

# Real-Time Vision Based Driver Drowsiness Detection Using Partial Least Squares Analysis

K. Selvakumar<sup>1</sup> · Jovitha Jerome<sup>1</sup> · Kumar Rajamani<sup>2</sup> · Nishanth Shankar<sup>3</sup>

Received: 10 December 2014 / Revised: 17 May 2015 / Accepted: 27 October 2015 / Published online: 1 December 2015  
© Springer Science+Business Media New York 2015

**Abstract** Robust eye state classification in real-time is very crucial for automatic driver drowsiness detection to avoid road accidents. In this paper, we propose partial least squares (PLS) analysis based eye state classification method and its real-time implementation on resource constraint digital video processor platform, to monitor the eye state during all time driving conditions. The drowsiness is detected using percentage of eye closure (PERCLOS) metric. In this approach, face in the infrared (IR) image is detected using Haar features based cascaded classifier and within the face, eye is detected. For binary eye state classification, PLS analysis is applied to obtain the low dimensional discriminative subspace, within which simple PLS regression score based classifier is used to classify test vector into open and closed. We compared our algorithm to recent methods on challenging test sequences and the result shows superior performance. The results obtained during on-vehicle testing show that the proposed system achieves significant improvement in classification accuracy at nearly 3 frames per second.

**Keywords** Embedded vision system · Face detection · Eye detection · Dimensionality reduction · Partial least squares · Drowsiness detection

---

✉ K. Selvakumar  
kskumareee@gmail.com

<sup>1</sup> Department of Instrumentation and Control Systems Engineering, PSG College of Technology, Coimbatore, Tamilnadu 641004, India

<sup>2</sup> Robert Bosch Engineering and Business Solutions Limited, Bangalore, India

<sup>3</sup> Lensbricks Technology Private Ltd, Bangalore, India

## 1 Introduction

Recent study on traffic safety [1] says nearly 1 in 4 drivers reported having driven when they were fatigue that had a hard time keeping their eyes open during last 30 days. In addition to this, American National Highway Traffic Administration (NHTSA) report says, 91.2 % of the single vehicle run-off-road (ROR) or on road (OR) accidents occur due to drowsiness [2]. Hence, to prevent these accidents, onboard driver drowsiness detection in vehicles is necessary. The drowsiness can be assessed using different measures [3] such as vehicle behavior [4], physiological features [5–7], and visual features [8–13]. In vehicle based measures, a number of metrics including lane departure, steering wheel movement, and pressure on the acceleration pedal are constantly monitored to detect the driver drowsiness. The main drawback of these methods is that their accuracy depends on the individual characteristics of the vehicle and its driver. In contrast, many researchers have used the features of physiological signals like electrocardiogram (ECG) [5], electroencephalogram (EEG) [6], and electro-oculogram (EOG) [7] to detect drowsiness. However, onboard acquisition of these signals may cause discomfort to the driver. Finally, the methods based on visual features detect drowsiness by using noninvasive visual information's of driver like yawning [8], facial expressions [9], head movement [10] and eye state [11, 12]. Due to noncontact in nature, the visual feature based approaches have emerged as the promising field of research for drowsiness detection. The methods based on yawning and head movement cannot detect the onset of drowsiness reliably since these are not directly indicating the drowsiness. On the other hand, eye state (open/close) information is well suited for such systems since the closing of eyes and the unusual blinking pattern have been shown to directly indicate the onset of the drowsiness.

Methods using eye state to detect driver drowsiness have generally done by measuring eye blink frequency, eye closure duration (ECD), and percentage of eye closure (PERCLOS) [11, 12]. Among them, PERCLOS is reported to be the most reliable measure for drowsiness detection due to its robustness against few classification errors when compare to ECD. PERCLOS is defined as proportion of times eyelids are closed at least 80 % over certain time period. Higher value of PERCLOS value indicates higher drowsiness level and vice versa. Real-time calculation of PERCLOS involves three important steps as follows; (i) face detection, (ii) eye detection, and (iii) eye state classification. In this paper, we have implemented real-time drowsiness detection system on an embedded video processor suitable for an automobile environment. The robustness and speed of the developed system are evaluated on test vehicle against different real-world scenarios, like extreme sunlight illumination and night time driving conditions.

The remainder of this paper is organized as follows. In section 2, we review the literature and describe the existing issues. In section 3, the proposed method for face detection, eye detection and eye state classification and its embedded implementation are discussed. Extensive evaluation results are given in section 4 and our conclusion is given in section 5.

## 2 Literature Review

In this section, recent developments on embedded vision systems are briefed, followed by the review of literatures on real-time face detection and those related to binary classifiers.

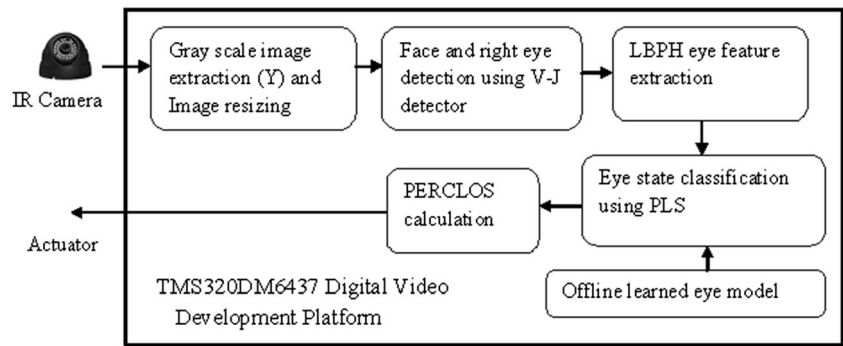
Nowadays automated vision systems are implemented as a firmware in advanced embedded video processors. Recently Cernium Corporation has developed a standalone product called *archerfish* ([http://ipvm.com/report/Archerfish\\_Solo\\_test\\_results](http://ipvm.com/report/Archerfish_Solo_test_results)), which detects and reports various events of interest in surveillance video in real time. In [14], the scope of embedded vision systems, different hardware platforms, software implementation and various optimization techniques are given in detail. It is also stated that in cases where a large number of different algorithms are involved, each running for short time, the software programmable architectures have an advantage over hardware programmable approaches. Among the available embedded hardware platforms like Digital Signal Processor (DSP), Field Programmable Gate Array (FPGA), and Graphical Processing Unit (GPU), we have chosen DSP for our case study for its advantageous development time, flexibility, and video friendly features like hardware image resizer, despite its higher power consumption.

For effective eye detection, robust frontal face detection under various illuminations and expressions is very crucial. Thorough review on much of face detection research can be found in [15, 16]. But most of the detection algorithms are concerned only about the detection accuracy and not the computation time. In

neural network based face detection system [17], it is reported that 2 to 4 s is required for a mere  $320 \times 240$  pixel image. Skin color based face detection [18] also achieves better detection rate but it suffers heavily due to cluttering. Viola and Jones (V-J) [19], have proposed a robust real-time face detection algorithm by using integral image and Haar cascaded classifier. In [20], Wang et al., have proposed schemes to optimize the V-J detector for real-time face detection on smart cameras. In order to reduce the search windows, skin color information is integrated with V-J detector and the result is reported as 28 ms on an Intel Xscale microprocessor PXA270 with 640 MHz for  $640 \times 480$  image. However, this method cannot be used for IR images due to lack of color information. In [11], Dasgupta et al., implemented V-J detection algorithm for monitoring loss of attention on automotive drivers and achieved 9.5 frames per second (fps) on an Intel atom processor with 1 GB RAM and 1.6 GHz clock. Due to the slight movement of the driver, once the face is detected, a kalman filter based tracker is employed to reduce the face search space in the consecutive frames. In that work, color images are used during day time and IR images during night. This approach requires different preprocessing algorithms at different time for illumination normalization which may degrade the classifier accuracy. Overall, the computational simplicity and robustness of V-J detector has motivated us to use it in our face and eye detection framework. To significantly reduce the drastic illumination variation between day and night, IR camera is used for all the time [21]. The main purpose of infrared illuminator is, it minimizes the impact of different ambient light conditions, therefore retaining image quality under varying real-world conditions including poor illumination, day and night. To combat the slight illumination variation in IR images of day and night, block local binary pattern histogram (LBPH) method is used [22].

In recent works of driver drowsiness detection [11, 12], eye state is classified into open and closed using a simple binary classifier called support vector machines (SVMs) [23]. For real-time eye state classification problem, obtaining low dimensional discriminative subspace for dimensionality reduction is of crucial importance. In their work, SVM classifier is learned on the low dimensional training feature vectors obtained using principal component analysis (PCA). However, for dimensionality reduction, supervised methods are more advantageous over unsupervised methods due to their class label incorporation with the training data. PCA is an unsupervised learning method since it does not use any class labels. It has been reported in [24] that PLS based supervised subspace learning method can significantly outperform PCA with respect to classification accuracy and computation time. Recently, this method of dimensionality reduction is used in different applications like vehicle detection [25], object tracking [26] and face recognition [27]. We take the advantage of nature of eye state classification problem by employing PLS analysis to extract low dimensional subspace. In our work, steps performed in eye state classification are as follows. For each frame, after eye detection, the feature vector is extracted using LBPH and projected

**Fig. 1** Proposed embedded vision system for driver drowsiness detection



onto PLS subspace, resulting in an extremely low dimensional feature vector. Then simple and efficient classifiers like linear SVM and PLS regression is applied on the low dimensional training feature vectors to learn and classify the eye state. Numerous experiments and evaluation on challenging test dataset bear out that the proposed algorithm is efficient and effective when compare to other representative approaches. After eye state classification, PERCLOS metric is calculated, and based on that driver’s drowsiness is detected.

The entire drowsiness detection system has been implemented on an embedded processor and testing has been carried out onboard for both day and night driving conditions. The computational and memory requirements for different algorithms involved are analyzed in detail to evaluate the processor performance.

### 3 Embedded Vision System Framework

The functional block diagram for embedded vision based driver drowsiness detection system is shown in Fig. 1. For each block, real-time implementation procedure and time consumption are discussed in detail.

#### 3.1 Image Resizing

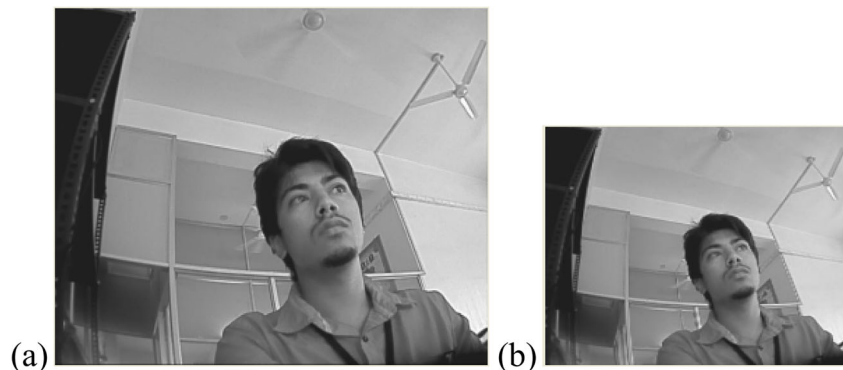
In the hardware setup, output obtained from the analog camera is in the YCbCr color space. It has been digitized to a 576×

720 resolution using the on-board video encoder. Since the proposed system does not require color information, only the grayscale component i.e. Luma (Y) is retained and other Chroma (Cb, Cr) components are discarded. Image resizing is mainly performed in order to reduce the memory and computation time without compromising the detection rate. This is achieved by down scaling the input frame using the hardware resizer available in the DSP processor. The input frame is downscaled to the desired frame size by configuring the image resizer hardware registers like output frame width and height, input frame width and height and downsampling filter coefficients. The filter coefficients are generated offline using the procedure given in [28]. The original and resized images are shown Fig. 2a, b respectively.

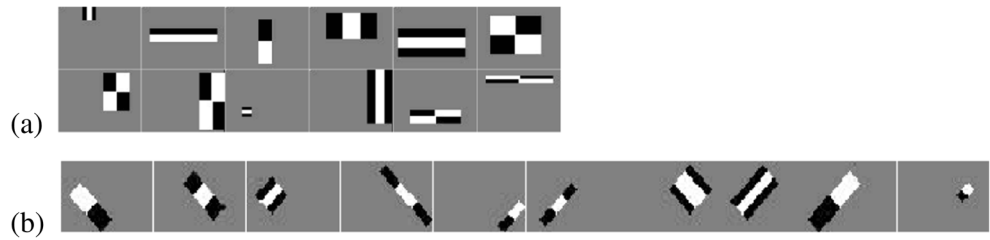
#### 3.2 Face and eye Detection

On the resized image, V-J detector is applied to detect the face. The V-J detector achieves robust face detection in real-time by extracting Haar like features using integral image. Two, three and four rectangle types of Haar feature kernels are used at various stages of classification are shown in Fig. 3. They are just like the convolutional kernel and each feature is a single valued obtained by subtracting sum of pixels under white rectangle from sum of pixels under block rectangle. For robust eye detection, tilted Haar feature kernels are used to effectively capture the curves.

**Fig. 2** a Input IR gray scale image (576× 720) b Hardware resized image (288× 360)



**Fig. 3** Example **a** Haar feature kernels used in V-J face detector and **b** tilted Haar feature kernels used in V-J eye detector



To avoid the huge computations involved in Haar feature extraction, integral images are used [19]. Each element of integral image is sum of all pixels in upper left region as given below;

$$F(x,y) = f(x,y) - F(x-1,y-1) + F(x-1,y) + F(x,y-1) \quad (1)$$

$$F(x,y) = \sum_{x' \leq x} \sum_{y' \leq y} f(x',y') \quad (2)$$

where  $f(x,y)$  is the intensity at the point  $(x,y)$ . The extracted single valued Haar features are used in cascaded weak classifiers to decide whether the window formed from that point is a face or a non-face. Although feature extraction has simple computation, extraction and evaluation of all possible Haar features of a given image window takes huge computation time. Therefore, to select a subset of most discriminating features to model a face, AdaBoost based feature selection method is used. By combining weak classifiers of each stage, cascaded weak classifier has been formed to reject non faces at the earliest possible stage. Both OpenCV and MATLAB computer Vision Toolbox have effectively implemented V-J detection algorithm by using an optimal set of feature extracting parameters and stage thresholds given in the *haarcascade\_frontoface\_alt2.xml* [29] file which is available in the Intel OpenCV library. We have used the same parameters for feature extraction and classification in our embedded implementation. After face detection, the right eye is detected using a V-J detector using regular and tilted Haar features. For eye detection, optimal set of feature extracting parameters and stage thresholds given in the *haarcascade\_mcs\_righteye.xml* [29] file are used.

For the V-J detector to classify a search window as face, it has to pass  $n$  cascaded stages (in our case, 20 stages). For each search window, time required for classification increases stage

after stage. It is evident that reducing the number of search windows will highly benefit real-time face detection in terms of computations required. The general assumption in a driver drowsiness detection system is that the face is not moved much in consecutive frames and number of faces in field of view is one [11]. Using this assumption, we have reduced the face search region drastically in consecutive frames based on face detected from the previous frame as shown in Fig. 4a. This approach helps in huge computation reduction for consecutive frames. If no face is detected in the reduced search space, the entire frame will be searched in the next frame.

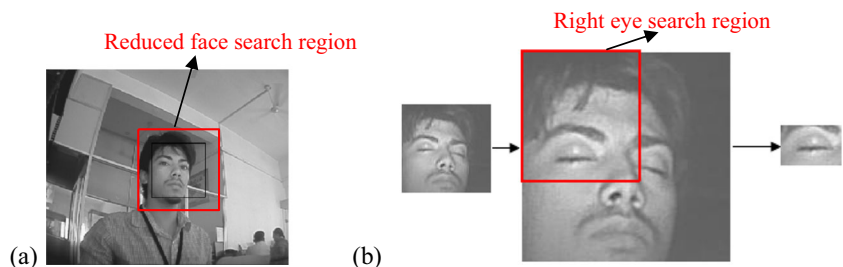
Since the face is detected in the low resolution (resized) image, the detected face is remapped onto the available higher resolution input grayscale image for robust eye detection and classification. The right eye is detected by applying V-J detector with tilted feature kernels on top left-half region of the remapped face as shown in Fig. 4b. The dimension of the detected eye image is  $37 \times 54$ .

### 3.3 Block LBP Histogram Feature Extraction

For robust eye state classification, illumination invariant description of eye region is required. This is achieved using a block LBP histogram. The LBP operator is theoretically simple yet powerful method for illumination invariant feature extraction [30]. It assigns a label to every pixel of an image by thresholding the center pixel value with the  $3 \times 3$  neighborhood of each pixel value and corresponding results as a binary number. Then histogram of labels can be used as feature descriptor. The LBP operator is defined as,

$$LBP = \sum_{i=0}^{P-1} s(g_i - g_c) 2^i \quad (3)$$

**Fig. 4** **a** Face detection in reduced search region and **b** Eye localization in remapped face image





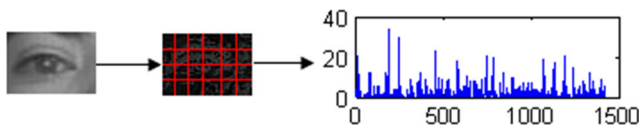


Fig. 5 Eye feature vector computation

where  $P$  is the total number of neighboring pixels  $g_c$  is the intensity value of center pixel,  $g_i$  is the intensity value of neighboring pixels and  $s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$ .

The LBP is called uniform if it contains at most two bitwise transitions from 0 to 1 or vice versa when the binary string is considered circular. For a  $3 \times 3$  neighborhood, uniform LBP can be represented using 59 bin histogram as given in [30]. However, global LBP representation of an eye region leads to loss of spatial relations. To avoid that, block LBP histogram is obtained as illustrated in Fig. 5. All local histograms obtained for each block are concatenated to form the discriminative eye feature vector. Since, the histogram is taken for each block; it is also robust against local pose variations. In our case, the detected eye image is divided into 24 blocks with dimension of  $9 \times 9$  empirically. The concatenated eye feature vector dimension is 1416 ( $24 \times 59$  bins).

### 3.4 Eye State Classification

Accurate classification of the eye state into open and closed is very crucial in estimating the PERCLOS value. The proposed PLS based dimensionality reduction technique and its implications on classifier with respect to accuracy and computation complexity is given in this section.

#### 3.4.1 Dimensionality Reduction Using PLS Analysis

PLS is a statistical learning method, which relates a set of observed variables by means of latent variables. The basic idea of PLS is to construct latent variables as linear combinations of the original predictor variables (features)  $X$  and response variables  $Y$ . The detailed theoretical background on PLS regression applied to vision can be found in [26]. A brief overview with respect to the proposed eye state classification is given in this section. Let  $X \in \mathcal{R}^{N \times m}$  be the predictor matrix

which denotes  $N$  feature vectors obtained from open and closed eye samples with a dimension  $m$  and  $Y \in \mathcal{R}^{N \times 1}$  be the response matrix which denotes class labels of the feature vectors. We use the PLS regression algorithm as a supervised subspace learning method by setting class labels for open and closed eyes to 1 and  $-1$  respectively. Both mean centered  $X$  and  $Y$  are decomposed using PLS algorithm as follows

$$X = TP^T + E; Y = UQ^T + f \tag{4}$$

where  $E \in \mathcal{R}^{N \times m}$  and  $f \in \mathcal{R}^{N \times 1}$  are the predictor and response residuals respectively.  $P \in \mathcal{R}^{m \times p}$  and  $Q \in \mathcal{R}^{1 \times p}$  are the loading matrices.  $T \in \mathcal{R}^{N \times p}$  and  $U \in \mathcal{R}^{N \times p}$  are the latent feature matrices. Using the nonlinear iterative PLS (NIPALS) algorithm [17],  $p$  PLS weight vectors is constructed, and stored in matrix  $W = [w_1, \dots, w_p] \in \mathcal{R}^{m \times p}$ , such that

$$[\text{cov}(t_i, u_i)]^2 = \max_{|w_i|=1} [\text{cov}(Xw_i, Y)]^2 \tag{5}$$

where  $t_i$  and  $u_i$  are the  $i^{\text{th}}$  column of matrices  $T$  and  $U$  respectively. After the extraction of  $t_i$  and  $u_i$ , matrices  $X$  and  $Y$  are deflated by subtracting their rank one approximations. This process is repeated until the residuals are very small or for  $p$  number of latent vectors. The optimal  $p$  value can be chosen based on amount of cumulative variances by cross validation. Here,  $W$  constructs a new bases matrix which has a large covariance with response variables and gives better discriminative strength to the target classes. The low dimensional latent feature matrix  $T$  is obtained by projecting original feature space  $X$  onto the latent feature space  $W$  as shown below;

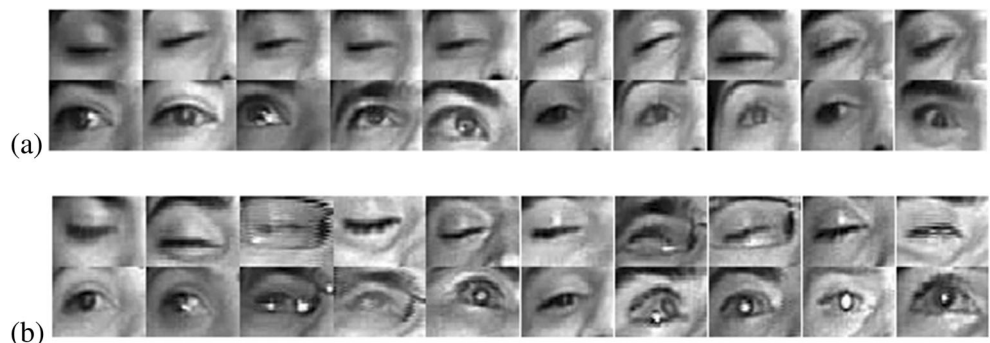
$$T = XW \tag{6}$$

Then, within this low dimensional subspace, any binary classifier like SVM can be applied. The dimensionality reduction for a new observation vector  $v \in \mathcal{R}^{1 \times m}$  can be done by projecting it on  $W$  as shown below;

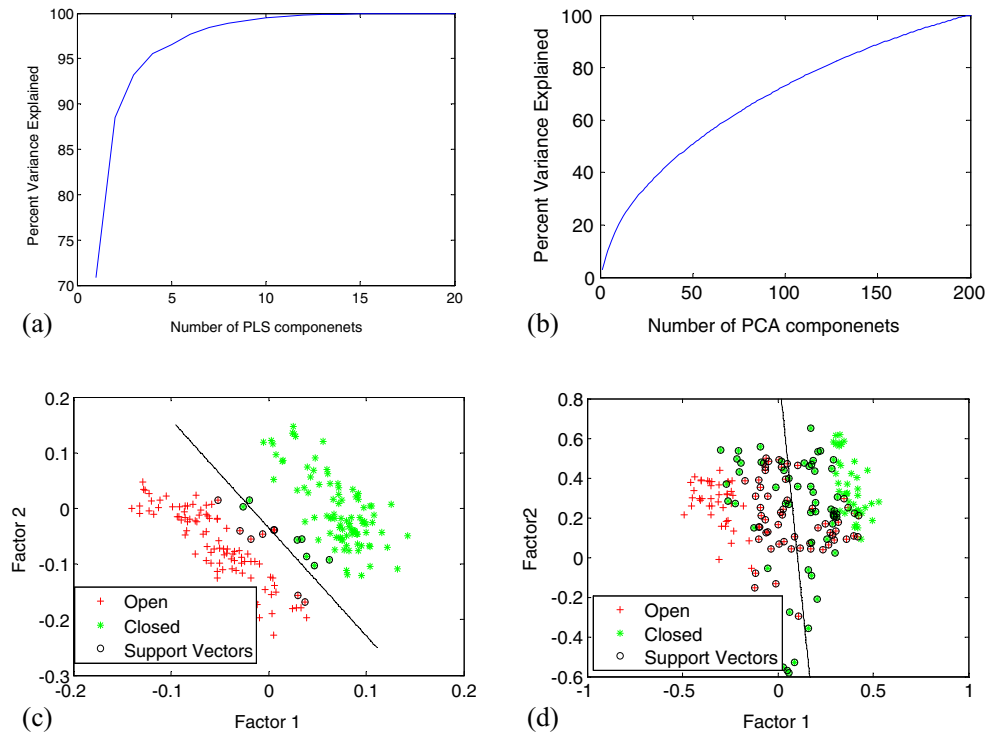
$$d = W^T v \in \mathcal{R}^{p \times 1} \tag{7}$$

where  $d$  is the observation vector with dimension  $p$  ( $p < m$ ) and this can be classified using the linear SVM classifier.

Fig. 6 a Eye training and b testing examples



**Fig. 7** Comparison of PCA and PLS for dimensionality reduction. Variance plots for **a** PLS and **b** PCA based subspace models for the training dataset. Plots of the first two factors of the dimensionality reduced training feature set obtained using **c** PLS and **d** PCA



On the other way, eye state can be classified efficiently as follows; once the low dimensional subspace of original data is obtained, the regression coefficients  $\beta \in \mathcal{R}^{m \times 1}$  can be estimated as given in [17],

$$\beta = WQ^T = W(T^T T)^{-1} T^T Y. \tag{8}$$

The regression model is given by

$$Y = X\beta + f \tag{9}$$

The regression response  $y_v$  for a test feature vector  $v$  can be obtained by

$$y_v = Y + \beta^T v \tag{10}$$

where  $Y$  is the sample mean of  $Y$ . In this case, if  $y_v \geq 0$ ,  $v$  is classified as eye open, otherwise it is classified as eye closed. It is important to note that (10) indicates only a single dot product of test feature vector with the regression coefficients is needed for binary classification.

**Table 1** Eye state classification results

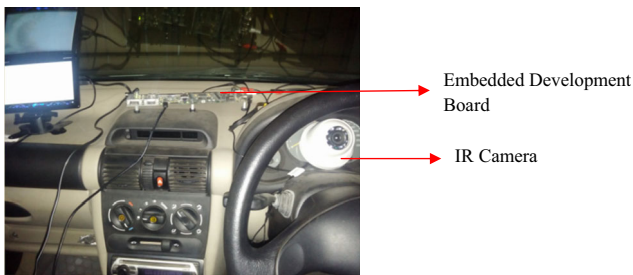
Method	tp	fn	tn	fp	tpr (%)	fpr (%)
PCA+ linear SVM [2]	526	4	392	58	99.24	12.88
PLS+ linear SVM	526	4	412	38	99.24	8.44
PLS regression	525	5	427	23	99.05	5.11

### 3.4.2 Dataset Description

The performance of proposed eye state classification algorithm is compared with that of the previous approaches in offline using MATLAB on a dataset collected using IR Camera. Our training dataset contains 200 eye images with 100 open and 100 closed. Our test dataset contains 1000 eye

**Table 2** Average time consumption for key functions of drowsiness detection on TMS320DM6437 video processor

Steps	Time Consumption (ms)
Stage 1: Data acquisition and color space transformation	
Image Resizing	0.028
Gray Scale image(Y) extraction	8
Stage 2: Face Detection	
Integral image computation	18
Cascaded classification for entire frame	568
Cascaded classification for reduced search region	57
Stage 3: Eye detection	
Eye localization in remapped image	194
LBPH representation	2
Stage 4: Eye state classification	
PLS regression	0.27
PLS + SVM Classification	6
PCA + SVM Classification	45
Stage 5: Driver drowsiness detection	
PERCLOS calculation	0.018



**Fig. 8** Experimental setup for embedded vision based driver drowsiness detection

images with 530 opened and 450 closed, captured during day and night with and without glasses from 50 subjects. The sample training and test samples are shown in Fig. 6.

### 3.4.3 Comparison of PLS and PCA Based Dimensionality Reduction

The discrimination strength of subspaces learned using PLS and PCA from our dataset has been compared with the help of linear SVM classifier. In order to select the optimal dimensions required, variance plots are used as shown in Fig. 7a, b for PLS and PCA based subspace models respectively. It is interesting to note that for the given training dataset, to achieve best classification performance, only 15 basis vectors are required for PLS method whereas nearly 180 basis vectors are required for the PCA method. Qualitatively, the advantage

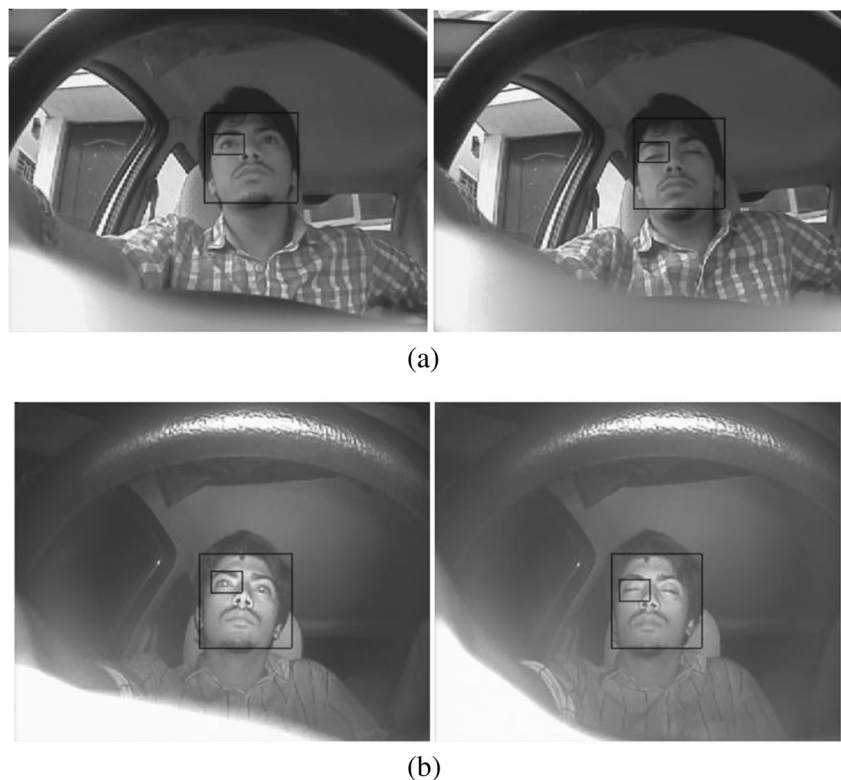
of PLS over PCA in terms of discrimination strength can be shown by plotting the first two factors of the dimensionality reduced training dataset. From Fig. 7c, d, it can be seen clearly that PLS achieves better class separation than PCA.

Moreover, as the dimensionality of the subspace increases, the time required to project the high dimensional input feature vector also increases thus leading to more time consumption. In our embedded implementation, to project input LBPH feature vector onto the 15-dimensional PLS subspace takes utmost 4 milliseconds while projection time for the 180 dimensional PCA subspace is close to 42 milliseconds. The low dimensional feature set obtained using PLS and PCA based dimensionality reduction methods are used to train the linear SVM classifier. As better discrimination is achieved using the PLS subspace, only 18 support vectors are required to get an optimal hyperplane whereas using the PCA subspace requires 173 support vectors. Thus, in addition to the superior class discrimination, the computation cost of the projection makes PLS more suitable for our real-time implementation than PCA. This characteristic also reduces the need for computationally complex non-linear SVM kernels for classification.

### 3.4.4 Classification Results

Compared to the approach given in [11], our method is evaluated against test eyes captured using IR camera during day and night with and without spectacles.

**Fig. 9** Example face and eye detected images with open and closed state during a day and b night



**Table 3** V-J detector performance for face and eye on TMS320DM6437 video processor

Methods	tpr (%)	fpr (%)
V-J Face Detector	99	$2.01 \times 10^{-3}$
V-J Eye Detector	96	$3.25 \times 10^{-3}$

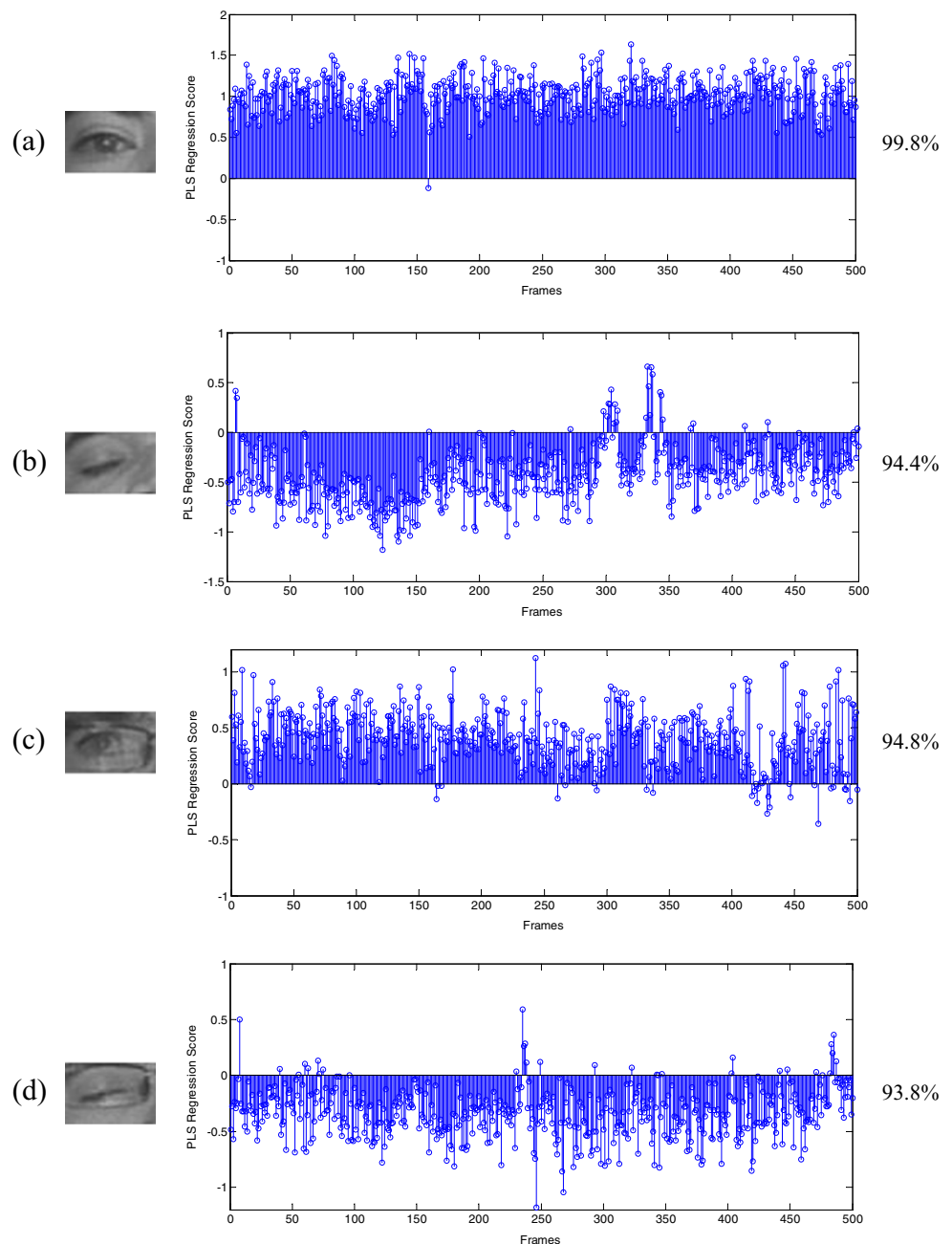
Moreover, in [11, 12], the SVM classifier has been applied after PCA based dimensionality reduction. However, the classification results provided in Table 1 reveals that with respect to false positive rate (fpr), the

proposed PLS regression and PLS based dimensionality reduction combined with linear SVM eye state classifiers significantly outperforms unsupervised PCA with linear SVM classifier. It should also be noted that with respect to driver drowsiness detection, fpr is more significant than true positive rate (tpr).

### 3.5 System Time Consumption

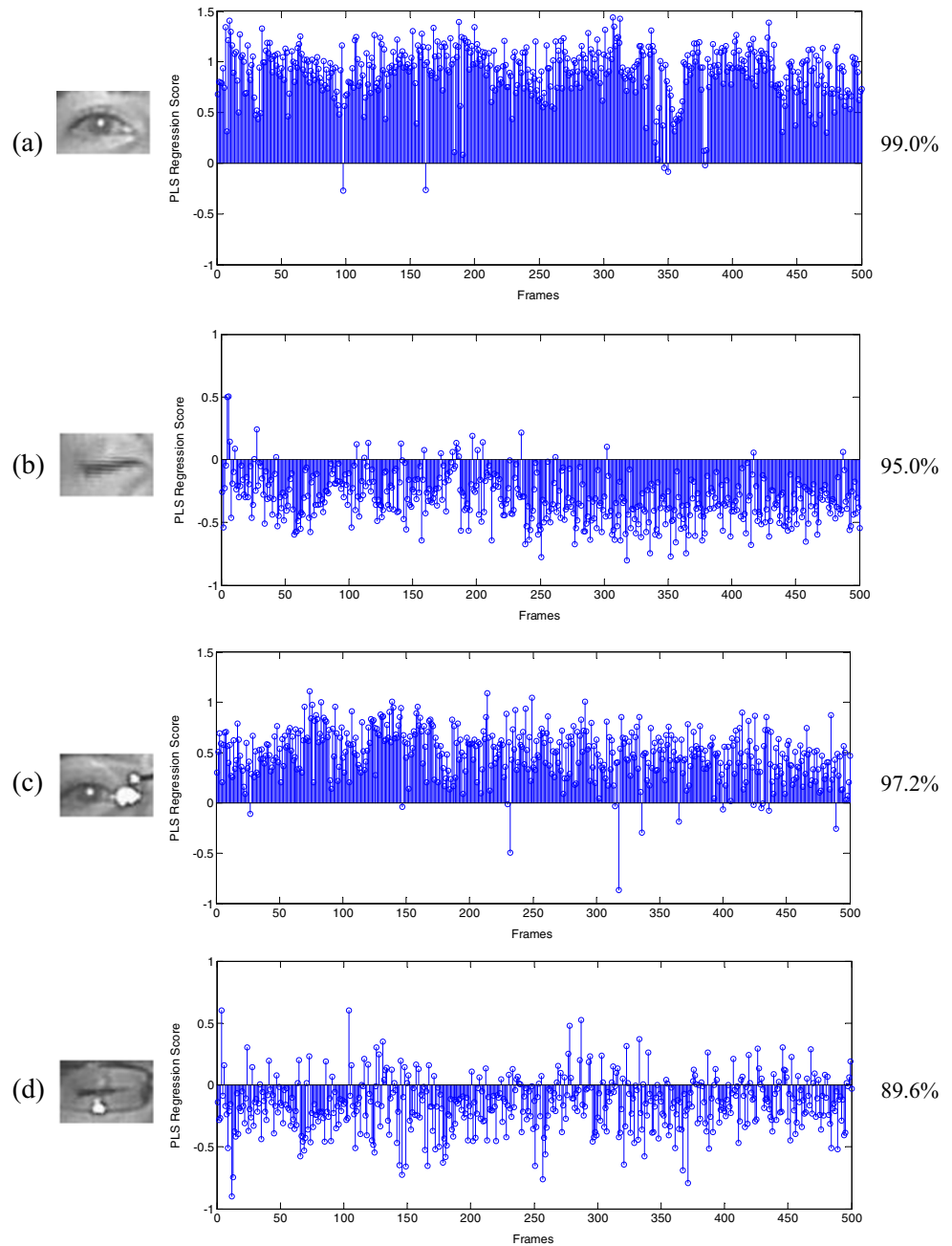
After laboratory implementation, the average computation time required for key functions of the proposed drowsiness

**Fig. 10** Eye state classification accuracy for onboard test sequence with different conditions during day time **a** open without glasses **b** closed without glasses **c** open with glass and **d** closed with glass





**Fig. 11** Eye state classification accuracy for onboard test sequence with different conditions during night time **a** open without glasses **b** closed without glasses **c** open with glass and **d** closed with glass



detection system on TMS320DM6437 video processor is measured and given in Table 2.

From Table 2, it can be observed that in face detection stage, cascaded classification for entire frame case, on average processor takes 568 ms. Once the face is detected, in consecutive frames, face search is done only in the reduced search region due to slight movement of the driver as shown in Fig. 4a. For this region, it takes only 57 ms. Then, eye is detected within the remapped high resolution face in 198 ms. If either face or eye is not detected, face search will be done in entire

next frame. It can also be noted that the proposed PLS based eye state classification functions consume very less amount of time compare to PCA based method due to the higher dimension of PCA subspace. Overall speed of the proposed system is found to be 3 fps for the case of face detection in reduced search region plus PLS regression based classification. More interestingly, our system uses only 600 MHz processor against 1.6 GHz processors used in previous approaches [11, 12]. The lower clock processor consumes less power which is a very essential characteristic for any portable application.

## 4 Real-Time Performance Evaluation

The developed drowsiness detection system has been evaluated in real-time to study its accuracy and speed. The experimental setup of the system is made inside the vehicle and IR camera is strategically placed to capture driver's eye as shown in Fig. 8. The TMS320DM6437 digital video development board is interfaced with an IR camera and 7 in. LED TV to capture and display the frames respectively. The display is partitioned to simultaneously view the results obtained at different stages in real-time. The development board is powered by +5 V external power supply. On board switching voltage regulators provide the +1.2 V CPU core voltage, +3.3 V for peripherals and +1.8 V to DDR2 memory. For all the experiments, the DSP is configured at 600 MHz clock frequency. Images used for results illustration are extracted from DDR2 DRAM memory of the development board.

### 4.1 Face and eye Detection Results

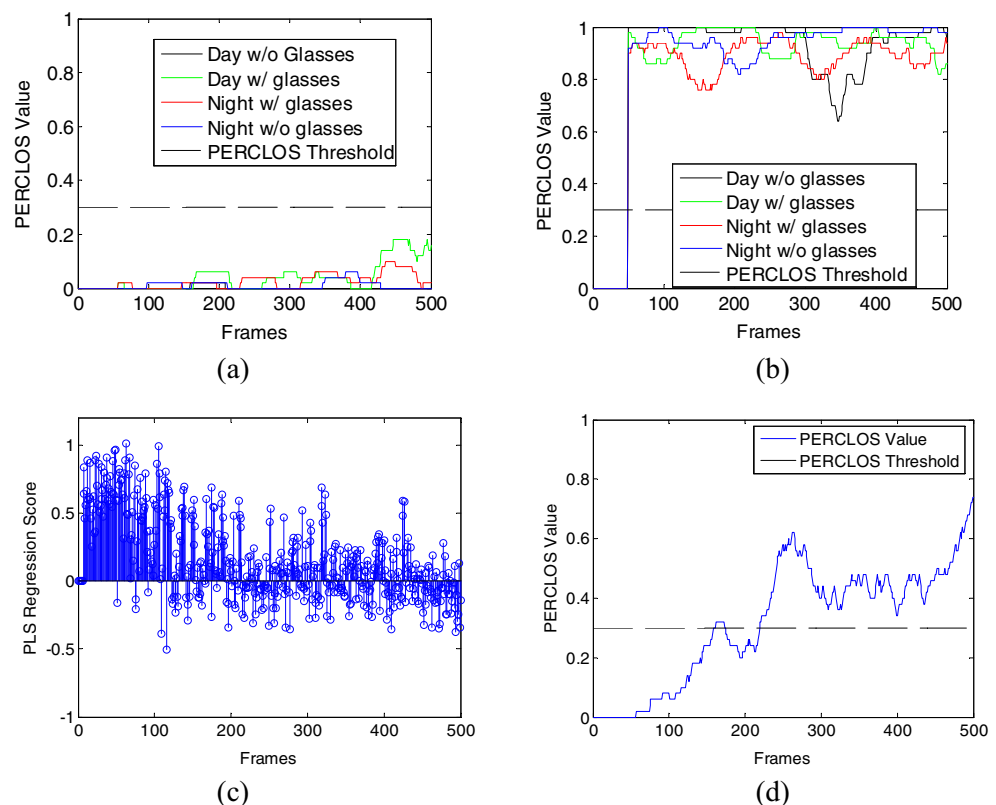
Our face detection system has been tested on-board during day and night under different challenges like extreme illumination variations, various frontal poses and expressions in a smooth road condition. Various detection results have been illustrated in Fig. 9a, b. To quantitatively evaluate the face and eye detection module, totally 10,000 frames of different

subjects are tested during day and night in real-time. It can be seen from Table 3 that the V-J detector achieves 99 % detection rate. Compared to face detection, eye detection rates are poor due to its confusion between eyelids and closed eye. It is also observed that without remapping, eye detection rates are extremely low due to low resolution face image.

### 4.2 Real-Time Eye State Classification

Based on the classification accuracy and computation time obtained in section 3, PLS regression based eye state classification method is used for on-vehicle testing. After the face and eye detection, to evaluate the classifier performance in real world driving conditions, tests are carried out for different conditions like eye closed and eye open with and without spectacles. During run time, the eye state classification outcomes like class value, score and PERCLOS values are written in DRAM memory. For each test condition, after 500 frames, the test results are sent to personal computer and analyzed in offline as shown in Figs. 10 and 11. It can be observed from Fig. 11d that classification accuracy during the night for closed eyes with glass is only 90 %, due to reflection of light by the glasses. Moreover, for all testing conditions, few classification errors in eye open cases are due to false positives and natural blinking by the drivers. Overall, the

**Fig. 12** PERCLOS values for eye state **a** open and **b** closed **c** Eye state sequence for a drowsy driver and **d** PERCLOS values drowsy driver eye state sequence



classification results are found to be accurate for different on-vehicle test conditions.

### 4.3 PERCLOS Calculation

After eye state classification, the driver drowsiness is detected by estimated PERCLOS value in real time. It is defined as [3],

$$\text{PERCLOS}[k] = \frac{\sum_{i=k-M+1}^k \text{blink}[i]}{M} \times 100 \quad (11)$$

where  $M$  is the total number of frames in the window,  $\text{PERCLOS}[k]$  is the PERCLOS value for the  $k^{\text{th}}$  frame,  $\text{blink}[i]$  represents the binary eye status of the  $i^{\text{th}}$  frame. The  $\text{blink}[i] \in \{0, 1\}$  indicates eye open and '1' indicates eye closed. The higher PERCLOS value indicates more drowsiness and lower PERCLOS value indicates more alertness. In our experiments, we have fixed the window size  $M$  as 50 frames empirically to estimate PERCLOS value in any arbitrary time. The PERCLOS value plots obtained using (11) for eye open and eye closed experiment results obtained in section 4.2 are shown in Fig. 12a, b respectively. The PERCLOS plots are used to clearly distinguish driver drowsiness and alertness and to detect the driver drowsiness PERCLOS value threshold is set to 0.3 empirically. To comprehensively validate the proposed system, it is tested on drowsy driver and his eye state classification sequence is shown in Fig. 12c. The PERCLOS metric plot for this eye state sequence clearly detects the driver drowsiness as shown in Fig. 12d.

From the obtained results, it is evident that the proposed system may be used as safety indicator to prevent a number of road accidents caused due to drowsiness. Once drowsiness is detected, a haptic feedback can be delivered to the driver to help regain his alertness.

## 5 Conclusion

In this paper, the design and implementation strategies for driver drowsiness detection using a video processor has been presented. In this approach, for robust detection of face for a specific scale in IR images, Haar features based cascaded classifier is used. The right eye is detected by applying V-J detector on remapped face image. To normalize the illumination variations, block LBPH features are extracted. The binary nature of eye state classification problem enables us to employ the supervised partial least squares analysis to obtain low dimensional subspace which eventually increases the inter-class distance. Experimental results on challenging test sequence show that the proposed PLS regression score based classifier outperformed other methods with least false positive rate of 5 %. Finally, based on the eye state classification, PERCLOS value is calculated for a 50 frames sliding window which

effectively indicates the drowsiness of the driver. The average speed of the proposed embedded vision system is 3fps which is sufficient enough for accurate detection of drowsiness. The developed system is tested in both laboratory and on-vehicle conditions during day and night.

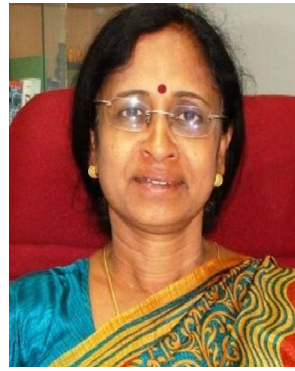
## References

1. 2014 Traffic safety culture index (2015). AAA Foundation for Traffic Safety, Washington.
2. Liu, C., & Subramanian, R. (2009). Factors related to fatal single-vehicle run-off-road crashes. *National Highway Traffic Safety Administration, DOT HS, 811*, 232.
3. Gupta, S., Kar, S., & Routray, A. (2010). Fatigue in human drivers: A study using ocular, psychometric, physiological signals. *Proceedings of IEEE. Students Technology Symposium*, 234–240.
4. Liu, 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.
5. Patel, M., Lal, S. L., Kavanagh, D., & Rossitier, P. (2011). Applying neural network analysis on heart rate variability data to assess driver fatigue. *Journal of Expert Systems with Applications*, 38(6), 7235–7242.
6. 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* (pp. 1887–1892).
7. Shuyan, H., & Gangtie, Z. (2009). Driver drowsiness detection with eyelid related parameters by support vector machine. *Journal of Expert Systems with Applications*, 36(4), 7651–7658.
8. Rongben, W., Lie, G., Bingliang, T., & Lisheng, J. (2004). Monitoring mouth movement for driver fatigue or distraction with one camera. In *Proceedings of IEEE international conference on intelligent transportation systems* (pp. 314–319).
9. Vural, E., Cetin, M., Ercil, A., Littlewort, G., Bartlett, M., & Movellan, J. (2007). Drowsy driver detection through facial movement analysis. *Lecture Notes in Computer Science*, 4796, 6–18.
10. Dongwook, L., Seungwon, O., Seongkook, H., & Minsoo, H. (2008). *Drowsy driving detection based on the driver's head movement using infrared sensors* (pp. 231–236). Communication: International Symposium on Universal.
11. Dasgupta, A., George, A., Happy, & S. L., Routray, A. (2013). A vision-based system for monitoring loss of attention in automotive drivers. *IEEE Transactions on Intelligent Transportation Systems*, 14(4), 1825–1838.
12. Jo, J., Lee, S. J., Park, K. R., Kim, I., & Kim, J. (2014). Detection driver drowsiness using feature-level fusion and user-specific classification. *Journal of Expert Systems with Applications*, 41(4), 1139–1152.
13. Jo, J., Lee, S. J., Jung, H. G., Park, K. R., Kim, I., & Kim, J. (2011). A vision-based method for detecting driver's drowsiness and distraction in driver monitoring system. *Optical Engineering*, 50(11), 1–24.
14. Kisananin, B., & Nikolic, Z. (2010). Algorithmic and software techniques for embedded vision on programmable processors. *Signal Processing: Image Communication*, 25(5), 352–362.
15. Yang, M. H., Kriegman, D. J., & Ahuja, N. (2002). Detecting faces in images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1), 34–58.

16. Zhang, C., & Zhang, Z. Y. (2010). A survey in recent advances in face detection. Technical Report, 66, Microsoft Research.
17. Rowley, H. A., Balija, S., & Kanade, T. (1998). Neural network based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1), 23–38.
18. Chai, D., & Ngan, K. N. (1999). Face segmentation using skin color map in videophone applications. *IEEE Trans. On Circuits and Systems for Video Technology*, 9(4), 551–564.
19. Viola, P., & Jones, M. (2004). Robust real-time face detection. *International Journal of Computer Vision*, 57(2), 137–154.
20. Wang, Q., Wu, J., Long, C., & Li, B. (2013). P-FAD: real time face detection scheme on embedded smart cameras. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 3(2), 210–222.
21. Zhiwei, Z., & Qiang, J. (2005). Robust real-time eye detection and tracking under variable lighting conditions and various face orientations. *Computer Vision and Image Understanding*, 98(1), 124–154.
22. Ojala, T., Pietikainen, M., & Maenpaa, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary pattern. *IEEE Trans. On Pattern Analysis and Machine Intelligence*, 24(7), 971–987.
23. Vapnik, V. (1999). An overview of statistical learning theory. *IEEE Transactions on Neural Networks*, 10(5), 988–999.
24. Schwartz, W. R., Kembavi, A., Harwood, D., & Davis, L. S. (2009). Human detection using partial least squares analysis. In *Proceedings of international conference on computer vision* (pp. 24–31).
25. Kembhavi, A., Harwood, D., & Davis, L. S. (2011). Vehicle detection using partial least squares. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(6), 1250–1265.
26. Wang, Q., Chen, F., Xu, W., & Yang, M. (2012). Object tracking via partial least squares analysis. *IEEE Transactions on Image Processing*, 21(10), 4454–4465.
27. William, R. S., Huimin, G., Jonghyun, C., & Larry, S. D. (2012). Face identification using large feature sets. *IEEE Transactions on Image Processing*, 21(4), 2245–2255.
28. SPRAAI7B (2008). Understanding the DaVinci Resizer. Texas Instruments.
29. Castrillon, M., Deniz, O., Guerra, C., & Hernandez, M. (2007). ENCARA2: real-time detection of multiple faces at different resolutions in video streams. *Journal of Visual Communication and Image Representation*, 18(2), 130–140.
30. Ahonen, T., Hadid, A., & Pietikainen, M. (2006). Face description with local binary patterns: application to face recognition. *IEEE Trans. On Pattern Analysis and Machine Intelligence*, 28(12), 2037–2041.



**K. Selvakumar** received his PhD degree in electrical engineering from Anna University, Chennai, India in 2015. His research interests include Signal Processing, Computer Vision and Embedded Vision Systems.



**Jovitha Jerome** received her BE (EEE) degree from College of Engineering, Guindy, Chennai. She received her ME and PhD degrees in Electrical Engineering from Government College of Technology, Coimbatore, India and Asian Institute of Technology, Bangkok, Thailand respectively. She has published numerous papers in international journals and conferences. Currently she is working as Professor at PSG College of Technology, Coimbatore, India. Her research interests include Signal Processing and Machine Learning.



**Kumar Rajamani** received his MSc (Mathematics) and MTech (CSc) from Sri Sathya Sai Institute of Higher Learning. He received his PhD degree in Biomedical Engineering from University of Bern, Switzerland. He has three patents to his credit and has published extensively in international journals and conferences. He is currently working as architect at Robert Bosch Engineering and Business Solutions Limited, Bangalore, India. His areas of interests include image processing, medical image analysis and pattern recognition.



**Nishanth Sankareswaran** received his BE from PSG College of Technology in Instrumentation and Control systems engineering. His interests lie in image processing and machine vision. He is currently employed as an Imaging expert in Lensbricks Technologies Private Limited, Bangalore, India.