrease when drivers become fatigued. Eriksson and Papanikolopoulos [16] proposed a method that eye states can be recognized by a camera fixed on dashboard. And drivers' fatigue can be detected by recognizing closed-state in continuous 2 or 2.5 seconds. Furthermore, when fatigue occurs, people often yawn and their mouths will open

obviously, so the detection of mouth state by a camera is also an effective method for detecting driver fatigue. Shi et al. [17] used BP neural network and computer vision to detect mouth states to estimate driver's mental state.

However, all above methods must use some special and extra equipment. For example, methods based on drivers' operation behavior [18] require pressure sensor [19] and angular transducer to detect driver's behavior on steering wheel. Methods based on both vehicle state and drivers' physiological reaction require cameras to record inside and outside conditions of the car. In particular, the methods based on physiological signal require more expensive and comprehensive EEG acquisition equipment such as EEG cap, electrodes, and signal amplifier. Moreover, all these methods usually need an extra computer or embedded computing board for signal processing and making decisions.

Nowadays, tablet and smartphone are so popular that almost every driver holds one. And most of them are equipped with excellent camera and powerful computing/processing unit. The main contribution of this paper is that we proposed an improved strategy and practical system to detect driving fatigue using smart device. Firstly, a practical machine vision system based on improved Adaboost algorithm is developed to detect driving fatigue by checking the eye state in real time. Then, this system is easily transplanted into a smart device such as a tablet to perform the task of fatigue detection. Practical tests demonstrate that it is convenient and of low cost to detect driving fatigue by smart device, but it is as effective as other methods.

The rest of this paper is organized as follows: Section 2 provides a detailed description of system including the Adaboost algorithm, face detection, eye localization, state recognition, and also the whole detection strategy. Experiment results and data analysis are presented in Section 3. The system transplantation is given in Section 4. Finally, the conclusion is presented in Section 5.

2. Adaboost Algorithm and Improved System

The system is composed of four parts, that is, image preprocessing, face detection, eye state recognition, and fatigue evaluation. In this system, images are obtained by external camera with high resolution which is placed on front left of drivers. The first step is to denoise the images. Then, after detecting the face region, eyes location and states can be obtained easily and quickly based on the detected face region. Finally, drivers' fatigue can be detected by analyzing the square wave diagram of eye states in real time.

2.1. Image Preprocessing. Naturally, video images are always contaminated by noise in different degree which roots in several factors, such as driving condition, underexposure, overexposure, or nonlinearity of devices. Original images are usually converted to gray scale images to be easily used in classifiers. Images are easily malfunctioned, including lacking contrast and blurring. Thus, histogram equalization is the next important step. The goal of histogram equalization is to highlight the features by enhancing contrast of gray scale images and reducing interference caused by the asymmetric illumination.

2.2. Adaboost Algorithm and Classifier Training. Adaboost algorithm is a kind of boosting algorithm proposed by Freund and Schapire [20], which selects some weak classifiers and integrates them into strong classifiers automatically.

Viola and Jones [21] proposed an Adaboost algorithm based on Haar characteristics and trained frontal face classifier with this algorithm. The algorithm is with high detection accuracy and faster than almost all the other real-time algorithms. The excellent performance of frontal face classifier contributes a lot to the Adaboost algorithm.

Steps of Adaboost Algorithm. Select training samples as follows: $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, and x_i is the ith training sample, while $y_i \in \{0, 1\}$ represents positive or negative samples, respectively. The positive samples could be used to train classifier to identify similar characteristics, the negative samples used to eliminate differences characteristics. The number of positive samples is s and the number of negative samples is m, $n = m + s$.

(1) Initialize the weight of each sample: $w_{1,i} = 1/n$, $i = 1, 2, \ldots, n$.

(2) Repeat the following four steps, for $t = 1, 2, \ldots, T$ (T is the optimal number of weak classifiers).

   (a) Normalize the weight into a probability distribution:

$$q_{t,i} = \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,i}}. \tag{1}$$

   (b) For each feature f, train a weak classifier h_f, and compute weighted error rates ε_f of weak classifiers corresponding to all features:

$$\varepsilon_f = \sum q_{t,i} \left| h_t(x_i) - y_i \right|. \tag{2}$$

   (c) Choose the best weak classifier $h_t(x)$ from step (b) which has the minimum error rate ε_t.

   (d) According to the best weak classifier, adjust weight:

$$w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}. \tag{3}$$

   $e_i = 0$ represents correct classification; $e_i = 1$ represents misclassification.

(3) Form the strong classifier eventually:

$$h(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & \text{others} \end{cases} \tag{4}$$

where $\alpha_t = \log \dfrac{1 - \varepsilon_t}{\varepsilon_t}$.

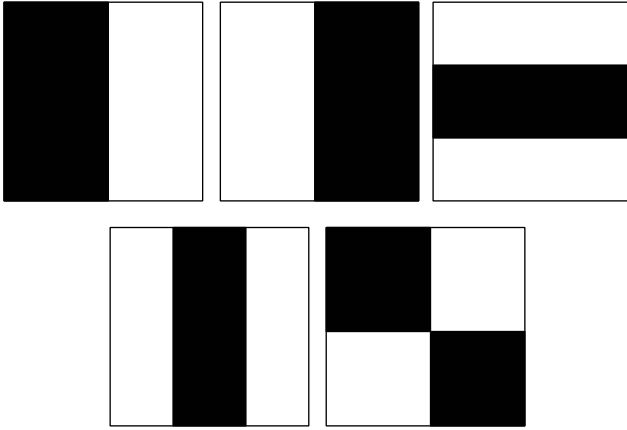Figure 1: Haar characteristics.

Figure 2: Face library and constructed eyes library.

In this paper, Haar features are used to train Adaboost classifier. Extraction of Haar characteristics and calculation of features are two important aspects of Adaboost algorithm. Figure 1 shows five simplest kinds of gray rectangle features. There are 160,000 rectangle features in a size of 24×24 pixels, which shows its complexity and diversity. The specific computing method is to subtract the number of pixels in black rectangles from that in white rectangles. Viola and Jones [21] illustrated that the speed of training and detection will be greatly enhanced by applying integral image to feature computation. In Viola's method, rectangle features can be obtained by computing several points of integral image. Integral image is defined as follows: for a point (x, y) in an image, its integral image is $ii(x, y)$:

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y'), \quad (5)$$

where $i(x', y')$ is the gray value of the point (x, y).

A large number of collections of target and nontarget images are used to build positive and negative samples library when training a specific target classifier.

In this paper, we use face images from MIT, Yale, and ORL face databases to train the face classifiers. In total, 3000 samples of front faces and 1500 samples of deflected faces are used and normalized into a resolution of 20×20.

For eye detection, this system also trained the eyes-open and eyes-closed classifiers with Adaboost algorithm. However, there is no public human eyes library available for both positive and negative samples of eyes. Hence, we established an eye library by ourselves. In Figure 2, a large number of single eyes can be obtained through processed images in face library. They are classified into an opened-eyes library and a closed-eyes library. Eventually, the numbers of open-eyes and closed-eyes samples are 1190 and 700, respectively, with normalized resolution of 15×7 pixels.

After establishing the eye library, Adaboost algorithm consumes most of its time on training the target classifiers, which shows a good precision and real-time performance in detection.

2.3. Improved Detection Strategy.

In this paper, Adaboost algorithm proposed by Viola is employed to train face classifiers and detect face. However, traditional methods are only focusing on front face samples. If the deflected angle of driver's face is small, the algorithm can be well applied to face detection in this condition. But classifier would be failed to capture the target when driver's face deflected in a large angle.

In view of the above shortcomings, we improved the face detection stratagem based on Viola's method. Left and right deflected face classifiers can be obtained by training left and right deflected face samples. If the front face detection is failed, the other two deflected-face classifiers will be involved in the detection task. In the worst case, there will be 3 times of detections for deflected faces. Hence, if face classifiers are constantly dealing with such kind of situation, the speed of face detection will be obviously reduced.

To deal with this problem, our system optimizes scheduling algorithm of face classifiers. Specifically, when the system misses the target using the front face classifier, right deflected face classifier is called firstly to re-detect. If it succeeds, the right deflected-face classifier will be used in next frame by default. Otherwise, the left deflected face classifier is called to work. If detection succeeds, similarly, left deflected-face classifier will be used default in next frame. The front face classifier is called to retest when both left and right face classifiers lose the target. This strategy consumes a little more time on the detection of frames which lose targets but improves the real-time performance of whole system.

In Adaboost algorithm, face detection region is obtained by searching the whole picture. The larger the picture, the longer the time of searching which will be consumed. In order to improve the detection speed, each frame of picture is downsampled into a quarter of the original picture. In practical application, reasonable scaling of pictures has little effect on accuracy of detection or precision of localization. Compared to the traditional style, the speed can be increased by half or more to meet the requirement of real time.

2.4. Eye Localization and State Recognition.

The common methods of eyes localization include integral projection, Hough transition, template matching, and principal component analysis. In this paper, Adaboost algorithm is exploited to train open-eyes and closed-eyes classifiers for locating eyes.

Eye localization is carried out in the detected face region. Face is a regular geometry, where each organ presents regular distribution. In order to improve precision and efficiency of detection, face region can be segmented before locating eyes.

Figure 3(a) shows face regions detected by face classifiers. Face can be divided by the following analysis.
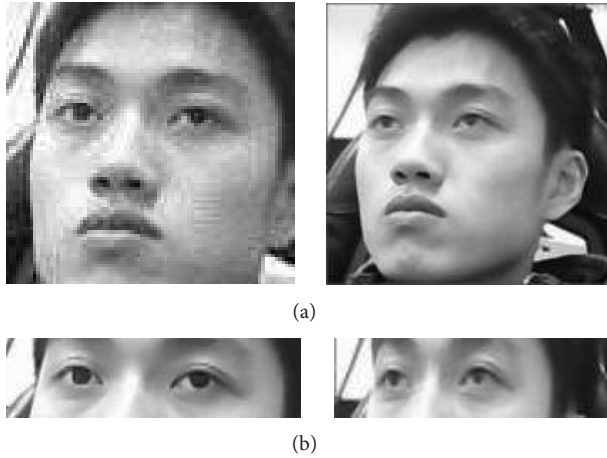
(a)

(b)

FIGURE 3: Estimation of eyes candidate region.

(1) The height of eyes region can be constrained to less than 1/3 height of the face and more than 1/10 height of the face.

(2) The width of single eye region is less than 1/2 width of the face and more than 1/4 width of the face.

To guarantee the eyes classifiers' performance without the influence of face segmentation, we make the following constraints:

(1) $y = 5H/16$,

(2) $h = 5H/24$,

where y and h represent the highest pixel and the height of estimated eyes region, respectively, and H represents height of face region.

Figure 3(b) shows the result after partition, and it illustrates that the region in face which is below eyebrows and above nostril can be used as eyes candidate region. In this region, eyes are detected by open-eyes and closed-eyes classifiers. This method avoids regions of nostril, mouth, and above forehead that could affect human eyes detection. In practical test, it obviously reduces the error probability of detection.

Open-eyes classifier is firstly used to detect target area. The results will be marked by red rectangles if open-eyes state is detected. If it loses the target, the closed-eyes classifier will be switched to work. And the detection result will be also marked by yellow rectangles for closed-eye state. If close classifier loses target, this frame will be regarded as an exception of eye detection.

2.5. Fatigue Judgment with Multiple Indexes. PERCLOS is the percentage of duration of closed-eye state in a specific time interval (1 min or 30 s) [22]. PERCLOS is a well-recognized and effective measure of neurophysiological fatigue level. For each frame, a result will be output at the end of detection. The result could be open-eyes state, closed-eyes state, face exception, and eyes exception. The RERCLOS is drawn in real time by counting detection results with certain frames
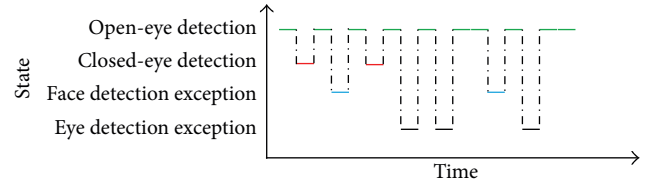

FIGURE 4: PERCLOS of human eyes state.

in a fixed period as shown in Figure 4. Then fatigue can be detected by analyzing PERCLOS and duration of closed-eyes state.

Only two eyes states (open and closed) can be detected in this system. The level of open-eyes state is not analyzed. PERCLOS will be simplified based on the percentage of duration of completely eye closed in 30 seconds. The formula of simplified PERCLOS value is as follows:

$$PERCLOS = \frac{t}{30} \times 100\%, \qquad (6)$$

where t is the duration that eyes are completely closed.

The duration of closed-eyes state in driver's blink will increase when fatigue occurs. In this paper, the duration of 1.2 seconds is the threshold which is used as judging fatigue state for drivers. Hence, driver's fatigue can be detected with two indexes which are PERCLOS index and the threshold of 1.2 s duration.

A driving fatigue detection system is developed by using PERCLOS and eye states as shown in Figure 5. The system will detect fatigue and ring the bell when one of the two indexes satisfies the requirement.

3. Experiment and Data Analysis

Our system runs on a PC with Intel Core(TM)2 Duo 2.10 GHz CPU and 2 GB RAM, using the third-party library OpenCV to perform system tests in Visual Studio 2008. Racing game was used as driving conditions. Nine subjects took part in the experiment with simulate driving environment. The resolution of these videos is 352×288 pixels and the frame rate is 25 fps. Each frame contains human face.

We recorded ten groups of video streams in this experiment. The videos are divided into two categories. The first category (i.e., 1st–4th groups) shows various facial expressions which occurred in three subjects' simulate driving experiments. There are 1630 frames in each group of videos. These videos are used to test the performance of face detection and eyes detection.

The second category consists of the remaining 6 groups (i.e., 5th–10th). Six subjects were asked to implement driving tasks for long enough time to become fatigued finally. These videos are used to extract fatigue index. In order to collect valid videos for assessment of driver fatigue, subjects are involved in training and experimental sessions. The whole experiment duration for one subject may last two or more days, it depends on his training performance. Before experiment, subjects are asked not to eat chocolate and drink coffee or alcohol. The length of each video is about 70 minutes.
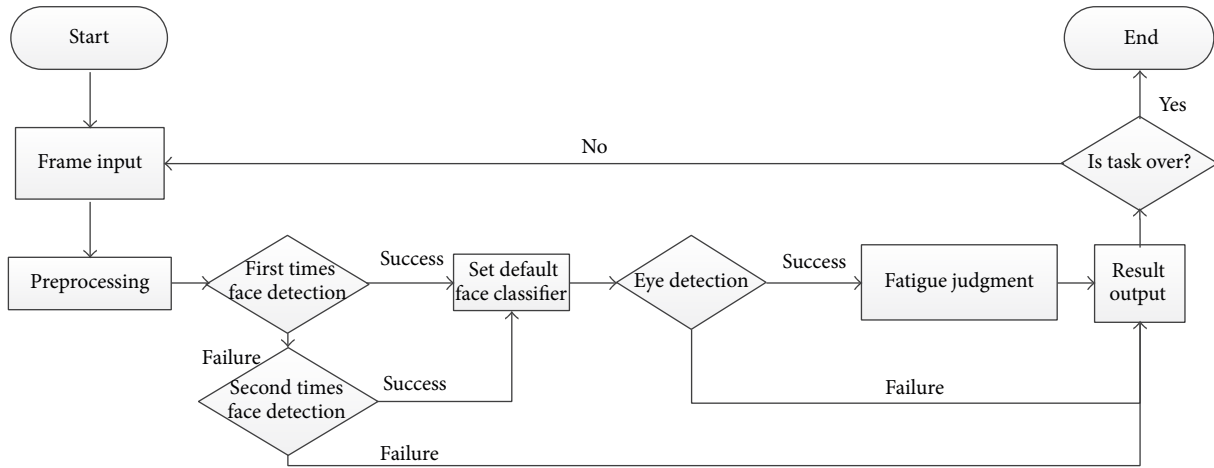
Figure 5: System flow chart.

Table 1: Experimental results.

Detected target	Target (frames)	Correct detection (frames)	Missed detection (frames)	Detection rate (%)
Human face	6520	6214	293	95
Open state	5392	4912	480	91
Close state	1128	1015	113	90

There are 105,000 frames in each group of videos. In the first 55 minutes of experiment, subjects are asked to take intensive simulate driving. Many curves and steep slopes are presented in driving conditions. And extra tasks of alert and vigilance (TAV) [23] were exerted in order to ensure subjects concentrating on driving highly during the experiment. In the last 15 minutes, we relieved subjects' stress by declining missions and using a flat road with fewer curves. The monotony of driving induced driver to be fatigued. Obvious features of fatigue for subjects in this simulate driving can be summarized as increased blink times, blinks frequency, and duration of closed-eyes state.

3.1. Detection Performance of Our Method.

Figure 6 shows the results of human face detection. In order to distinguish detection results, the result of front face classifier was marked as a white rectangle and the result of deflected face classifier was marked as a green rectangle. The first row shows the results which were detected by the front face classifier marked with white rectangles. Face was lost because the right deflection angle of face is too large in 3rd frame. The second row shows the results which were detected by deflected face classifiers marked with green rectangles. And in the second row, face was lost by the deflected classifier in the first and fourth frame because the faces were in the state of looking straight ahead. The third row shows the results detected by the two classifiers. Hence, all faces including front and deflected conditions were detected successfully. This method can carry out real-time detection with a high accuracy.

Figure 7 shows the results of eyes localization and the state recognition under different expressions. Eyes are marked with rectangles; specifically, closed-eye state was

targeted with yellow rectangle. And open-eye state was targeted with red rectangle.

In this section, we also revealed the general performances of the proposed method in terms of face detection, eyes localization, and state recognition. We perform our method on the first category of videos (i.e., 1st–4th groups) which contained 6520 frames consisting of open-eye state and close-eye state. The correct detected frames are manually counted. Detail results are presented in Table 1. From Table 1, we can conclude that the proposed system works quite well with accuracy of at least 90%.

3.2. Comparison Experiment.

Compared with the traditional Adaboost method that trains eye classifiers to detect eye state directly, we proposed an improved strategy to detect eye state. The proposed strategy mainly includes 3 steps and 2 categories of classifiers corresponding to face and eye, respectively. Specifically, we first detect face efficiently by classifiers of both front face and deflected face. Then, candidate region of eye is determined according to geometric distribution of facial organs. Finally, trained classifiers of open eyes and closed eyes are used to detect eyes in the candidate region quickly and accurately. Table 2 shows the comparison of elapsed time for 2 methods in PC system. The shortest and longest time for processing a single frame are also presented. Concerning our method, although it needs to detect both face and eye, the average processing speed of the system is up to 20 fps. Hence, the system has a good performance of real time. The longest processing time of single frame is more than 2 times of the shortest one, because it takes at least 2 times to detect face and eye state due to the situation of deflected face. Concerning the traditional

Figure 6: Results of face detection.

Figure 7: Results of eyes location and state recognition.

TABLE 2: Comparison of time consumption in PC system (ms).

Methods	Mean time consumption of several detection steps						
	Front face	Deflected face	Open eye	Close eye	Other	Single Frame (longest)	Single Frame (shortest)
Our method	16	18	5	4	12	73	33
Traditional method	—	—	96	97	—	245	93

FIGURE 8: Comparison results with our method and the traditional method. The frames of top row are detected by direct Adaboost training eye classifier, while the frames of bottom row are detected by our improved method.

method, in spite of detecting eye with only one step directly, it costs more time to detect eyes. The reason is that it needs to use eye classifier searching in the whole frame instead of a small candidate region as our method.

Figure 8 shows comparison of detection performance between our method and the traditional method. Figure 8 illustrates that our method can detect eyes precisely in the restricted face region. However, the traditional method employed the eye classifier to detect eye in the whole picture. As a result, there are a lot of false detections. Based on the comparison experiment, we can conclude that our method outperforms the traditional method in terms of detection speed and accuracy.

3.3. Determining the Fatigue Index. In order to judge the fatigue state automatically, we should fix the parameter of threshold to determine alert or fatigue. There is an obvious change of PERCLOS values at the beginning and the end of driving experiment. The subject's blink frequency and closed-eye state duration will increase after subjects become fatigued. Figure 9 shows PERCLOS values of six subjects in the periods of 10~17 (the beginning) and 63~70 minutes (the end). From Figure 9, we can conclude that PERCLOS values rise significantly at the end of experiment, and the escalating rate of mean values is more than 200%. Additionally, six subjects' PERCLOS values showed individual differences. Subjects' mean values, such as SUB1, SUB2, SUB4, and SUB5, are relatively low (<0.025) both in the beginning and in the end of experiment. But for some subjects, such as SUB3 and SUB6, the PERCLOS mean values are relatively higher than other subjects. For example, the PERCLOS values of SUB3 and SUB6 at the beginning and the end are 0.016, 0.042, 0.059, and 0.146, respectively.

Variation of PERCLOS mean values of each subject are presented in Table 3. In this table the PERCLOS value shows individual differences. Although the PERCLOS values of SUB3 and SUB6 are extremely large, their escalating rates keep quite stable as the other subjects. From Table 3, we find that the escalating rate of all subjects is between 2.0 and 3.0 except SUB1. Thus, it should be a good choice to treat the escalating rate of PERCLOS value as an index for evaluating driver fatigue. In this paper, we judge the driver as fatigued when his PERCLOS increment rate is more than 2.

4. Fatigue Detection on Smart Device

Nowadays, smartphones and tablet devices have quite powerful processing capacity, and most of these devices are also equipped with high resolution camera (front or rear). Such kinds of smart devices are installed with operating system, such as Android and iOS. Especially for Android operating system, it is a free and open system; hence, it is very convenient to do secondary development for users. Furthermore, OpenCV is compatible with these Android-based devices. Therefore, it is feasible to transplant our fatigue detection system based on machine vision into smart devices. As a result, we designed a Driving Fatigue Detection Warning System (DFDWS) on the Android smart device with machine vision. The composition of detection system is shown in Figure 10.

This system is developed based on Android OS and OpenCV Library which are called OpenCV4Android. OpenCV4Android is available as a java lib on the Internet. It is easy to install and configure OpenCV4Android SDK on the smart handheld device according to the tutorial. OpenCV4Android provides an interface named
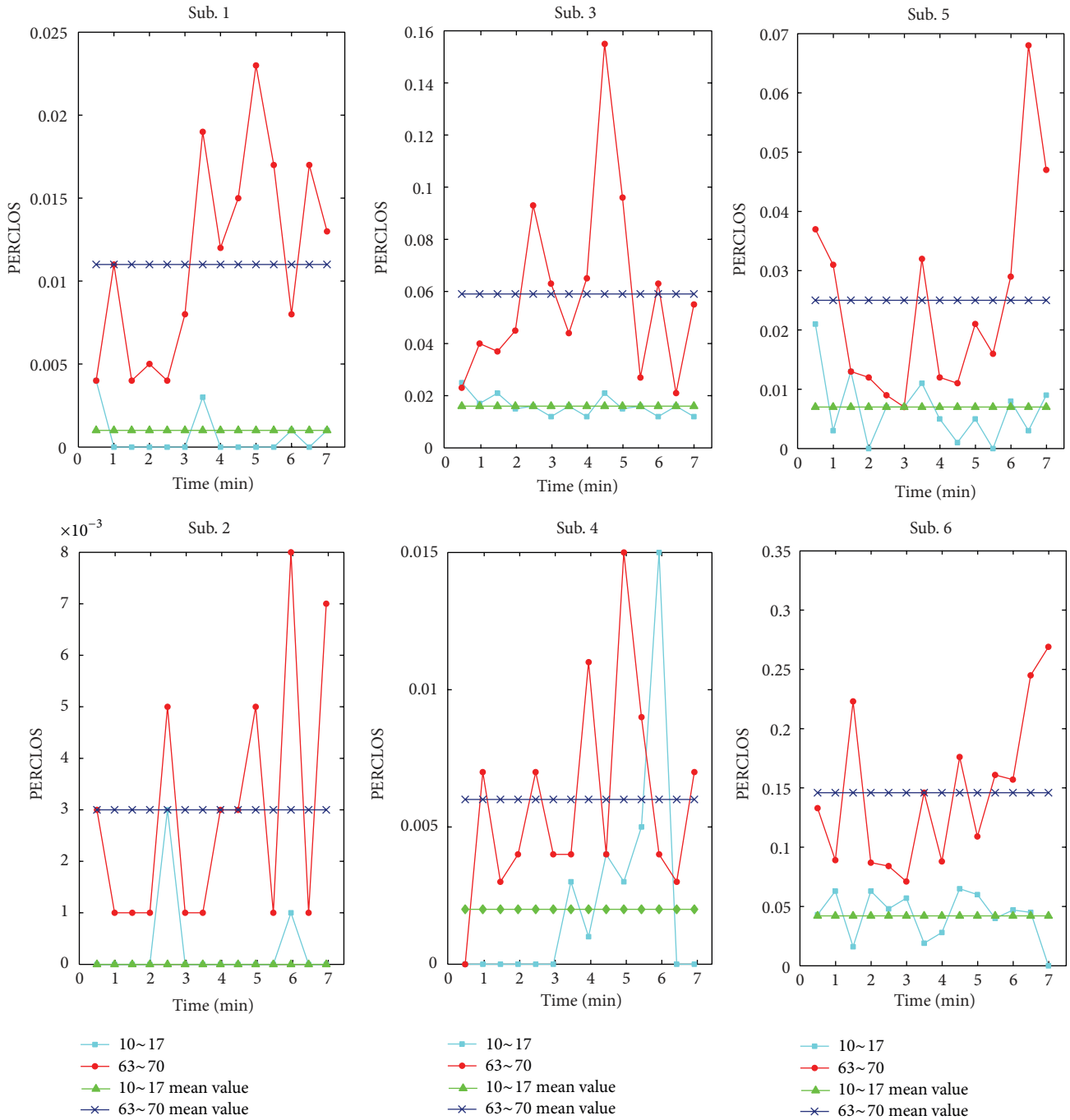
FIGURE 9: PERCLOS values in the beginning (10~17 m) and the end (63~70 m) of driving experiment.

TABLE 3: Mean values of PERCLOS in different stages in 6 subjects.

PERCLOS	Subjects						
	Sub. 1	Sub. 2	Sub. 3	Sub. 4	Sub. 5	Sub. 6	Mean
10~17 m (at the beginning)	0.001	0.007	0.016	0.001	0.002	0.042	0.012
63~70 m (at the end)	0.011	0.025	0.059	0.003	0.006	0.146	0.042
Difference	0.010	0.018	0.043	0.002	0.004	0.104	0.030
Escalating rate	10.000	2.571	2.688	2.000	2.000	2.476	3.623

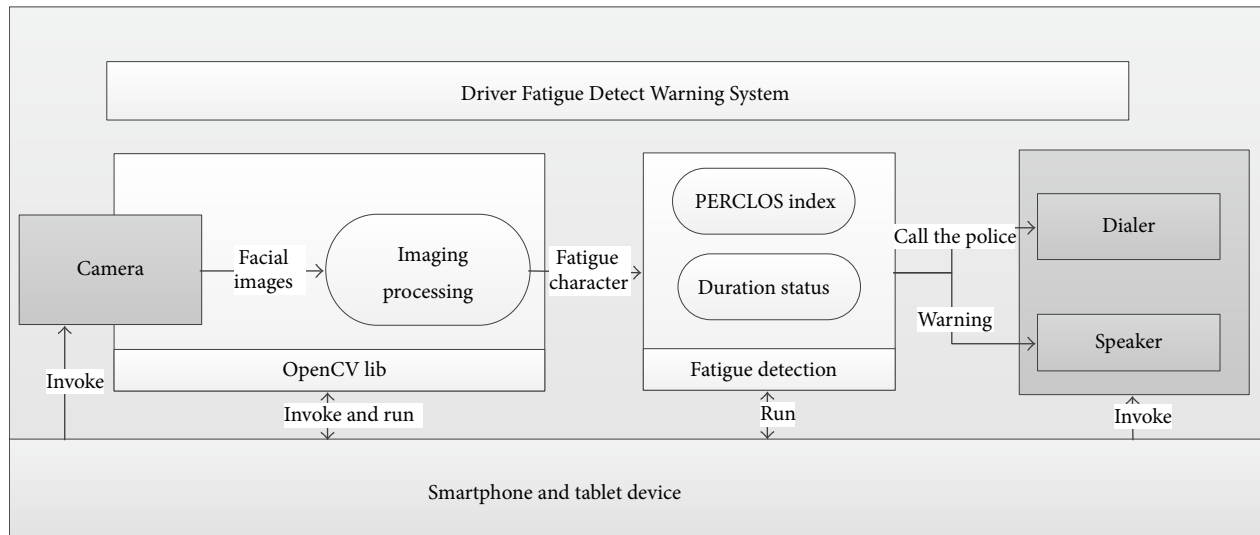FIGURE 10: Detection system composition.

(a)   (b)

FIGURE 11: Detection result on tablet.

CvCameraViewListener to catch camera frames, and a class *CascadeClassifier* to detect object. In our application, we create several *CascadeClassifier* classes, which are implemented as different classifiers corresponding to face, open-eye, or closed-eye. When a frame is captured, the method *onCameraFrame* will be invoked automatically. In the method, frame image will be translated into class *Mat* which is represented as images in OpenCV. Then, the method *detectMultiScale* will be invoked by *CascadeClassifier* to detect face feature.

Thus any Android phone or tablet, with a camera, dialer, and speaker, could transplant this system. DFDWS use OpenCV library to call the camera to collect facial images. The image frames in the memory constitute an image stream; then, the proposed algorithm would deal with it. After that, the system starts to judge driver's fatigue state according to PERCLOS or duration time of closed-eye state. If the driver falls into fatigue state, then the system would turn on the speaker to remind him/her or dial the emergency center.

Also, detection result would be shown on the screen in time. The interface of the proposed transplanting system and detection result are shown in Figure 11.

In this paper, our system runs well on a Nexus7 tablet which is with 1 GHz CPU and 1 GB RAM. In the beginning of the experiment, the system would collect a period of 10 min to calculate the PERCLOS index as a baseline for judging fatigue state. During this period, drivers should adjust device and body posture to make sure that the system could accurately identify faces and eyes. The index of PERCLOS escalating rate corresponding to baseline is used to detect fatigue in smart handheld system. When PERCLOS escalating rate increased more than 200%, the system on Nexus7 tablet will judge the driver as fatigued. Table 4 shows the detection performance on Nexus7. It demonstrates the average consumption of kinds of classifiers and also presents the shortest and longest time of processing a single frame. Compared to the detection performance on the PC system (see Table 2), we can find that the consumption time for each step is extended for Nexus7 tablet. Because the main frequency of the tablet is much lower than the PC (2.1 GHz), accordingly, the longest and the shortest processing time for single frame are also extended. For normal situation, that is, the driver being almost in front face, the performance of DFDWS on Nexus7 tablet can achieve the speed of 14 frames/sec. That means mobile device

TABLE 4: Average consumption of each step on tablet system (ms).

Running items	Consuming time
Front face detection	39
Left and right deflected face detection	49
Open state detection	8
Close state detection	4
Other time consumptions	36
The longest processing time of single frame	98
The shortest processing time of single frame	70

can smoothly run this application. But for the worst case such as the situation of side face, the system performance will decrease to 10 frames/sec.

5. Conclusions

This paper provides a practical driving fatigue detection system based on Adaboost algorithm. We proposed a new strategy to detect eye state instead of detecting eye directly. In our detection strategy, we first detect face efficiently using classifiers of both front face and deflected face. Then, candidate region of eye is determined according to geometric distribution of facial organs. Finally, trained classifiers of open eyes and closed eyes are used to detect eyes in the candidate region quickly and accurately. As a result, PERCLOS escalating rate could be calculated and used as the index of fatigue. When the PERCLOS escalating rate increased more than 200%, the driver could be considered in fatigue state. Moreover, this paper implemented a Driving Fatigue Detection Warning System and transplanted it on Nexus7 tablet. The system makes decision of fatigued or not according to PERCLOS and duration of closed-eyes state. Experiments demonstrated that the proposed system has a high accuracy. Meanwhile, the processing speed can reach 30 fps on PC and 14 fps on tablet, which meets the requirement of real time.

Of course, this system could make further improvement on accuracy and speed of detection by using discrete cosine coefficients [24] and covariance feature [25], respectively. In addition, this paper has dodged the conditions under poor illumination. It should be perfected in the future research.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Authors' Contribution

Wanzeng Kong and Lingxiao Zhou contributed equally to this work.
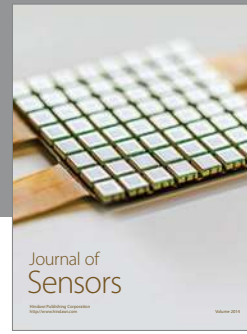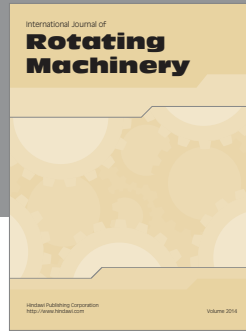
Acknowledgment

References

[1] S. R. Paul, *NHTSA's Drowsy Driver Technology Program*, 2005.

[2] L. Wang, X. Wu, and M. Yu, "Review of driver fatigue/drowsiness detection methods," *Journal of Biomedical Engineering*, vol. 24, no. 1, pp. 245–248, 2007.

[3] G. Borghini, L. Astolfi, G. Vecchiato, D. Mattia, and F. Babiloni, "Measuring neurophysiological signals in aircraft pilots and car drivers for the assessment of mental workload, fatigue and drowsiness," *Neuroscience and Biobehavioral Reviews*, vol. 44, pp. 58–75, 2014.

[4] T. Myllylä, V. Korhonen, E. Vihriälä et al., "Human heart pulse wave responses measured simultaneously at several sensor placements by two MR-compatible fibre optic methods," *Journal of Sensors*, vol. 2012, Article ID 769613, 8 pages, 2012.

[5] M. Simon, E. A. Schmidt, W. E. Kincses et al., "EEG alpha spindle measures as indicators of driver fatigue under real traffic conditions," *Clinical Neurophysiology*, vol. 122, no. 6, pp. 1168–1178, 2011.

[6] S. K. L. Lal and A. Craig, "Reproducibility of the spectral components of the electroencephalogram during driver fatigue," *International Journal of Psychophysiology*, vol. 55, no. 2, pp. 137–143, 2005.

[7] B. T. Jap, S. Lal, P. Fischer, and E. Bekiaris, "Using EEG spectral components to assess algorithms for detecting fatigue," *Expert Systems with Applications*, vol. 36, no. 2, pp. 2352–2359, 2009.

[8] J. Krajewski, D. Sommer, U. Trutschel et al., "Steering wheel behavior based estimation of fatigue," in *Proceedings of the 5th International Driving Symposium on Human Factors in Driver Assessment, Training and Vehicle Design*, pp. 118–124, 2009.

[9] Y. Wu, J. Li, D. Yu, and H. Cheng, "Research on quantitative method about driver reliability," *Journal of Software*, vol. 6, no. 6, pp. 1110–1116, 2011.

[10] A. Eskandarian and A. Mortazavi, "Evaluation of a smart algorithm for commercial vehicle driver drowsiness detection," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 553–559, June 2007.

[11] C. X. Huang, W. C. Zhang, C. G. Huang, and Y. J. Zhong, "Identification of driver state based on ARX in the automobile simulator," *Technology & Economy in Areas of Communications*, vol. 10, no. 2, pp. 60–63, 2008.

[12] Z. Zhang and J.-S. Zhang, "Driver fatigue detection based intelligent vehicle control," in *Proceedings of the 18th International Conference on Pattern Recognition (ICPR '06)*, vol. 2, pp. 1262–1265, IEEE, Hong Kong, August 2006.

[13] Q. Wang, J. Yang, M. Ren, and Y. Zheng, "Driver fatigue detection: a survey," in *Proceedings of the 6th World Congress on Intelligent Control and Automation (WCICA '06)*, vol. 2, pp. 8587–8591, Dalian, China, 2006.

[14] Z. Zhang and J. Zhang, "A new real-time eye tracking for driver fatigue detection," in *Proceedings of the IEEE 6th International Conference on ITS Telecommunications*, pp. 8–11, June 2006.

[15] R. Parsai and P. Bajaj, "Intelligent monitoring system for driver's alertness (a vision based approach)," in *Knowledge-Based Intelligent Information and Engineering Systems*, vol. 4692 of *Lecture Notes in Computer Science*, pp. 471–477, Springer, Berlin, Germany, 2007.

[16] M. Eriksson and N. P. Papanikolopoulos, "Eye-tracking for detection of driver fatigue," in *Proceedings of the IEEE Conference on Intelligent Transportation Systems (ITSC '97)*, pp. 314–319, 1997.

[17] S. M. Shi, L. S. Jin, R. B. Wang, and B. L. Tong, "Driver mouth monitoring method based on machine vision," *Journal of JiLin University*, vol. 34, no. 2, pp. 232–236, 2004.

[18] W. Wang, J. Xi, and H. Chen, "Modeling and recognizing driver behavior based on driving data: a survey," *Mathematical Problems in Engineering*, vol. 2014, Article ID 245641, 20 pages, 2014.

[19] Q. Tan, M. Yang, T. Luo et al., "A novel interdigital capacitor pressure sensor based on LTCC technology," *Journal of Sensors*, vol. 2014, Article ID 431503, 6 pages, 2014.

[20] Y. Freund and R. E. Schapire, "A desicion-theoretic generalization of on-line learning and an application to boosting," in *Computational Learning Theory*, vol. 904 of *Lecture Notes in Computer Science*, pp. 23–37, Springer, Berlin, Germany, 1995.

[21] P. Viola and M. J. Jones, "Robust real-time object detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[22] D. F. Dinges and R. Grace, *PERCLOS: A Valid Psychophysiological Mesure of Alertness As Assessed by Psychomotor Vigilance*, Federal Highway Administration, Office of Motor Carriers, 1998.

[23] W. Kong, Z. Zhou, L. Zhou, Y. Dai, G. Borghini, and F. Babiloni, "Estimation for driver fatigue with phase locking value," *International Journal of Bioelectromagnetism*, vol. 14, no. 3, pp. 115–120, 2012.

[24] J. P. Dukuzumuremyi, B. Zou, C. P. Mukamakuza, D. Hanyurwimfura, and E. Masabo, "Discrete cosine coefficients as images features for fire detection based on computer vision," *Journal of Computers*, vol. 9, no. 2, pp. 295–300, 2014.

[25] R. Li and C. F. Li, "Adaboost face detection based on improved covariance feature," *Journal of Computers*, vol. 9, no. 5, pp. 1077–1082, 2014.