Expert Systems with Applications 41 (2014) 1139–1152

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Detecting driver drowsiness using feature-level fusion and user-specific classification

Jaeik Jo [a], Sung Joo Lee [a], Kang Ryoung Park [b], Ig-Jae Kim [c], Jaihie Kim [a,d,*]

[a] School of Electrical and Electronic Engineering, Yonsei University, Seoul 120-749, Republic of Korea
[b] Division of Electronics and Electrical Engineering, Dongguk University, Seoul 100-715, Republic of Korea
[c] Imaging Media Research Center, Korea Institute of Science and Technology, Seoul 136-130, Republic of Korea
[d] Sunway University, 46150 Petaling Jaya, Selangor, Malaysia

ARTICLE INFO

Keywords:
Drowsiness detection system
Blink detection
Eye state classification
Feature-level fusion
User-specific classification

ABSTRACT

Accurate classification of eye state is a prerequisite for preventing automobile accidents due to driver drowsiness. Previous methods of classification, based on features extracted for a single eye, are vulnerable to eye localization errors and visual obstructions, and most use a fixed threshold for classification, irrespective of variations in the driver's eye shape and texture. To address these deficiencies, we propose a new method for eye state classification that combines three innovations: (1) extraction and fusion of features from both eyes, (2) initialization of driver-specific thresholds to account for differences in eye shape and texture, and (3) modeling of driver-specific blinking patterns for normal (non-drowsy) driving. Experimental results show that the proposed method achieves significant improvements in detection accuracy.

1. Introduction

According to the American National Highway Traffic Safety Administration (NHTSA) Liu & Subramanian, 2009, approximately 100,000 accidents per year occur because of driver drowsiness. In response to this mounting problem, methods for detecting driver drowsiness have been intensively studied in the automotive field (Adachi et al., 2008; Bergasa, Nuevo, Sotelo, Barea, & Lopez, 2006; Bhowmick & Chidanand Kumar, 2009; Damousis & Tzovaras, 2008; Dong, Hu, Uchimura, & Murayama, 2011; Eriksson & Papanikotopoulos, 1997; Ersal, Fuller, Tsimhoni, Stein, & Fathy, 2010; Flores & Armingol, 2010; Ince & Yang, 2009; Jimenez-Pinto & Torres-Torriti, 2009; Jo, Lee, Jung, Park, & Kim, 2011; Jo et al., 2010; Kawato & Ohya, 2000; Kircher, Uddman, & Sandin, 2002; Kurylyak, Lamonaca, & Mirabelli, 2012; Li, 2008; Liu, Hosking, & Lenn, 2009; Minkov, Zafeiriou, & Pantic, 2012; Murphy-Chutorian, Doshi, & Trivedi, 2007; Noguchi, Shimada, Ohsuga, Kamakura, & Inoue, 2009; Orazio, Leo, Guaragnella, & Distante, 2007; Panning, Al-Hamadi, & Michaelis, 2011; Parmar, 2002; Patel, Lal, Kavanagh, & Rossiter, 2011; Rongben, Lie, Bingliang, & Lisheng, 2004; Saradadevi & Bajaj, 2008; Shuyan & Gangtie, 2009; Sukno, Pavani, Butakoffand, & Frangi, 2009; Torkkola, Massey, & Wood, 2004; Tran, Craig, Wijesuriya, & Nguyen, 2010; Tsuchida, Bhuiyan, & Oguri, 2010; Uliyar & Ukil, 2012; Vural et al., 2007; Wang, Ding, Fang, & Liu, 2009; Wu & Chen, 2008; Wu & Trivedi, 2007; Yang et al., 2009; Yunq, Meiling, Xiaobing, Xiuxia, & Jiangfan, 2009). As with the detection of driver inattention (Dong et al., 2011; Jo et al., 2011), the methods used for drowsiness detection can be divided into three categories: those based on driving behavior (Ersal et al., 2010; Kircher et al., 2002; Liu et al., 2009; Torkkola et al., 2004; Yang et al., 2009), those based on physiological features (Damousis & Tzovaras, 2008; Patel et al., 2011; Shuyan & Gangtie, 2009; Tran et al., 2010), and those based on visual features (Adachi et al., 2008; Bergasa et al., 2006; Bhowmick & Chidanand Kumar, 2009; Eriksson & Papanikotopoulos, 1997; Flores & Armingol, 2010; Ince & Yang, 2009; Jimenez-Pinto & Torres-Torriti, 2009; Jo et al., 2010, 2011; Kawato & Ohya, 2000; Kurylyak et al., 2012; Li, 2008; Minkov et al., 2012; Murphy-Chutorian et al., 2007; Noguchi et al., 2009; Orazio et al., 2007; Panning et al., 2011; Parmar, 2002; Rongben et al., 2004; Saradadevi & Bajaj, 2008; Sukno et al., 2009; Tsuchida et al., 2010; Uliyar & Ukil, 2012; Vural

Abbreviations: ASM, active shape model; CCA, canonical correlation analysis; ECD, eye closure duration; EER, equal error rate; ESD-Value, eyelid state detection value; FEC, frequency of eye closure; GLCM, grey-level co-occurrence matrix; GRBF, Gaussian radial basis function; LBP, local binary pattern; LDA, linear discriminant analysis; NHTSA, national highway traffic safety administration; PCA, principal component analysis; PERCLOS, percentage of eye closure; PHOGs, pyramid histogram of oriented gradients; ROI, region of interest; SVM, support vector machine.

* Corresponding author. Address: Sunway University, 46150 Petaling Jaya, Selangor, Malaysia. Tel.: +82 2 2123 2869; fax: +82 2 312 4584.

E-mail addresses: jaeik@yonsei.ac.kr (J. Jo), sungjoo@yonsei.ac.kr (S.J. Lee), parkgr@dongguk.edu (K.R. Park), kij@imrc.kist.re.kr (I.-J. Kim), jhkim@yonsei.ac.kr (J. Kim).

0957-4174/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2013.07.108

et al., 2007; Wang et al., 2009; Wu & Chen, 2008; Wu & Trivedi, 2007; Yunq et al., 2009). Methods based on driving behavior detect drowsiness by monitoring vehicle speed, lane observation, steering, acceleration, braking, and gear changes. The main drawback of these methods is that their accuracy depends on the individual characteristics of the vehicle and its driver. In contrast, methods based on physiological features detect drowsiness by measuring heart rate and brain activity. Although these methods show good detection accuracy, they also depend on peripheral measuring equipment that must be attached to the driver's body. Finally, methods based on visual features detect drowsiness using information obtained from a camera, and thus neither depend upon vehicle or driver characteristics nor require intrusive measuring equipment. As such, visual feature-based methods have emerged as the preferred avenue for research.

A great number of "visual feature-based methods" for drowsiness detection have been proposed and studied. Among the visual features used by these methods are eye state information (Adachi et al., 2008; Bergasa et al., 2006; Bhowmick & Chidanand Kumar, 2009; Eriksson & Papanikotopoulos, 1997; Flores & Armingol, 2010; Ince & Yang, 2009; Jo et al., 2010, 2011; Kurylyak et al., 2012; Li, 2008; Minkov et al., 2012; Noguchi et al., 2009; Orazio et al., 2007; Panning et al., 2011; Parmar, 2002; Sukno et al., 2009; Tsuchida et al., 2010; Uliyar & Ukil, 2012; Wang et al., 2009; Wu & Trivedi, 2007; Yunq et al., 2009), head movement (Kawato & Ohya, 2000; Murphy-Chutorian et al., 2007), yawning (Rongben et al., 2004; Saradadevi & Bajaj, 2008) and facial expression (Jimenez-Pinto & Torres-Torriti, 2009; Vural et al., 2007). Methods using eye state to measure driver drowsiness have generally done so by calculating values such as the percentage of eye closure (PERCLOS) Wierwille, Ellsworth, Wreggit, Fairbanks, & Kim, 1994, eye closure duration (ECD), and the frequency of eye closure (FEC) Orazio et al., 2007. Methods using head movement measure drowsiness by estimating head posture. Methods based on yawning locate the driver's mouth and then train the system with images of normal and yawning mouths. Methods that use facial expressions to detect drowsiness generally combine several facial cues, such as yawning, blinking, and eyebrow rising. It should be noted that some of the above features suffer deficiencies in timing: yawning generally occurs well before drowsiness sets in, and head nodding generally occurs after the driver falls asleep. Thus, methods based on these features cannot detect the onset of drowsiness precisely, and are therefore unsuitable for reliable detection systems. On the other hand, eye status information is well-suited for such systems, since the closing of eyes and the appearance of unusual patterns of blinking have been shown to directly indicate the onset of drowsiness. Indeed, methods based on eye status information have already shown superior accuracy in detecting drowsiness (Vural et al., 2007).

The core technology used by these methods is an algorithm for classifying eye state (i.e., as open or closed). Prior work on such algorithms can be categorized into three methods: texture-based methods (Bergasa et al., 2006; Eriksson & Papanikotopoulos, 1997; Flores & Armingol, 2010; Jo et al., 2010, 2011; Kurylyak et al., 2012; Li, 2008; Minkov et al., 2012; Orazio et al., 2007; Panning et al., 2011; Parmar, 2002; Uliyar & Ukil, 2012; Wu & Trivedi, 2007), shape-based methods (Adachi et al., 2008; Ince & Yang, 2009; Noguchi et al., 2009; Sukno et al., 2009; Tsuchida et al., 2010; Wang et al., 2009), and combined texture–shape methods (Bhowmick & Chidanand Kumar, 2009; Yunq et al., 2009). Texture-based methods extract texture features for eye state classification using various feature extraction methods. Minkov et al. (2012) described a blinking detection method using the following features: raw-image intensities, the magnitude of the responses of Gabor filters, the pyramid histogram of oriented gradients (PHOGs), and optical flow. Classification was achieved through a support vector machine (SVM) using Gaussian radial basis function (GRBF) kernels. Jo et al. (2011) introduced an eye state classification method that uses the combination of appearance and statistical features. The appearance features are extracted using principal component analysis (PCA) and linear discriminant analysis (LDA), and the statistical features are acquired using the sparseness and kurtosis of the histogram from the eye-edge image. Uliyar and Ukil (2012) proposed a method based on canonical correlation analysis (CCA) coupled with local binary pattern (LBP) histogram features calculated from the input eye image. Their experiments show that the coupling of these features results in 10–12% improvement in eye state classification accuracy, compared to methods using normalized intensity-based features. Panning et al. (2011) introduced an algorithm for eye state classification using the eyelid state detection value (ESD-Value) calculated by comparing the pixel values in the region of interest (ROI) with an experientially selected threshold value. In addition to the above, features such as Tensor PCA (Wu & Trivedi, 2007), Gabor response waves (Flores & Armingol, 2010; Li, 2008), frame differencing (Bergasa et al., 2006; Kurylyak et al., 2012), and histogram (Eriksson & Papanikotopoulos, 1997; Parmar, 2002) and edge (Jo et al., 2010; Orazio et al., 2007) of the local eye image, have all received attention in the growing body of research on texture-based methods of eye state classification.

As for shape-based methods, eye state has generally been classified using a measurement of the distance between the upper and lower eyelids. Wang et al. (2009), Sukno et al. (2009), and Noguchi et al. (2009) extracted eye contours using an active shape model (ASM) and measured eyelid distance using landmarks in both eyes. Adachi et al. (2008) carried out a similar measurement, but used two search windows to detect the eyelids.

In the combined methods of classification, both texture and shape features are extracted from the input eye image to increase robustness. Bhowmick and Chidanand Kumar (2009) proposed a method in which texture features such as histogram energy and contrast on grey-level co-occurrence matrix (GLCM) are fused with several different shape features, including Hu's seventh moment, compactness, and top-hat and bottom-hat area ratio and eccentricity. The combined features are then used to train the nonlinear SVM in eye state classification.

Unfortunately, all of the above classification methods have two distinct problems. First, they suffer direct and unavoidable performance degradation when eye localization errors occur, or when the eye region is obstructed (e.g., by eyeglasses). Second, these methods generally apply the same classification thresholds to all drivers, regardless of individual differences in eye shape, scale, and blinking frequency.

To address the first of these problems, we propose a method of eye state classification that uses feature-level fusion of both eyes, instead of only one. This approach eliminates a number of errors attributable to single-eye obstructions. To address problems caused by the application of fixed thresholds, we examine an initial period of driving to calculate a baseline probability that the driver will have open eyes at any given moment, and then classify eye state as open or closed using a maximum a posteriori (MAP) classifier and a user-specific threshold. Further, during the initial period of driving, we obtain the driver's blinking pattern, based on which normal (non-drowsy) driving behavior is learned using a two-dimensional (2D) Gaussian model.

The remainder of this paper is organized as follows: in Section 2, we describe the proposed method, comprising eye detection and tracking, eye state classification, and drowsiness detection; in Section 3, we present experimental results with an image database collected from a vehicle under various conditions; and in Section 4, we provide some conclusions.

2. Proposed method

Fig. 1 illustrates the general processes of our drowsiness detection system: face and eye detection and tracking, eye state classification, and drowsiness decision.

2.1. Eye detection & tracking

In our system, the face and eye detection and tracking methods introduced by Jo et al. (2011) and Lee, Jo, Jung, Park, & Kim, 2011 was applied. The face is detected in the first frame using adaptive boosting (AdaBoost) (see Fig. 2), after which an adaptive template matching method is applied to track the face efficiently and with good tolerance of facial variations. Through this process, the face region is accurately localized and unnecessary background imaging is removed.

Next, the eye region is located within the detected face region using AdaBoost and blob detection (see Fig. 2). Since the eye has fewer features than the face, AdaBoost is sometimes unable to detect the eye correctly, in which case blob detection provides additional support. Adaptive template matching is again used to track the detected eye quickly and accurately. As in all systems of eye detection, an eye validation procedure follows every frame to see that the detected or tracked object is truly an eye image. In the validation process, an SVM classifier is trained to classify the detected eye image as eye or non-eye using the features extracted by PCA and LDA. Further details on face and eye detection procedures are described in Jo et al. (2011) and Lee et al., 2011.

2.2. Eye state classification

The eye state classification process (i.e., classifying the detected eye as either open or closed) is split into four sub-processes, as shown in Fig. 1: feature extraction, feature-level fusion, SVM classification, and user-specific classification.

2.2.1. Feature extraction

The four methods used in Jo et al. (2010, 2011), PCA, LDA, sparseness, and kurtosis, are for extracting the features needed for eye state classification. PCA is a well-known unsupervised algorithm for feature extraction. It is a linear mapping that uses the eigenvectors with the largest eigenvalues (Turk & Pentland, 1991). The PCA algorithm can be used to find a linear transformation orthonormal matrix that maps the original n-dimensional feature space into m-dimensional feature subspaces ($m \ll n$). LDA is a supervised learning method, which utilizes the category information associated with each sample (Martinez & Kak, 2001). The goal of LDA is to maximize the between-class scatter while minimizing the within-class scatter. In the extraction process, PCA and LDA features are obtained after projecting an eye image onto PCA and LDA axes. The other two features, kurtosis and sparseness, are extracted from histogram analysis. As shown in Fig. 3, a horizontal edge image is obtained using a horizontal Sobel convolution mask. By applying a p-tile threshold method, the image is transformed into a binary image, which is then projected onto a horizontal axis to obtain the vertical projection histogram. Fig. 3 shows how an eye can be classified as open or closed using the sparseness and kurtosis of the projected histogram. The open eye has a larger sparseness and kurtosis than the closed eye, since the edge pixels in the horizontal edge image of the open eye are more concentrated at the center.

Table 1 shows the equations and descriptions of these feature extraction methods. Overall, 14 features (12 coefficients from PCA and LDA, 1 from sparseness, and 1 from kurtosis) are obtained for each eye, bringing the total number features for both eyes to 28. How these features are combined is the subject of the next section.

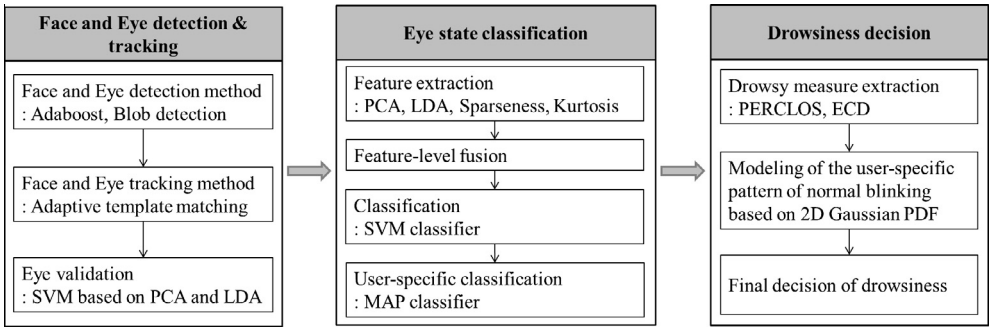

Fig. 1. Flow chart of general processes in the proposed system.

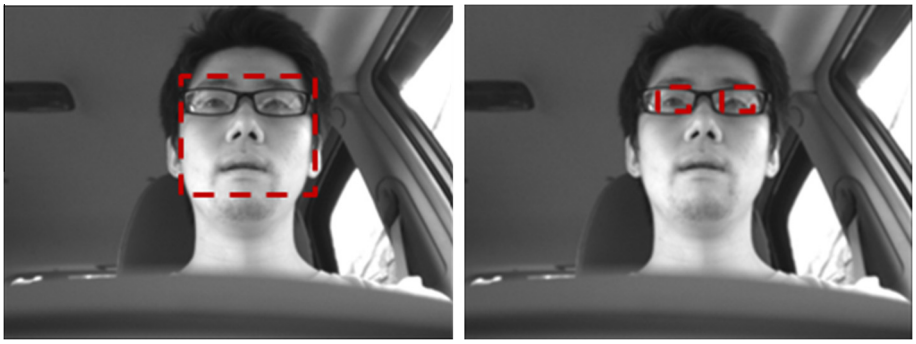

Fig. 2. Detected face and eye regions.

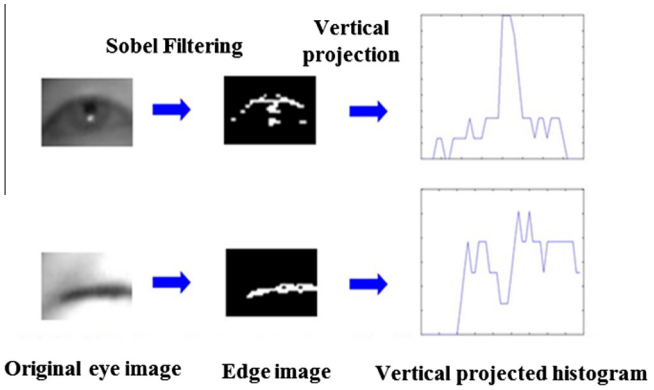

Fig. 3. Histograms of projected edges from open eye and closed eye.

Table 1
The equations for feature extraction methods.

Feature extraction method	Equations	Descriptions
PCA	$x_i = (x_{1i}, x_{2i}, \ldots, x_{ni})^T \in R^n$	x_i is a column vector of pixels of an eye image
	$X = \{x_1, x_2, \ldots, x_l\}$	X denotes the training sample set that consists of l eye images
	$y_i = W_{PCA}^T \cdot x_i \ (i = 1, 2, \ldots, l)$	y_i is the reduced feature vector. W_{PCA} are the m eigenvectors
LDA	$S_B = \sum_{j=1}^{c} (\mu_i - \mu)(\mu_i - \mu)^T$	S_B is the between-class scatter matrix, μ is the mean image of all classes, and c is the number of classes
	$S_W = \sum_{j=1}^{c} \sum_{i=1}^{N_j} (x_i^j - \mu_j)(x_i^j - \mu_j)^T$	S_W is the within-class scatter matrix, x_i^j is the ith sample of class j, and N_j is the number of samples of class j
PCA + LDA	$F_i = W_{LDA}^T W_{PCA}^T x_i \ (i = 1, 2, \ldots, l)$	F are the final reduced feature vectors
Kurtosis	$F = \frac{1}{\sigma^4} \sum_{n=0}^{N-1} (n - \mu)^4$	μ is the mean of n and σ is the standard deviation of n
Sparseness	$F(x) = \frac{\sqrt{d} - (\sum \lvert x_i \rvert) / \sqrt{\sum \lvert x_i \rvert^2}}{\sqrt{d} - 1}$	d represents the dimensionality of the vector x whose jth component is x_j

2.2.2. Feature-level fusion

As shown in Fig. 4, the accuracy of eye state classification can be reduced by eye localization failures or obstruction of the eye region. Our method addresses these problems by fusing features from both eyes, so that even when the features of one eye cannot be properly extracted, the features of the other eye can supply sufficient information to make a good classification.

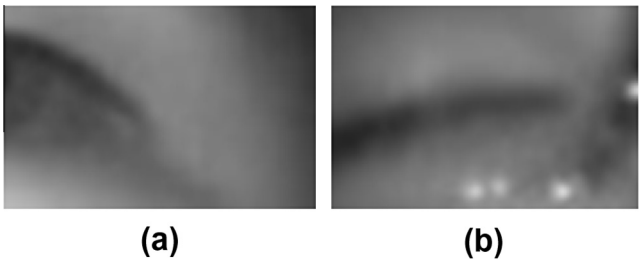

Fig. 4. Sources of error in eye state classification: (a) eye localization failure, (b) obstruction (by eyeglasses).

In general, fusion at the feature level involves the integration of feature sets obtained from multiple modalities. In score-level and decision-level fusion, some of the information is lost as the multi-dimensional feature set of an object is simplified into a one-dimensional match score and a final decision. That is, the feature set contains richer information about the open and closed eye data than the match score or the final decision (Ross & Govindarajan, 2004). Thus, we can expect that fusion at the feature level will provide better classification results than fusion at the score level or the decision level. The details of our feature-level fusion method are described below.

Let $L = \{l_1, l_2, \ldots l_n\}$ and $R = \{r_1, r_2, \ldots r_n\}$ represent the feature vector of the left and right eyes of a driver, respectively. The normalized vectors L' and R' are computed by applying a min–max normalization scheme to the individual feature values in order to make the feature values compatible. The fused feature vector $F = \{f_1, f_2, \ldots f_{2n}\}$ can be obtained by concatenating the normalized feature vectors of L' and R'. The feature-level fusion procedure is shown in Fig. 5. The fused feature vector F is then used as the input for SVM, as explained in the next section.

2.2.3. SVM decision score calculation

In SVM training, support vectors are selected and the optimal decision hyper-plane that discriminates two or more classes is determined by maximizing the distances between the support vectors. In general, the SVM can be defined as (Vapnik, 1999):

$$f(x) = \mathrm{sgn}\left(\sum_{i=1}^{k} \alpha_i y_i K(x, s_i) + b\right) \quad (1)$$

where $y_i = \{-1, 1\}$ represents the class label of training data and $K(x, s_i)$ is a SVM kernel function. The Lagrange multiplier α_i can be found by solving a quadratic programming problem with linear constraints, with b as the bias term. s_i represents the support vectors and k represents the number of support vector points. The expression $\sum_{i=1}^{k} \alpha_i y_i K(x, s_i) + b$ is the SVM decision score. Generally, the performance of SVM changes depending on the type of kernels used. The following four kernel functions are most commonly used for the SVM classifier:

$$\text{Linear kernel}: \quad K(x, s) = x \cdot s \quad (2)$$

$$\text{Polynomial kernel}: \quad K(x, s) = \gamma(x \cdot s + r)^d \quad (3)$$

$$\text{RBF kernel}: \quad K(x, s) = \exp\left(-\gamma \|x - s\|^2\right), \gamma > 0 \quad (4)$$

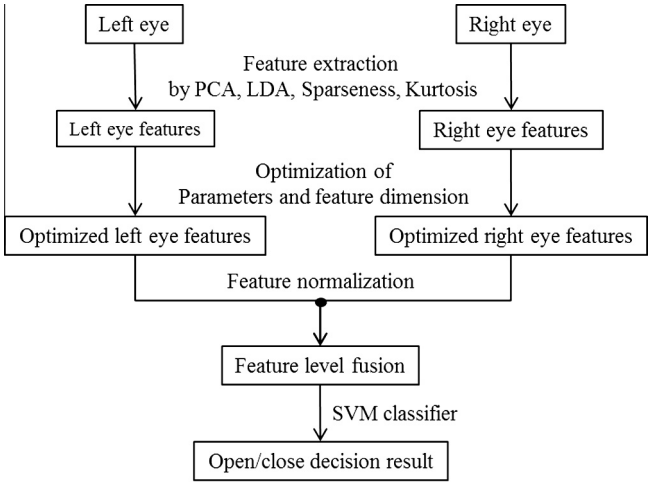

Fig. 5. Feature-level fusion procedure.

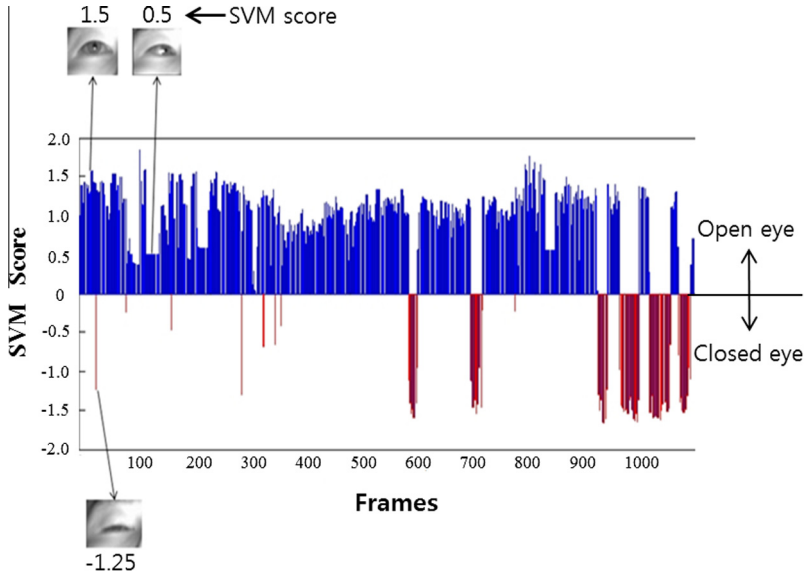

Fig. 6. Distribution of SVM score.

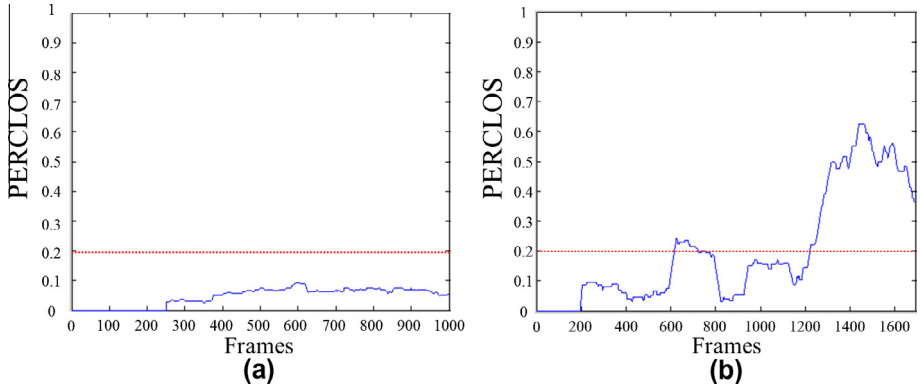

Fig. 7. PERCLOS plotted for (a) normal driving and (b) drowsy driving.

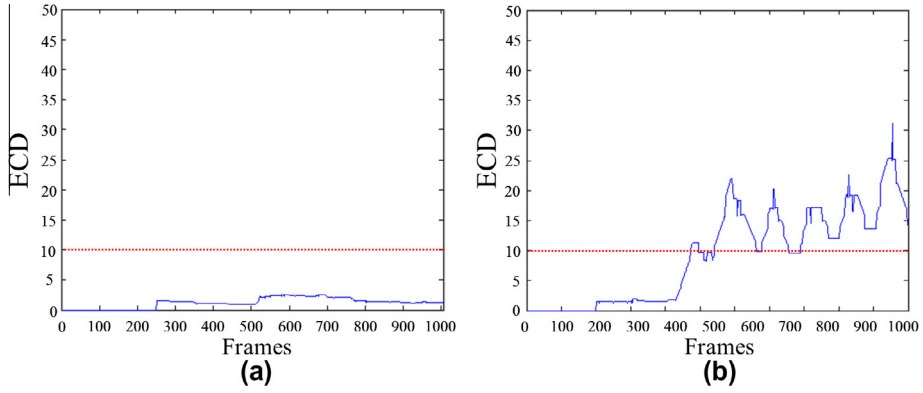

Fig. 8. ECD plotted for (a) normal driving and (b) drowsy driving.

$$\text{Sigmoid kernel}: \quad K(x,s) = \tanh(\gamma x \cdot s + r) \tag{5}$$

where γ, r, and d are kernel parameters. These parameters need to be optimized to maximize the classification performance.

The fused feature vector F (from Section 2.2.2) is applied to the SVM classifier and a one-dimensional SVM decision score is obtained. The SVM classifier is previously trained using features of some number of open and closed eyes, and it gives a score of "1" to an open eye and "0" to a closed eye. As shown in Fig. 6, for an input image of an open eye the SVM decision score approximates to "1," while for an image of a closed eye, it approximates to "0." Accordingly, if the score is greater than a specific threshold,

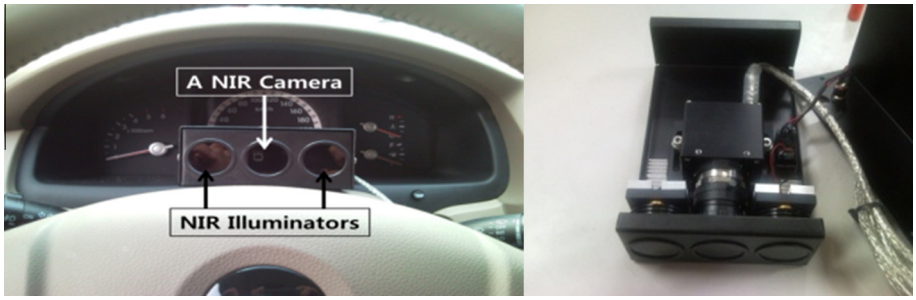

Fig. 9. Image acquisition device.

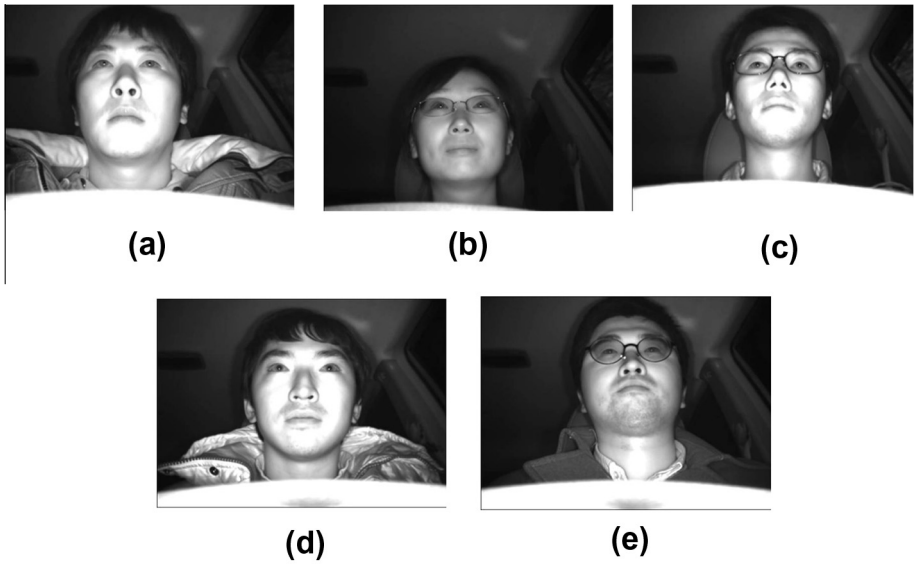

Fig. 10. Example images taken (a) during daytime without glasses, (b) during daytime with glasses, (c) during daytime with sunglasses, (d) during nighttime without glasses, (e) during nighttime with glasses.

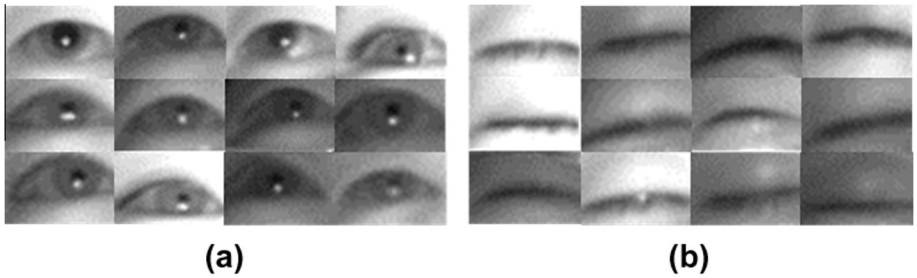

Fig. 11. Some example images from the dataset: (a) open-eye images, (b) closed-eye images.

the image is classified as an open eye, and in all other cases it is classified as closed. Given the decision score provided by SVM, a user-specific classification (see Section 2.2.5) will then be made.

2.2.4. Score-level fusion

The superiority of our feature-level fusion is confirmed by comparison to the score-level fusion described in this section. For score-level fusion, the 14 features (described in Section 2.2.1) for the left eye are input into one SVM and the 14 features of the right eye are input into another SVM. This yields two decision scores, for which the following seven fusion methods can be compared.

The first fusion method calculates the mean of the two SVM decision scores using the following:

$$S_F = \frac{S_L + S_R}{2} \tag{6}$$

where S_L and S_R represent the SVM decision score of the left and right eyes respectively, and S_F indicates the SVM decision score fused by the equation.

The second fusion method is by weighted mean. A different weight (w) is applied to each eye to calculate the fused score, as given in the following:

Table 2
Profile of training database.

Day	Number of subjects	12 persons			
		9 males		3 females	
		No glasses:1	Glasses: 8	No glasses:1	Glasses: 2
	Number of images	44,197 images from 216 image sequences (12 persons × 18 sequences)			
Night	Number of subjects	11 persons			
		8 males		3 females	
		No glasses:2	Glasses: 6	No glasses:1	Glasses: 2
	Number of images	41,155 images from 198 image sequences (11 persons × 18 sequences)			

Table 3
Profile of testing database.

Day	Number of subjects	9 persons				
		6 males			3 females	
		No glasses:6	Glasses:3	Sunglasses:3	No glasses:3	Glasses:2	Sunglasses:1
	Number of images used for eye state classification evaluation	68,756 images from 36 sequences (Left eye: 34,486 images, Right eye: 34,270 images)				
	Number of sequences used to measure drowsiness level	189 video sequences driving normally 72 video sequences driving drowsily (Total number of images: about 157,100 images from 261 image sequences.)				
Night	Number of subjects	10 persons				
		7 males			3 females	
		No glasses: 4	With glasses:3		No glasses: 1	With glasses:2
	Number of images used for eye state classification evaluation	64,579 images from 20 sequences (Left eye: 32,419 images, Right eye: 32,160 images)				
	Number of sequences used to measure drowsiness level	120 video sequences driving normally 40 video sequences driving drowsily (Total number of images: about 76,400 images from 160 image sequences.)				

Table 4
Numbers of open and closed eye images in testing database taken under different conditions.

Conditions		Left open eye	Left closed eye	Right open eye	Right closed eye
Time	Glass type				
Day	Not wearing glasses	18,670	2302	18,305	2259
	Eyeglasses	7700	1025	7714	1200
	Sunglasses	4313	476	4337	455
Night	Not wearing glasses	15,948	2663	15,811	2454
	Eyeglasses	12,375	1433	12,626	1269
Total		59,006	7899	58,793	7637

$$S_F = \frac{wS_L + (2-w)S_R}{2} \quad (7)$$

The third fusion method selects the minimum of the two (left and right) SVM decision scores and the fourth fusion method selects the maximum of the two scores. The fifth multiplies the two scores and adds a constant (2.5) to prevent the multiplied score from being negative, as given in the following:

$$S_F = (S_L + 2.5) \times (S_R + 2.5) \quad (8)$$

The sixth fusion method uses a weighted product, as given in the following:

$$S_F = (S_L + 2.5)^w \times (S_R + 2.5)^{1-w} \quad (9)$$

Finally, the seventh fusion method inputs the left and right decision scores into yet another SVM classifier to produce the fused score. Having analyzed the performance of SVM using the various kernels listed in Eqs. (2)–(5), we selected the RBF kernel (Eq. (4)) as optimal.

2.2.5. User-specific classification method

Using the fused decision scores described above as input, eye state classification can be performed. Previous methods (Adachi et al., 2008; Bergasa et al., 2006; Bhowmick & Chidanand Kumar, 2009; Eriksson & Papanikotopoulos, 1997; Flores & Armingol, 2010; Ince & Yang, 2009; Jimenez-Pinto & Torres-Torriti, 2009; Jo et al., 2010, 2011; Kurylyak et al., 2012; Li, 2008; Minkov et al., 2012; Noguchi et al., 2009; Orazio et al., 2007; Panning et al., 2011; Parmar, 2002; Sukno et al., 2009; Tsuchida et al., 2010; Uliyar & Ukil, 2012; Vural et al., 2007; Wang et al., 2009; Wu & Trivedi, 2007; Yunq et al., 2009) applied a fixed threshold (T_F) to all drivers. For example, if the distance between upper and lower eyelids is larger than some preset value (typically the result of training the system on a database of different subjects' eye images) the eye is classified as open. However, since each driver has different eye shapes and textures, the threshold value ought to vary from driver to driver. To achieve this variation, we propose a classification method that finds a user-specific threshold (T_U) for each driver.

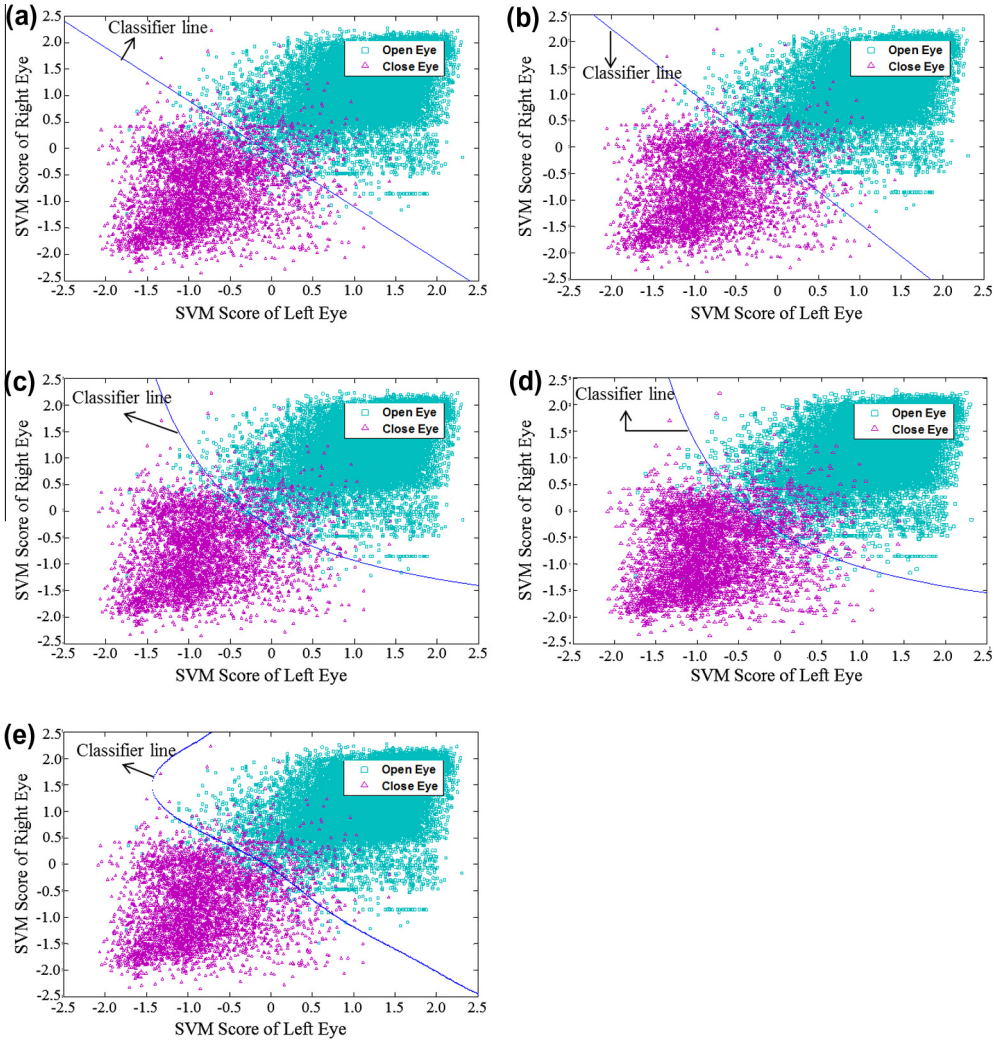

Fig. 12. Classification performance of score-level fusion methods: (a) mean, (b) weighted mean, (c) product, (d) weighted product, (e) SVM with RBF kernel.

Table 5
Comparisons of EERs according to various methods.

Fusion rule			EER (%)
Score level fusion	Mean		1.94
	Weighted mean		1.87
	Minimum		2.80
	Maximum		2.75
	Product		2.32
	Weighted product		2.01
	SVM	RBF	1.81
		Linear	1.83
		Sigmoid	1.83
		Polynomial	1.92
Feature level fusion with RBF kernel based SVM (proposed method)			1.40
Without fusion (Jo et al., 2011)			Left eye:3.84 Right eye: 3.90

An MAP classifier is employed to find the user-specific threshold, as given in the following:

$$\text{Decide} \quad w_1 \ \text{if} \ P(w_1|S_F) > P(w_2|S_F) \quad \text{otherwise decide } w_2 \tag{10}$$

where $P(w_1|S_F)$ denotes the probability that the eye is open given a score (S_F) and $P(w_2|S_F)$ is the probability that the eye is closed given a score (S_F). Here, S_F is the fused output score of SVM explained in Sections 2.2.3 and 2.2.4. $P(w_1|S_F)$ can then be expressed as a multiplication of likelihood $P(S_F|w_1)$ and prior possibility $P(w_1)$, as given in the following:

$$P(w_1|S_F) = \frac{P(S_F|w_1)P(w_1)}{P(x)} \tag{11}$$

The value of $P(x)$ does not affect the result of eye state classification as it is only a scaling factor that makes $P(w_1|S_F) + P(w_2|S_F)$ equal to 1. $P(S_F|w_1)$ is a distribution of S_F given that the eye is open. From the eye images captured in an initial driving period, the mean and standard deviation of S_F for open eyes are calculated. Using these, $P(S_F|w_1)$ is then calculated on the assumption that $P(S_F|w_1)$ is Gaussian. Also, $P(w_1)$ is found from the ratio of open eye frames over the entire set of initial driving images. However, in calculating $P(S_F|w_1)$ and $P(w_1)$ from the initial driving images, no ground truth data on open and closed eyes exists. To obtain such data in the initial stage of driving, eye state classification is performed based on a fixed threshold, obtained through experimentation.

Because the number of closed eye images in the initial set is very low compared to that of open eye images, it is difficult to correctly calculate $P(w_2|S_F)$ based on the assumed Gaussian distribution. Therefore, instead of calculating $P(w_2|S_F)$, the eye state is classified as open when $P(w_1|S_F)$ is larger than the user-specific threshold, T_U, and vice versa. T_U is experimentally set to the

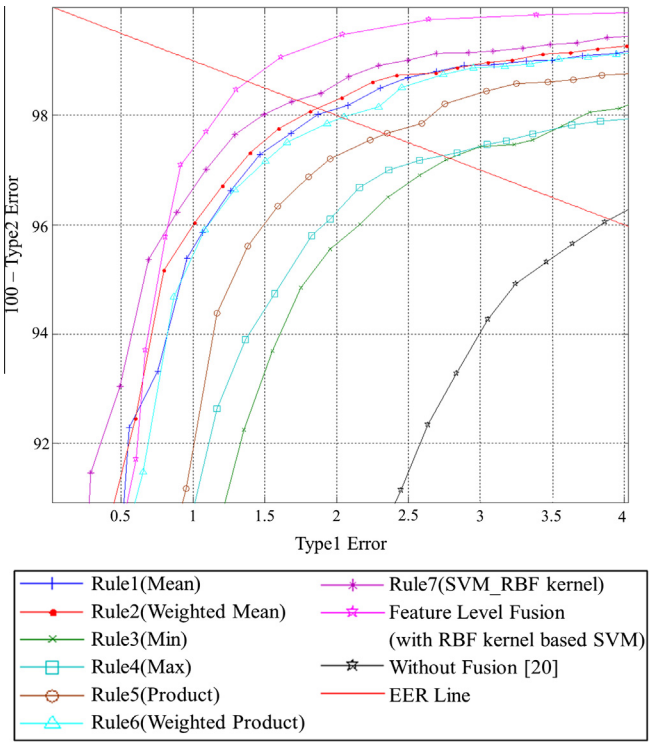

Fig. 13. Receiver operating characteristic (ROC) curves for various fusion methods.

Table 6
EERs with fixed and user-specific threshold.

	EER (%)
A fixed threshold	1.40
An user-specific threshold	0.93

bottom 0.05% of a posteriori probability distribution, as given in the following:

$$\text{Decide} \quad w_1 \ \text{if} \ P(w_1|S_F) > T_U; \\ \text{otherwise decide } w_2 \tag{12}$$

2.3. Drowsiness detection

After having classified eyes as open or closed, the drowsiness of a driver is determined by 2D Gaussian mixture model which takes PERCLOS (Wierwille et al., 1994) and ECD (Orazio et al., 2007) of the driver's normal state driving images as two inputs.

PERCLOS is a parameter that is widely used to monitor the drowsiness of a driver. It is defined as the proportion of frames in which the driver's eyes are closed over a certain period, as given in the following (Jo et al., 2011):

$$\text{PERCLOS}[k] = \frac{\sum_{i=k-n+1}^{k} \text{Blink}[i]}{n} \times 100(\%) \tag{13}$$

where PERCLOS[k] is the PERCLOS value in the kth frame and n is a window size and the total number of frames within the period measuring PERCLOS. Blink[i] is a single binary value that represents the status of the eye at ith frame. Blink[i] is "0" when the eye is open and "1" when the eye is closed. The blink pattern information during the previous 200 frames, which is the same as the window size, is required to calculate PERCLOS at an arbitrary time. The higher the PERCLOS is, the greater the level of driver drowsiness. Fig. 7 shows the result of PERCLOS measurement for image sequences of normal (non-drowsy) and drowsy driving. In the case of the drowsy driving, the driver was initially in normal driving state then enters the slightly drowsy driving state. Finally the driver is in a drowsy driving state after 600 frames. In our experiments, when PERCLOS exceeds 0.2, it was quite probable that driver is in a drowsy state.

ECD is defined as the mean duration of clusters over a certain period, where a cluster is a set of continuous frames in which the eyes are closed, as given by the following:

$$\text{ECD}[k] = \frac{\sum_{i=1}^{P} \text{Duration}[C[k-n]+i]}{P} \tag{14}$$

where Duration[i] is the number of continuous closed eye frames in ith cluster, n is the total number of frames within the period measuring ECD, P is the number of clusters in the most recent n frames, and C[k] is the total number of clusters in 0 to k frames. An example calculation of ECD is given by the following:

$$000001110000111110011110 0$$
$$(1\text{for closed eye},0 \text{ for open eye})$$
$$\text{ECD} = (3+5+4)/3 = 4$$

Fig. 8 shows ECD for image sequences of normal and drowsy driving. Note that ECD follows a pattern similar to that of PERCLOS, and is a similarly effective indicator of drowsy driving. In our experiments, when ECD exceeds 10, it was quite probable that driver is a drowsy state.

PERCLOS is not greatly affected by eye state classification errors over a given period. That is, a few frames with eye state classification errors over the course of several hundreds of frames does not significantly influence the proportion of frames in which the driver's eyes are closed for the period covered. However, given the wide variation in blinking frequency, the measured PERCLOS for a driver who blinks often can be quite high, resulting in a false detection of drowsy driving. On the other hand, ECD is not greatly affected by blinking frequency, but it can be greatly affected by eye state classification errors. When classification errors occur in the middle of closed eye sequences, the measured ECD can be significantly reduced, resulting in a failure to identify drowsy driving. Only by combining these two complementary measures, we can improve the accuracy of drowsiness detection.

The combination of the two measures is accomplished by using a Gaussian mixture model for the "normal driving" pattern of a given driver. The Gaussian mixture model is a probabilistic model that assumes that all the data points are generated from a finite number of Gaussian distributions with unknown parameters, and this model is widely used to model complex multidimensional distributions. The 2D Gaussian probability density function can be represented as follows:

$$G\left(\mu, \sum\right)(x) = \frac{1}{\sqrt{2\pi \cdot \det(\sum)}} \cdot \exp^{-1/2(x-\mu)^T \sum^{-1} (x-\mu)} \tag{15}$$

where $x = [\text{PERCLOSECD}]^T$, μ is mean vector of PERCLOS and ECD over the initial normal driving period, and $\sum$ is covariance. Rather than applying the same Gaussian mixture model for all users, user-specific Gaussian mixture model constructed using the PERCLOSE and ECD during the initial driving period. After the construction of the normal driving model, drowsy driving can be detected by calculating the probability of drowsy measurements in the test image sequences belonging to the constructed model: the lower the calculated probability, the higher the probability that the driver is drowsy. A threshold of 2D Gaussian pdf that determines the drowsiness of the driver was predetermined experimentally and set to 0.005. System will generate warning if the calculated probability

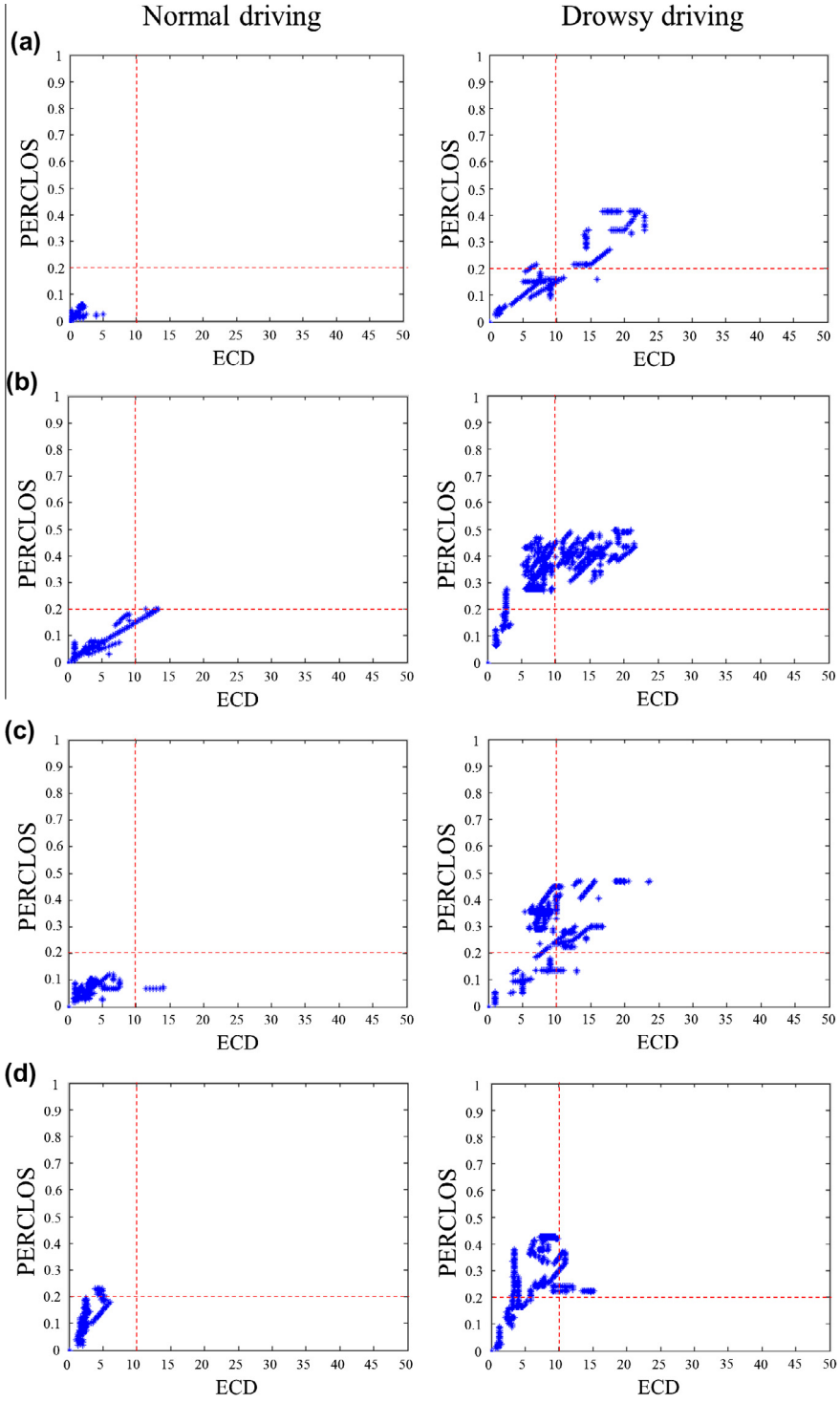

Fig. 14. Distribution of PERCLOS and ECD in normal and drowsy driving sequences for four drivers: (a) driver 1, (b) driver 2, (c) driver 3, and (d) driver 4.

stays below the predetermined threshold (0.005) for over 10 frames.

3. Experiments

3.1. Database

In order to evaluate the proposed method, we have collected images of drivers in various circumstances: daytime, nighttime, with and without eyeglasses and sunglasses. In order to prevent variations in environmental lighting, a near-infrared (NIR) filter is attached to the camera lens and two 850 nm NIR illuminators are installed on either side (Jo et al., 2011). The assembled device is then installed in the test vehicle as shown in Fig. 9. In Fig. 10, we see samples of images acquired with this device (see Fig. 11).

Tables 2–4 describe the composition and specification of the database in different circumstances and conditions. Numbers of images of left eye and right eye are different for two reasons; one of the eyes is not visible in the acquired image due to head rotation and those images failed in ROI detection were not used

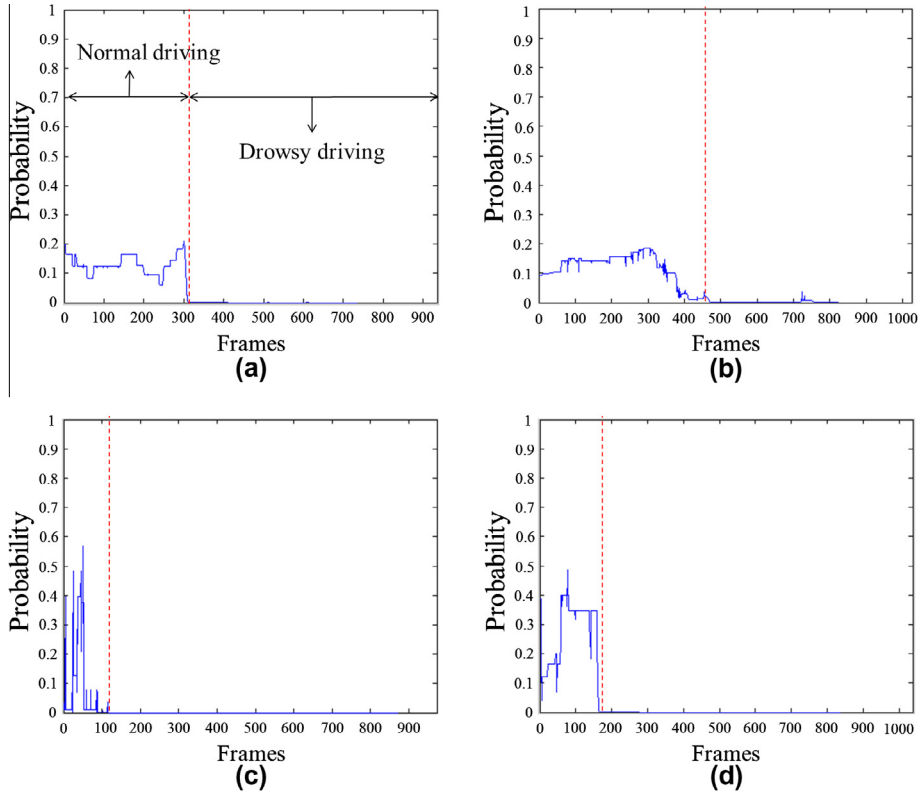

Fig. 15. Some examples of the drowsiness detection using the 2D Gaussian model for four drivers: (a) driver 1, (b) driver 2, (c) driver 3, and (d) driver 4.

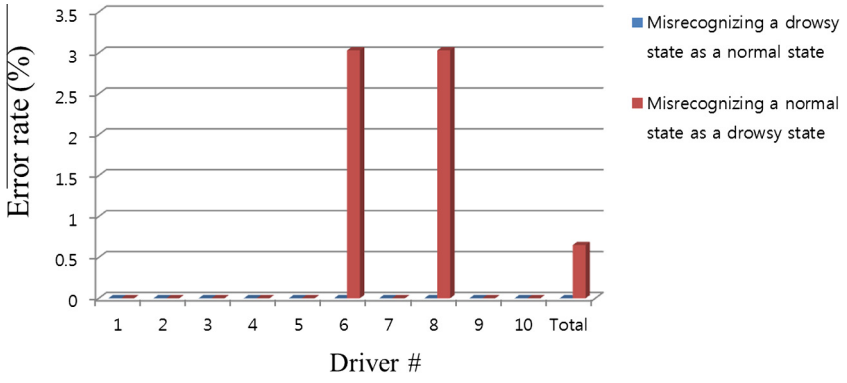

Fig. 16. Drowsiness detection results for each subject.

in eye state classification. Images of Table 2 were used as training data. Images of Tables 3 and 4 were used as testing data to measure performances of eye state classification and drowsiness detection.

3.2. Experimental results

3.2.1. Fusion method

In the first experiment, we began by measuring eye state classification performance using score-level fusions. As explained in Section 2.2.2, seven fusion methods (mean, weighted mean, min, max, product, weighted product and SVM) were employed to fuse SVM decision scores for both eyes. The classifiers for these methods were trained using the database profiled in Table 2, and the classification performance of the methods was then tested using the database profiled in Table 3. Fig. 12 shows the results of these tests.

Next, the classification performance of our proposed feature-level fusion method was tested using the same test database. This performance, along with those of all score-level fusion methods, and that of a non-fused method, are provided in Table 5 and Fig. 13. The equal error rate (EER) line seen in the graph is defined as the rate at which Type 1 errors (misclassification of closed eyes as open) and Type 2 errors (misclassifications of open eyes as closed) are equivalent. Note that all of the fusion methods showed better performance than the non-fusion method. Among the fusion methods, feature-level fusion using SVM with the RBF kernel demonstrated the lowest EER.

3.2.2. User-specific threshold method

In the second experiment, we measured the performance of our eye state classification method when a user-specific threshold is

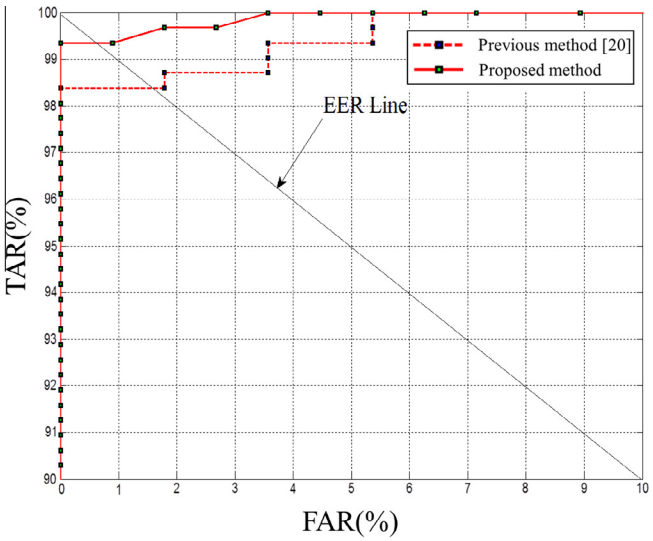

Fig. 17. ROC curves for drowsiness detection accuracy.

employed. This user-specific threshold was obtained during an initial (normal) driving period based on the feature-level fusion using the RBF kernel-based SVM, the most accurate fusion method. This threshold was then applied to eye classification during the remainder of the driving session. The EERs for fixed threshold and user-specific threshold detection are provided in Table 6, showing the significant improvement in accuracy when using the user-specific threshold.

3.2.3. Drowsiness detection results

After PERCLOS and ECD are obtained from eye state classification, the system decides the drowsiness of a driver using the 2D Gaussian model given in Eq. (15). Fig. 14 shows the distribution of PERCLOS and ECD in normal and drowsy driving sequences for each driver. Classification using this two-dimensional model clearly outperforms classification using just one dimension. As shown in the normal driving sequence in Fig. 14(c), the PERCLOS value indicates normal driving, while the ECD value indicates drowsy driving, and vice versa in the normal driving sequence in Fig. 14(d). This discrepancy can be resolved by combining the two drowsiness measures. However, as shown in Fig. 14(b), since all drivers have different eye shapes and textures, the values of the two measures can be exceeded even in the normal driving sequence. As mentioned above, this problem can be solved by applying a user-specific method using 2D Gaussian model. Fig. 15 provides some examples of drowsiness detection results using the 2D Gaussian model. The drivers were told to drive normally during the initial driving period and then to pretend to be drowsy during the remainder of the session. Thus, a high probability is obtained during the initial period and almost zero probability during the remainder. That is, the lower the probability obtained by 2D Gaussian PDF, the higher the probability of drowsy driving.

Drowsiness detection was performed on image sequences from the testing database profiled in Table 3. For each of the subjects, 72 drowsy driving sequences and 189 normal driving sequences were recorded during the daytime, and 40 drowsy driving sequences and 120 normal driving sequences were recorded during the nighttime. For each of the drowsy driving sequences, about 5–10 min of observation was recorded. The subjects were told to look forward for 5 s and then drive normally or drowsily for the remainder of the session. For each normal driving sequence, about 1 min of observation was recorded, during which the subjects were told to drive normally.

Detection results for these video sequences are provided in Table 7 and Figs. 16 and 17. As seen in Table 7, the rate of Type 1 errors (drowsiness detection failures) is 0% while the rate of Type 2 errors (false alarms) is about 0.65%. Whereas in previous work (Jo et al., 2011), differences in eye size and blinking frequency were the main causes for Type 1 errors, in our work, we were able to directly addressing these variations and virtually eliminate Type 1 errors.

Note that user-specific thresholds were specifically adjusted to achieve the 0% rate of Type 1 errors. Our assumption was that while an occasional false alarm may annoy the driver, reliable drowsiness detection is the overriding priority. In Fig. 17, we see that number of false alarms could have been reduced at the direct expense of increasing the number of detection failures, an undesirable tradeoff given our assumption.

Finally, our system was developed so that the system designer can set up the system to alert the driver in anytime between the slightly drowsy to heavily drowsy state by adjusting the window size parameter. To detect the driver drowsiness, information of previous 200 frames from the current frame is required. In this process, average delay time is 2–3 s and maximum delay is about 6 s. However, a car accident does not occur immediately at the beginning of the drowsy driving. Also delay time can be reduced by adjusting the window size but this leads to increase in false alarm rate; therefore, window size was set to 200 frames in our work.

Table 7
Drowsiness detection results measured from video sequences in a vehicle.

Error rates (%)			
Previous method (Jo et al., 2011): drowsiness detection results by PERCLOS		Proposed method: drowsiness detection results by both PECLOS and ECD	
Misrecognizing a drowsy state as a normal state (%)	Misrecognizing a normal state as a drowsy state (%)	Misrecognizing a drowsy state as a normal state (%)	Misrecognizing a normal state as a drowsy state (%)
0	1.62	0	0.65

Table 8
Average processing time of proposed driver drowsiness detection system.

Process			Average processing time (ms/frame)	
Face detection & tracking (Lee et al., 2011)			14.1	31.16 (33.06)
Eye detection & tracking (Jo et al., 2011)			14.5	
Eye state classification	Feature extraction: (PCA, LDA, Sparseness, Kurtosis)	0.15	2.55 (4.45)	
	Feature level fusion (Score level fusion)	2.4 (4.3)		
Drowsiness decision	2D Gaussian model	0.01		

3.2.4. Processing time

The proposed system software was implemented using Visual C++ on an Intel Pentium-M processor running at 1.60 GHz with 500 MB RAM. As shown in Table 8, the total average processing time for continuous input of 1000 frames was 31.16 ms/frame on the experimental hardware. Further, note that our proposed feature-level fusion method actually has a shorter processing time than less accurate score-level fusion methods.

4. Conclusion

In this paper, we proposed a driver drowsiness detection method that can be used to warn drivers of the onset of drowsiness reliably. The proposed method has three advantages over previous methods. First, it fuses information from both eyes in order to correctly classify eye state even when one eye has not been properly localized or is obstructed. Second, the method calculates a user-specific threshold for eye state classification in order to improve accuracy across a wider range of drivers, including those with small eyes or high blinking frequency. Third, the method learns the particular driver's blinking pattern by fusing two drowsiness measurements (PERCLOS and ECD) over an initial (normal) driving period.

Experimental results showed that this new eye state classification method provides robust performance under a variety of conditions and more accurate classification results than previous methods (Jo et al., 2011). In addition, the use of 2D Gaussian modeling with PERCLOS and ECD shows a further improvement of detection performance for drivers with unusual blinking patterns. These results underline the value of using an initial driving period to calibrate the detection system.

In future work, other physiological, behavioral, and visual information can be integrated with the proposed system to improve detection accuracy even further. As the system matures, its efficiency and convenience can be significantly boosted by adding user identification through facial recognition, as well as on-line update of user-specific data.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (No. 2012-0005223).

References

Adachi, J., Hayashi, T., Ogawa, K., Suzuki, T., Ishiguro, H., Nishina, T., et al. (2008). Development of a sensor detecting eyelid positions. In *15th World congress on ITS*. Jacob K. Javits Convention Center, New York City.

Bergasa, L. M., Nuevo, J., Sotelo, M. A., Barea, R., & Lopez, M. E. (2006). Real-time system for monitoring driver vigilance. *IEEE Transactions on Intelligent Transportation Systems, 7*(1), 63–77.

Bhowmick, B., & Chidanand Kumar, K. S. (2009). Detection and classification of eye state in IR camera for driver drowsiness identification. In *Proceedings of the IEEE international conference on signal and image processing applications* (pp. 340–345).

Damousis, I. G., & Tzovaras, D. (2008). Fuzzy fusion of eyelid activity indicators for hypovigilance-related accident prediction. *IEEE Transactions on Intelligent Transportation Systems, 9*(3), 491–500.

Dong, Y., Hu, Z., Uchimura, K., & Murayama, N. (2011). Driver inattention monitoring system for intelligent vehicles: A review. *IEEE Transactions on Intelligent Transportation Systems, 12*(2), 596–614.

Eriksson, M., & Papanikotopoulos, N. P. (1997). Eye-tracking for detection of driver fatigue. In *Proceedings of the IEEE conference on intelligent transportation systems* (pp. 314–319).

Ersal, T., Fuller, H. J. A., Tsimhoni, O., Stein, J. L., & Fathy, H. K. (2010). Model-based analysis and classification of driver distraction under secondary tasks. *IEEE Transactions on Intelligent Transportation Systems, 11*(3), 692–701.

Flores, M. J., & Armingol, J. M. (2010). Real-time warning system for driver drowsiness detection using visual information. *Journal of Intelligent and Robotic Systems, 59*(2), 103–125.

Ince, I. F., & Yang, T. (2009). A new low-cost eye tracking and blink detection approach: Extracting eye features with blob extraction. *Emerging Intelligent Computing Technology and Applications, LNCS, 5754*, 526–533.

Jimenez-Pinto, J., & Torres-Torriti, M. (2009). Driver alert state and fatigue detection by salient points analysis. In *Proceedings of the IEEE international conference on systems, man, and cybernetics* (pp. 455–461).

Jo, J., Lee, S. J., Lee, Y. J., Jung, H. G., Park, K. R., & Kim, J. (2010). An edge-based method to classify open and closed eyes for monitoring driver's drowsiness. In *Proceedings of the international conference of electronics, information and communication (ICEIC)* (pp. 510–513).

Jo, J., Lee, S. J., Jung, H. G., Park, K. R., & Kim, J. (2011). A vision-based method for detecting driver's drowsiness and distraction in driver monitoring system. *Optical Engineering, 50*, 127202-1–127202-24.

Kawato, S., & Ohya, J. (2000). Real-time detection of nodding and headshaking by directly detecting and tracking the between-eyes. In *Proceedings of 4th international IEEE conference on automatic face and gesture recognition* (pp. 40–45).

Kircher, A., Uddman, M., & Sandin, J. (2002). Vehicle control and drowsiness. Swedish National Road and Transport Research Institute.

Kurylyak, Y., Lamonaca, F., & Mirabelli, G. (2012). Detection of the eye blinks for human's fatigue monitoring. In *Proceedings of the IEEE international symposium on medical measurements and applications proceedings (MeMeA)* (pp. 1–4).

Lee, S. J., Jo, J., Jung, H. G., Park, K. R., & Kim, J. (2011). Real-time gaze estimator based on driver's head orientation for forward collision warning system. *IEEE Transactions on Intelligent Transportation Systems, 12*(1), 254–267.

Li, J.-w. (2008). Eye blink detection based on multiple Gabor response waves. In *Proceedings of the international conference on machine learning and cybernetics* (pp. 2852–2856).

Liu, C., & Subramanian, R. (2009). Factors related to fatal single-vehicle run-off-road crashes. National Highway Traffic Safety Administration, DOT HS 811 232.

Liu, C. C., Hosking, S. G., & Lenn, M. G. (2009). Predicting driver drowsiness using vehicle measures: Recent insights and future challenges. *Journal of Safety Research, 40*(4), 239–245.

Martinez, A. M., & Kak, A. C. (2001). PCA versus LDA. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 23*(2), 228–233.

Minkov, K., Zafeiriou, S., & Pantic, M. (2012). A comparison of different features for automatic eye blinking detection with an application to analysis of deceptive behavior. In *Proceedings of the international symposium on communications control and signal processing (ISCCSP)* (pp. 1–4).

Murphy-Chutorian, E., Doshi, A., & Trivedi, M. M. (2007). Head pose estimation for driver assistance systems: A robust algorithm and experimental evaluation. In *Proceedings of 10th international IEEE conference on intelligent transportation systems* (pp. 709–714).

Noguchi, Y., Shimada, K., Ohsuga, M., Kamakura, Y., & Inoue, Y. (2009). The assessment of driver's arousal states from the classification of eye-blink patterns. In *Proceedings of the international conference on engineering psychology and cognitive ergonomics* (pp. 414–423). Berlin Heidelberg: Springer-Verlag.

Orazio, T. D. '., Leo, M., Guaragnella, C., & Distante, A. (2007). A visual approach for driver inattention detection. *Pattern Recognition, 40*(8), 2341–2355.

Panning, A., Al-Hamadi, A., & Michaelis, B. (2011). A color based approach for eye blink detection in image sequences. In *Proceedings of the IEEE international conference on signal and image processing applications (ICSIPA)* (pp. 40–45).

Parmar, N. (2002). Drowsy driver detection system. Engineering Design Project Thesis, Ryerson University.

Patel, M., Lal, S. L. L., Kavanagh, D., & Rossiter, P. (2011). Applying neural network analysis on heart rate variability data to assess driver fatigue. *Expert Systems with Applications, 38*, 7235–7242.

Rongben, W., Lie, G., Bingliang, T., & Lisheng, J. (2004). Monitoring mouth movement for driver fatigue or distraction with one camera. In *Proceedings* of the *IEEE international conference* on intelligent *transportation systems* (pp. 314–319).

Ross A., & Govindarajan R., (2004). Feature level fusion in biometric system. In *Proceedings of the biometric consortium conference (BCC)*. Crystal City.

Saradadevi, M., & Bajaj, P. (2008). Driver fatigue detection using mouth and yawning analysis. *International Journal of Computer Science and Network Security, 8*(6), 183–188.

Shuyan, H., & Gangtie, Z. (2009). Driver drowsiness detection with eyelid related parameters by support vector machine. *Expert Systems with Applications, 36*, 7651–7658.

Sukno, F. M., Pavani, S. K., Butakoffand, C., & Frangi, A. F. (2009). Automatic assessment of eye blinking patterns through statistical shape models. In *ICVS 2009. LNCS* (Vol. 5815, pp. 33–42). Berlin: Springer.

Torkkola, K., Massey, N., & Wood, C. (2004). Driver inattention detection through intelligent analysis of readily available sensors. *Proceedings of the IEEE International Conference on Intelligent Transportation Systems*, 326–331.

Tran, Y., Craig, A., Wijesuriya, N., & Nguyen, H. (2010). Improving classification rates for use in fatigue countermeasure devices using brain activity. In *Proceedings of IEEE international conference on engineering in medicine and biology society (EMBC)* (pp. 4460–4463).

Tsuchida, A., Bhuiyan, M.S., & Oguri, K. (2010). Estimation of drivers' drowsiness level using a neural network based 'error correcting output coding' method. In *Proceedings of the IEEE international conference on intelligent transportation systems* (pp. 1887–1892).

Turk, M., & Pentland, A. (1991). Eigenfaces for recognition. *Journal Cognitive Neuroscience, 3*, 71–86.

Uliyar, M., & Ukil, S. (2012). A fast blink detector using canonical correlation analysis. In *Proceedings of the IEEE international conference on consumer electronics (ICCE)* (pp. 33–34).

Vapnik, V. (1999). An overview of statistical learning theory. *IEEE Transactions on Neural Networks, 10*(5), 988–999.

Vural, E., Çetin, M., Ercil, A., Littlewort, G., Bartlett, M. S., & Movellan, J. R. (2007). Drowsy driver detection through facial movement analysis. In *Proceedings of the IEEE international conference on human–computer interaction* (pp. 6–18).

Wang, L., Ding, X., Fang, C., & Liu, C. (2009). Eye blink detection based on eye contour extraction. *Proceedings of SPIE Image Processing: Algorithms and Systems VII, 7245*. 72450R(1-7).

Wierwille, W. W., Ellsworth, L. A., Wreggit, S. S., Fairbanks, R. J., & Kim, C. L. (1994). Research on vehicle-based driver status/performance monitoring: development, validation, and refinement of algorithms for detection of driver drowsiness. National Highway Traffic Safety Administration, Final Report: DOT HS 808 247.

Wu, J. D., & Chen, T. R. (2008). Development of a drowsiness warning system based on the fuzzy logic images analysis. *Expert Systems with Applications, 34*, 1556–1561.

Wu, J., & Trivedi, M. M. (2007). Simultaneous eye tracking and blink detection with interactive particle filters. *EURASIP Journal on Advances in Signal Processing, 2008*, 823695-1–823695-20.

Yang, J. H., Mao, Z.-H., Tijerina, L., Pilutti, T., Coughlin, J., & Feron, E. (2009). Detection of driver fatigue caused by sleep deprivation. *IEEE Transactions on Systems, Man, Cybernetics. A, Systems Humans, 39*(4), 694–705.

Yunq, L., Meiling, Y., Xiaobing, S., Xiuxia, L., & Jiangfan, O. (2009). Recognition of eye states in real time video. In *Proceedings of the international conference on computer, engineering and technology* (pp. 554–559).