

## Research Article

# Accurate Fatigue Detection Based on Multiple Facial Morphological Features

Kangning Li , Shangshang Wang , Chang Du , Yuxin Huang, Xin Feng ,  
and Fengfeng Zhou 

*BioKnow Health Informatics Lab, College of Computer Science and Technology, and Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun, Jilin 130012, China*

Correspondence should be addressed to Xin Feng; 573175490@qq.com and Fengfeng Zhou; fengfengzhou@gmail.com

Received 20 September 2018; Accepted 30 December 2018; Published 28 February 2019

Academic Editor: Giorgio Pennazza

Copyright © 2019 Kangning Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Fatigue driving is becoming a dangerous and common situation for drivers and represents a significant factor for fatal car crashes. Machine learning researchers utilized various sources of information to detect driver's drowsiness. This study integrated the morphological features of both the eye and mouth regions and extensively investigated the fatigue detection problem from the aspects of feature numbers, classifiers, and modeling parameters. The proposed algorithm REcognizing the Drowsy Expression (REDE) achieved the 10-fold cross-validation accuracy 96.07% and took about 21 milliseconds to process one image. REDE outperformed the existing four studies on both fatigue detection accuracy and running time and is fast enough to handle the task of real-time fatigue monitoring captured at the rate of 30 frames per second. To further facilitate the research of fatigue detection, the raw data and the feature matrix were also released.

## 1. Introduction

The AAA Foundation released a report on traffic safety in 2015, and the data strongly suggested the extent and seriousness of fatigue driving [1]. During 2009-2013, 21% of fatal car crashes involved a drowsy driver, and more than two out of five drivers (43.2%) admitted the experiences of fatigue driving. However, there are only 17% of fatal car crashes involving a drowsy driver during 1999-2008. The increasing rate of fatigue driving is becoming more common and calls for better fatigue detection and warning system [2].

Machine learning-based fatigue detection algorithms may be roughly grouped into three categories. Firstly, fatigue-specific patterns in driver's physical behaviors, including eye behaviors [3-5] and yawning frequency [6], extracted from the captured video may be useful to detect fatigue. The integrated analysis of these multicue features may potentially improve the detection accuracies of the fatigue detection accuracy [7-10]. Secondly, fatigue-specific measurements may be extracted from a drowsy driver's physiological signals, e.g., heart rate variability [11] and EEG (electroencephalography)

[2, 12, 13]. Thirdly, the external monitoring of the vehicular behaviors may also facilitate driver's fatigue detection. For example, Morris et al. predicted driver's drowsiness by the car's lane deviation and vehicle heading variation [14]. And Krajewski et al. monitored the behaviors of the steering wheel to predict whether the driver is in fatigue [15].

This study hypothesized that the integration of multiple physical behaviors might achieve better fatigue detection performance. Many previous studies focused on a single type of driver's physical behaviors, e.g., eye activities or mouth yawning. This study extracted the Local Binary Pattern (LBP) features from the eye and mouth regions and summarized the principal components from LBP features for training the fatigue detection models. The proposed algorithm REcognizing the Drowsy Expression (REDE) outperformed the existing four fatigue detection studies on both accuracies and running times.

## 2. Material and Methods

*2.1. Problem Definition.* Fatigue detection is defined as a binary classification problem of determining whether a person

is in fatigue or not based on this person's face photo. The positive samples are the face photos of persons in fatigue and are denoted as  $P = \{P_1, P_2, P_3, \dots, P_n\}$ , while the face photos of persons without fatigue are the negative samples and are denoted as  $N = \{N_1, N_2, N_3, \dots, N_m\}$ . The total number of samples is  $s = n + m$ . Let  $X$  be the  $k$ -feature vector  $X = \langle X_1, X_2, \dots, X_k \rangle$  for a given sample, and the binary classification problem investigated in this study is to assign  $X$  as either a positive or a negative sample [16].

**2.2. Data Collection.** There is no public database of fatigue visual images and videos, so we built our video database captured from our recruited volunteers. Table 1 describes the details of the volunteered participants. Seven male and seven female participants were recruited and each of the fourteen volunteers signed the informed consent form. All of the participants do not have sleeping disorders that may affect neurocognitive ability and do not take food/drink/medicine that affects the neurocognitive system [17].

Two videos were captured for each participant at the statuses of nonfatigue and fatigue, respectively. A participant took a normal diet and a full rest for the first day and had one video recording at 8:00 AM of the next day in a rest state. After this time, the participant had no rest for 18 hours, and another video recording was taken at 3:00 AM of the third day in a rest state. The video was captured using the CMOS 5-megapixel camera in the MacBook Pro (13-inch screen), and the video resolution is  $1280 \times 720$  in pixels. The capturing frequency is 30 frames per second. Each video was recorded for 5 minutes and has 9000 ( $= 5 \times 60 \times 30$ ) images.

We randomly extracted 300 images from each video and labeled them as being fatigue or not by the majority voting from 100 volunteers. Two videos were recorded for each participant, so we have 8400 ( $= 300 \times 14 \times 2$ ) images in total. Let  $F_i$  be the number of votes for that the  $i^{\text{th}}$  image is fatigue and  $A_i$  be the number of votes against this statement. An image was removed from further analysis if  $|F_i - A_i| < 20$ . There were 1581 images removed for this rule. An image with a nonfatigue expression was defined as a positive sample, while an image with the fatigue expression was a negative sample. A person usually stays nonfatigue longer than the time of being fatigue. So we randomly chose 80 positive images and 40 negative images for each participant. The final dataset has 1680 samples with 1120 positive and 560 negative samples, respectively.

The raw images of the eyes and mouths were released at <http://www.healthinformatics.org/supp/resources.php>. The data matrix of grayscale values and extracted LBP (Local Binary Pattern) features were also released as a benchmark dataset at this website.

**2.3. Data Preprocessing.** This study hypothesized that the morphological patterns in the regions of the eyes and mouth may well represent the fatigue status. So we detected and extracted the regions of two eyes and one mouth from each image by Dlib version 19.4.99 [18] described by Kazemi and Sullivan [19]. The detection accuracy of a random subset was inspected manually and illustrated in Figure 1.

TABLE 1: Basic information of the participants. Each participant has the information on gender, age, and whether a pair of glasses was worn.

ID	Gender	Age	Glasses	ID	Gender	Age	Glasses
1	Female	20	No	8	Male	49	No
2	Female	20	No	9	Male	49	No
3	Female	20	No	10	Male	20	No
4	Female	19	No	11	Male	21	No
5	Female	25	No	12	Male	20	No
6	Female	49	No	13	Male	20	No
7	Female	20	Yes	14	Male	20	Yes

Each image was preprocessed by the following steps, as illustrated in Figure 2. An image was first transformed into the grayscale colors by the formula  $\text{GrayScale} = 0.299 \times R + 0.587 \times G + 0.114 \times B$ , where  $R/G/B$  were the pixel values of the red/green/blue channels [8]. This conversion was provided by the function `imread()` from the library OpenCV. A gamma correction  $1/2.2$  was usually applied to normalize the light variations for human face detection, as recommended in [20]. Let  $\text{MaxV}$  be the maximal grayscale pixel value of the image matrix  $M$  after the gamma correction. The final image matrix was calculated as  $M' = M / (\text{MaxV} \times 255)$ . After being detected using Dlib, the image sizes of the face, eye, and mouth were scaled as  $320 \times 320$ ,  $64 \times 32$ , and  $64 \times 64$  in pixels, respectively.

**2.4. Extracting Features from the Images.** This study extracted the Local Binary Patterns (LBP) from the eye and mouth images and then calculated the eigenvector of the LBP feature vector, as illustrated in Figure 2.

The parameter `pCellSize` of the LBP algorithm was set to  $32 \times 32$  by default. So each sample has 512 features for each eye image and 1024 features for the mouth image. There are 2048 features in total for each sample with two eye images and one mouth image.

**2.5. Experimental Environments.** This study integrated the morphological features of both eyes and mouth to REcognizing the Drowsy Expression (REDE). A binary classification model was trained over the principal components calculated from the LBP features of the facial images.

The experiments were implemented using Python and C++ on an Intel Core-I5 2.70 GHz CPU and 8 GB memory. The machine learning experiment was conducted based on the Python scikit-learn version 0.18.2 [21]. The processing of video and face recognition was implemented using the software Photo Booth version 9.0.0 [22]. The whole procedure of extracting features and predicting the binary classification result for an image took approximately 0.02 seconds. So the pipeline may be applied to do the real-time monitoring of videos captured at 30 frames per second in this study.

**2.6. Classification Performance Measurements.** Sensitivity (Sn), specificity (Sp), accuracy (Acc), and Matthews' correlation coefficient (MCC) were widely used to evaluate how well

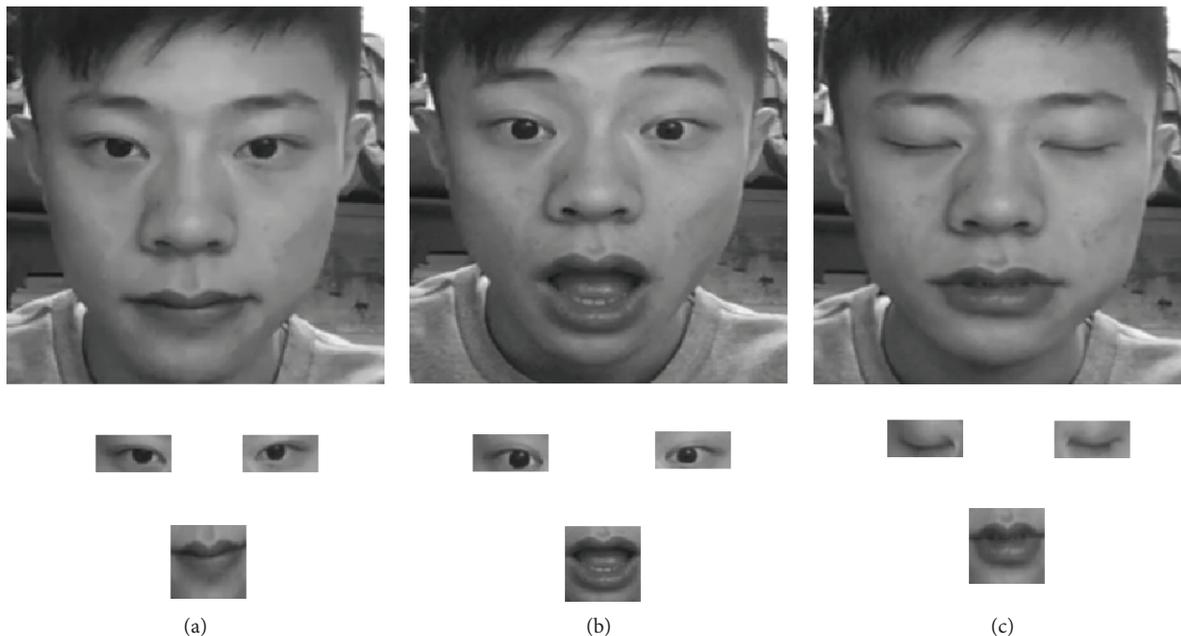


FIGURE 1: Detection of the eyes and mouth in a facial image. (a, b) The nonfatigue images 1 and 2 and (c) the fatigue image for participant 10.

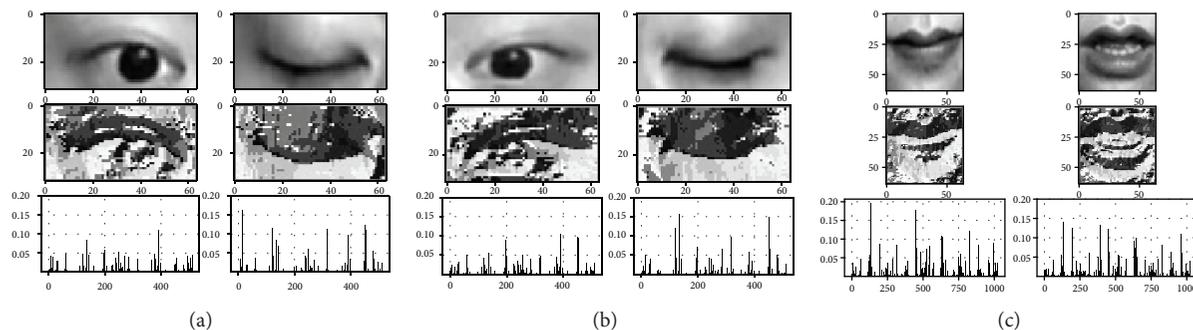


FIGURE 2: Original images, abstracted images using eigenvalues, and histograms of the nonfatigue and fatigue images. (a) The left eye, (b) the right eye, and (c) the mouth of the 10<sup>th</sup> participant.

TABLE 2: Baseline information of the dataset. This table gives the numbers of the positive (nonfatigue) and negative (fatigue) images.

		Positive	Negative
Male	Glasses	80	40
	Nonglasses	480	240
Female	Glasses	80	40
	Nonglasses	480	240
Total		1120	560

a binary classification model performs [16, 23, 24]. This study chose these four performance measurements to demonstrate the performance of fatigue detection algorithms. Let TP and FN be the numbers of positive samples that are predicted as positives and negatives, respectively. The measurements TN and FP are defined as the numbers of negative samples that are predicted as negatives and positives, respectively. So the number of positive samples is  $P = TP + FN$ . And the number of negative samples is  $N =$

$TN + FP$ . The performance measurements Sn and Sp are the percentages of correctly predicted positive and negative samples, i.e.,  $S_n = TP / (TP + FN) = TP / P$  and  $S_p = TN / (TN + FP) = TN / N$ . The measurement accuracy is defined as the overall percentage of correctly predicted samples, i.e.,  $Acc = (TP + TN) / (P + N)$  [25]. Matthews' correlation coefficient (MCC) ranges between -1 and 1 and is defined as  $MCC = (TP \times TN - FP \times FN) / \sqrt{((TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN))}$  [26–28]. The function  $\sqrt{X}$  returns the square root of X.

All the performance measurements mentioned above were calculated using the 10-fold cross-validations [29].

### 3. Results and Discussion

**3.1. Baseline Information of the Dataset.** After the video capturing and image preprocessing, we retrieved 1120 positive images and 560 negative images, as shown in Table 2. One male and one female participants wear glasses. So 1/7 facial images were included for further analysis proportionally.

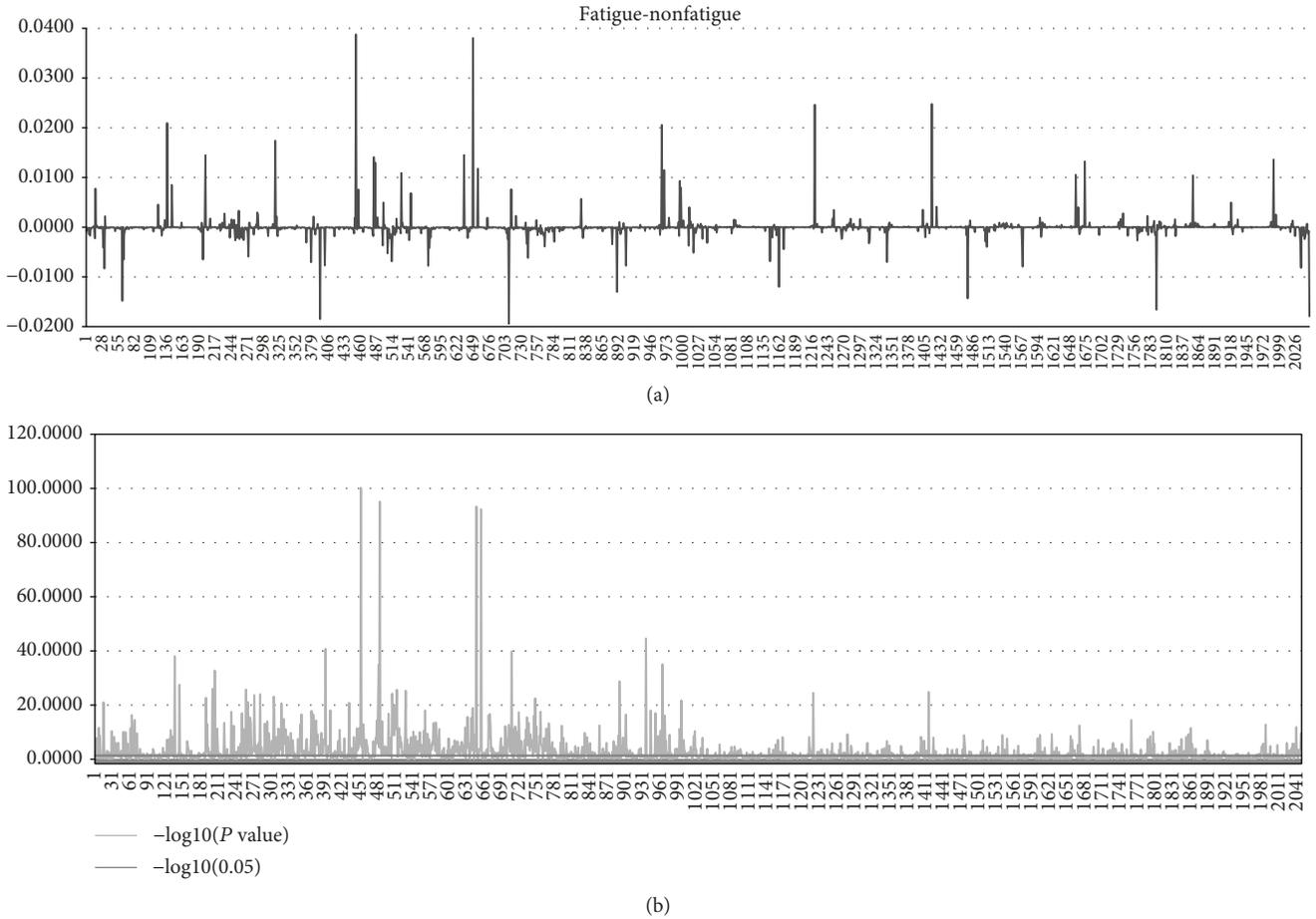


FIGURE 3: The differences of LBP features between the two groups of samples. (a) For each feature, the averaged value of the fatigue samples minus that of the nonfatigue samples was plotted. (b) The  $t$ -test  $P$  value was calculated between the two groups of samples for each feature, and the value  $-\log_{10}(P \text{ value})$  was plotted.

### 3.2. Not All the Features Contribute to Fatigue Detection.

Firstly, we investigated the differences of the 2048 LBP features between the fatigue and nonfatigue samples, as shown in Figure 3. The maximal difference was 0.0387 for the 452<sup>nd</sup> feature. Moreover, there were only 26 features with the differences at least 0.01. The  $t$ -test was used to evaluate the discriminative power of each LBP feature, as shown in Figure 3(b) [30]. 512 LBP features had significant discriminative powers of fatigue detection, i.e.,  $t$ -test  $P$  value  $\leq 0.05$ . So the majority of the LBP features did not have discriminative powers for detecting fatigue samples. The  $P$  value was calculated by the  $t$ -test function in the Python package `scipy` version 1.1.0 [31].

So we calculated the principal components of all the 2048 LBP features and evaluated the top 20 principal components for the fatigue detection problem, as shown in Figure 4. The principal components were calculated on the data mixture of both fatigue and nonfatigue samples, but they already demonstrated significant differences between these two groups of samples, as shown in Figure 4(a). The discriminative power of these principal components was evaluated by the  $t$ -test  $P$  value. Figure 4(b) demonstrates that 11 out of the top 20 principal components had a  $P$  value  $\leq 0.05$ . The

9<sup>th</sup> principal component achieved the best discrimination by  $P$  value =  $5.2805 \times 10^{-35}$ . The principal components were calculated by the PCA function in the Python package `scikit-learn` version 0.18.2 [31].

This study evaluated three image feature extraction algorithms, i.e., LBP, HOG [32], and LeNet [33], for their classification performances on the fatigue detection problem. The performances were calculated by the 10-fold cross-validation of the classifier decision tree (`scikit-learn` version 0.18.2). Figure 5 demonstrates that LBP achieved the best classification performances among these three algorithms. HOG performed slightly worse than LBP, and LeNet features achieved 0.0458 worse in Acc than LBP. So the following sections used the LBP-extracted features to build the fatigue detection model.

**3.3. Optimizing the Features.** We evaluated different values for the parameter cell size (`pCellSize`) of the image feature extraction algorithm LBP and the parameter component number (`pComponentNum`) of the principal component analysis (PCA). The 10-fold cross-validation accuracy of the Support Vector Machine (SVM) classifier with the radial basis function (RBF) kernel was calculated to evaluate a given

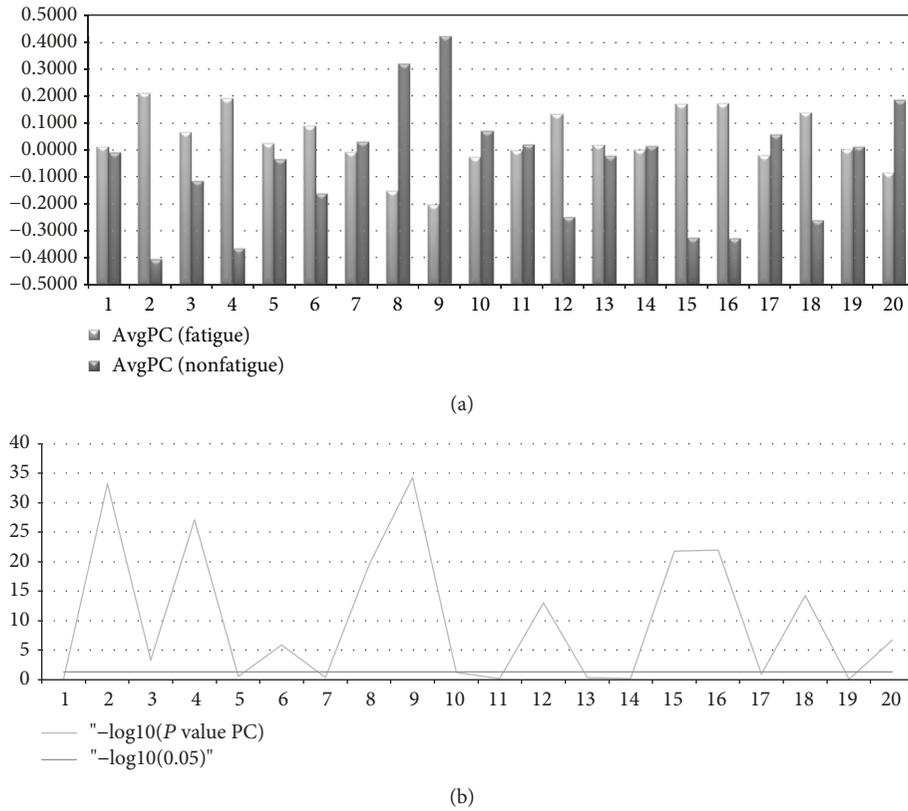


FIGURE 4: Top 20 principal components of the LBP features in the two groups of samples. (a) The averaged values of the top 20 principal components and (b) the  $t$ -test  $P$  value between the two groups of samples were plotted.

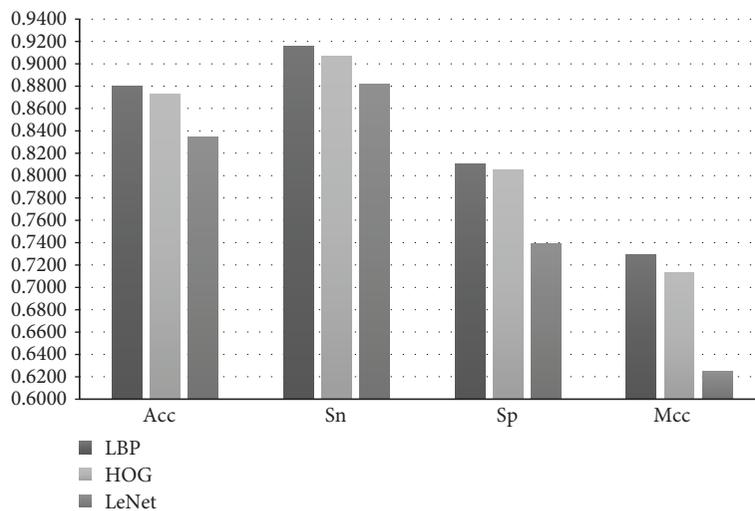


FIGURE 5: Classification performances of different image feature extraction algorithms. Three image feature extraction algorithms LBP, HOG, and LeNet were compared.

feature subset. The experiment was implemented by the Python scikit-learn version 0.18.2, and all the other parameters of these algorithms were used with their default values.

The best prediction accuracy 90.60% was achieved with  $pCellSize = 32 \times 32$ , as shown in Figure 6. Overall, the classifier SVM-RBF performed very well on the fatigue detection problem and achieved the accuracies ranging between 87.86% and 90.60%. If Width was fixed as 16

or 32, SVM-RBF performed the best when Height = 32. If Width = 8, SVM-RBF performed slightly better (1.01%) on Height = 32 than Height = 64. If Height was fixed as 16 or 32, Width = 32 was the best choice for the classifier SVM-RBF, while the SVM classifier performed slightly better (0.12%) in accuracy on Width = 16 than Width = 32, if Height was 8. So the parameter  $pCellSize = 32 \times 32$  was chosen as the default value for further analysis.

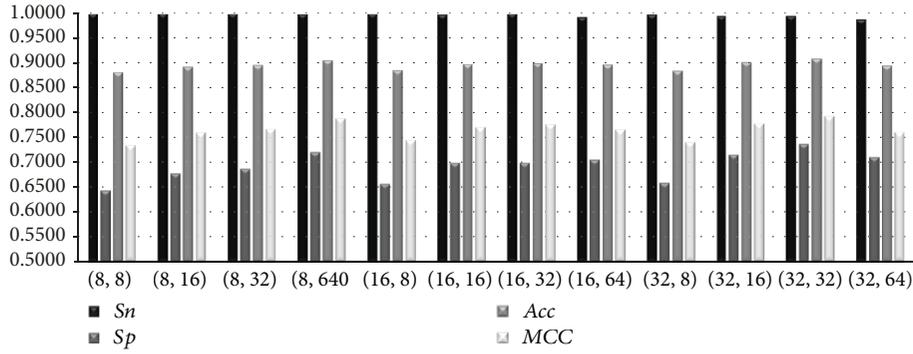


FIGURE 6: Prediction accuracies of the fatigue detection problem by different values of the parameter pCellSize. A cell of the LBP algorithm has two values, i.e., Width and Height, and pCellSize is represented as Width × Height.

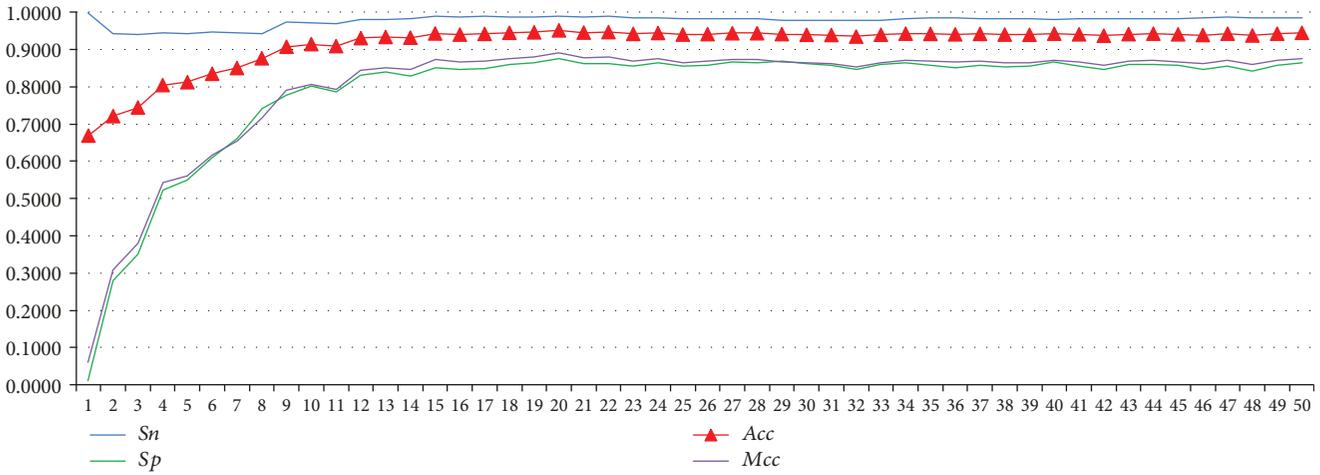


FIGURE 7: SVM-RBF performances on the fatigue detection problem by different values of the parameter pComponentNum. The parameter pComponentNum is the number of principal components used in the binary classification model.

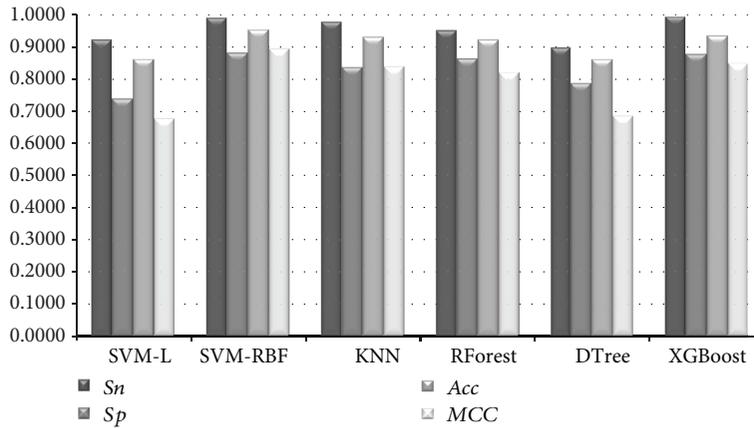


FIGURE 8: Performance measurements of the six classifiers on the fatigue detection dataset. The classification performance was measured by Sn, Sp, Acc, and MCC.

The classifier SVM-RBF reached the best accuracy 95.18% at the value pComponentNum = 20, as shown in Figure 7. A smaller or larger value of pComponentNum reduced the fatigue prediction accuracy, and the classifier SVM-RBF performed stably well after the parameter pComponentNum is greater than 11, with the maximal decrease

0.65% in accuracy when pComponentNum = 50. Experiments in the following sections were conducted using pComponentNum = 20.

3.4. Choosing a Good Classifier. Six binary classification algorithms were compared for their prediction performances of

Sn		C																								
		0.125	0.250	0.375	0.500	0.625	0.750	0.875	1.000	1.125	1.250	1.375	1.500	1.625	1.750	1.875	2.000	2.125	2.250	2.375	2.500	2.625	2.750	2.875	3.000	
Gamma	0.100	0.9946	0.9911	0.9902	0.9920	0.9911	0.9902	0.9911	0.9911	0.9911	0.9902	0.9902	0.9884	0.9884	0.9875	0.9875	0.9875	0.9866	0.9875	0.9866	0.9866	0.9866	0.9866	0.9857	0.9848	
	0.178	0.9991	0.9929	0.9929	0.9911	0.9902	0.9911	0.9902	0.9911	0.9902	0.9902	0.9902	0.9875	0.9875	0.9875	0.9848	0.9848	0.9839	0.9821	0.9804	0.9804	0.9795	0.9786	0.9786	0.9777	0.9768
	0.316	1.0000	0.9982	0.9973	0.9946	0.9938	0.9938	0.9920	0.9893	0.9857	0.9830	0.9830	0.9821	0.9821	0.9804	0.9795	0.9795	0.9804	0.9804	0.9795	0.9795	0.9795	0.9786	0.9786	0.9777	0.9768
	0.562	1.0000	1.0000	0.9982	0.9982	0.9964	0.9964	0.9946	0.9946	0.9938	0.9929	0.9911	0.9911	0.9902	0.9902	0.9893	0.9893	0.9884	0.9884	0.9884	0.9884	0.9884	0.9884	0.9884	0.9884	0.9884
	1.000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9991	0.9982	0.9973	0.9973	0.9973	0.9964	0.9964	0.9964	0.9964	0.9964	0.9964	0.9964	0.9964	0.9964	0.9964	0.9964	0.9964	0.9964	0.9964	0.9964
	1.334	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9991	0.9982	0.9982	0.9982	0.9982	0.9982	0.9982	0.9982	0.9982	0.9982	0.9982	0.9982	0.9982	0.9982	0.9982	0.9982	0.9982	0.9982	0.9982
	1.778	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9991	0.9991	0.9991	0.9991	0.9991	0.9991	0.9991	0.9991	0.9991	0.9991	0.9991	0.9991	0.9991	0.9991	0.9991	0.9991

Sp		C																								
		0.125	0.250	0.375	0.500	0.625	0.750	0.875	1.000	1.125	1.250	1.375	1.500	1.625	1.750	1.875	2.000	2.125	2.250	2.375	2.500	2.625	2.750	2.875	3.000	
Gamma	0.100	0.632	0.766	0.818	0.852	0.864	0.877	0.882	0.889	0.891	0.896	0.896	0.895	0.895	0.896	0.898	0.900	0.907	0.907	0.904	0.905	0.904	0.904	0.904	0.904	
	0.178	0.4196	0.6875	0.7750	0.8161	0.8446	0.8554	0.8607	0.8732	0.8750	0.8804	0.8839	0.8875	0.8893	0.8911	0.8911	0.8929	0.8929	0.8911	0.8911	0.8911	0.8911	0.8911	0.8929	0.8946	0.8982
	0.316	0.1339	0.4518	0.6036	0.6875	0.7571	0.7804	0.8000	0.8179	0.8286	0.8393	0.8446	0.8500	0.8518	0.8500	0.8518	0.8518	0.8518	0.8536	0.8536	0.8536	0.8536	0.8554	0.8536	0.8554	0.8554
	0.562	0.0000	0.1643	0.3250	0.4500	0.5393	0.5911	0.6357	0.6750	0.6964	0.7089	0.7143	0.7143	0.7143	0.7143	0.7161	0.7161	0.7179	0.7196	0.7196	0.7196	0.7196	0.7196	0.7196	0.7196	0.7196
	1.000	0.0000	0.0000	0.0500	0.1357	0.2500	0.3196	0.3804	0.4179	0.4518	0.4768	0.4821	0.4821	0.4821	0.4821	0.4821	0.4821	0.4821	0.4821	0.4821	0.4821	0.4821	0.4821	0.4821	0.4821	0.4821
	1.334	0.0000	0.0000	0.0036	0.0321	0.0946	0.1732	0.2214	0.2786	0.3268	0.3554	0.3554	0.3554	0.3554	0.3554	0.3554	0.3554	0.3554	0.3554	0.3554	0.3554	0.3554	0.3554	0.3554	0.3554	0.3554
	1.778	0.000	0.000	0.000	0.009	0.039	0.063	0.086	0.129	0.171	0.200	0.204	0.204	0.204	0.204	0.204	0.204	0.204	0.204	0.204	0.204	0.204	0.204	0.204	0.204	0.204

Acc		C																								
		0.125	0.250	0.375	0.500	0.625	0.750	0.875	1.000	1.125	1.250	1.375	1.500	1.625	1.750	1.875	2.000	2.125	2.250	2.375	2.500	2.625	2.750	2.875	3.000	
Gamma	0.100	0.874	0.916	0.933	0.945	0.949	0.952	0.955	0.957	0.958	0.958	0.958	0.957	0.957	0.957	0.958	0.958	0.961	0.961	0.959	0.960	0.959	0.959	0.958	0.958	
	0.178	0.8060	0.8911	0.9202	0.9327	0.9417	0.9458	0.9470	0.9518	0.9518	0.9536	0.9548	0.9542	0.9548	0.9536	0.9536	0.9536	0.9524	0.9506	0.9506	0.9500	0.9494	0.9500	0.9500	0.9506	
	0.316	0.7113	0.8161	0.8661	0.8923	0.9149	0.9226	0.9280	0.9321	0.9333	0.9351	0.9369	0.9381	0.9387	0.9369	0.9369	0.9369	0.9375	0.9381	0.9375	0.9381	0.9375	0.9381	0.9375	0.9369	
	0.562	0.6667	0.7214	0.7738	0.8155	0.8452	0.8613	0.8750	0.8881	0.8946	0.8982	0.8988	0.8988	0.8982	0.8982	0.8982	0.8982	0.8982	0.8988	0.8988	0.8988	0.8988	0.8988	0.8988	0.8988	0.8988
	1.000	0.6667	0.6667	0.6833	0.7119	0.7500	0.7726	0.7923	0.8042	0.8155	0.8282	0.8250	0.8250	0.8250	0.8250	0.8250	0.8250	0.8250	0.8250	0.8250	0.8250	0.8250	0.8250	0.8250	0.8250	0.8250
	1.334	0.6667	0.6667	0.6679	0.6774	0.6982	0.7244	0.7405	0.7589	0.7744	0.7839	0.7839	0.7839	0.7839	0.7839	0.7839	0.7839	0.7839	0.7839	0.7839	0.7839	0.7839	0.7839	0.7839	0.7839	0.7839
	1.778	0.667	0.667	0.667	0.670	0.680	0.688	0.695	0.710	0.724	0.733	0.734	0.734	0.734	0.734	0.734	0.734	0.734	0.734	0.734	0.734	0.734	0.734	0.734	0.734	0.734

MCC		C																							
		0.125	0.250	0.375	0.500	0.625	0.750	0.875	1.000	1.125	1.250	1.375	1.500	1.625	1.750	1.875	2.000	2.125	2.250	2.375	2.500	2.625	2.750	2.875	3.000
Gamma	0.100	0.720	0.812	0.849	0.877	0.885	0.893	0.898	0.904	0.905	0.905	0.905	0.902	0.902	0.903	0.905	0.905	0.911	0.911	0.907	0.909	0.907	0.907	0.906	0.904
	0.178	0.5681	0.7576	0.8221	0.8493	0.8690	0.8784	0.8809	0.8916	0.8915	0.8955	0.8982	0.8966	0.8979	0.8951	0.8951	0.8950	0.8922	0.8881	0.8881	0.8867	0.8854	0.8867	0.8867	0.8881
	0.316	0.3057	0.5912	0.7043	0.7610	0.8105	0.8276	0.8390	0.8476	0.8497	0.8534	0.8574	0.8600	0.8613	0.8571	0.8571	0.8571	0.8585	0.8598	0.8584	0.8584	0.8597	0.8570	0.8583	0.8569
	0.562	0.0000	0.3404	0.4879	0.5898	0.6582	0.6932	0.7227	0.7518	0.7659	0.7734	0.7741	0.7741	0.7725	0.7725	0.7722	0.7722	0.7720	0.7733	0.7733	0.7733	0.7733	0.7733	0.7733	0.7733
	1.000	0.0000	0.0000	0.1841	0.3078	0.4264	0.4858	0.5342	0.5623	0.5891	0.6085	0.6106	0.6106	0.6106	0.6106	0.6106	0.6106	0.6106	0.6106	0.6106	0.6106	0.6106	0.6106	0.6106	0.6106
	1.334	0.0000	0.0000	0.0488	0.1472	0.2552	0.3501	0.3992	0.4497	0.4894	0.5136	0.5136	0.5136	0.5136	0.5136	0.5136	0.5136	0.5136	0.5136	0.5136	0.5136	0.5136	0.5136	0.5136	0.5136
	1.778	0.000	0.000	0.000	0.077	0.163	0.206	0.243	0.299	0.348	0.375	0.378	0.378	0.378	0.378	0.378	0.378	0.378	0.378	0.378	0.378	0.378	0.378	0.378	0.378

FIGURE 9: Optimizing the parameters C and Gamma of SVM-RBF. Multiple values of the two parameters C and Gamma were evaluated for the prediction accuracy of the algorithm SVM-RBF. The classification performance measurements were calculated by the 10-fold cross-validation. The four heatmaps illustrated the data for the four performance measurements Sn, Sp, Acc, and MCC. The maximal, mean, and minimal values of each measurement were colored as red, yellow, and blue, respectively. The values in between were illustrated in gradient colors.

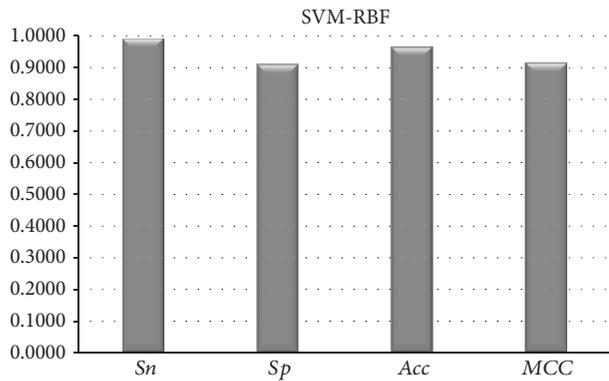


FIGURE 10: Performance of the best model for detecting fatigue in this study. This model was measured by the performance metrics Sn, Sp, Acc, and MCC.

the fatigue detection problem, as shown in Figure 8. The Support Vector Machines with both Linear and RBF kernels were denoted as SVM-L and SVM-RBF. The  $k$ -nearest-neighbor (KNN) was also used to evaluate how a simple distance-based classifier performed on the fatigue detection problem. A decision tree (DTree) classifier has an inherent nature of easy interpretability, and random forest (RForest) classifier tends to have a better classification performance.

The extreme gradient boost (XGBoost) algorithm was a recently developed meta-algorithm and outperformed the existing classification algorithms in many cases [34]. XGBoost was implemented in the Python package XGBoost version 0.71. All the other classifiers were provided in the Python package scikit-learn. The features were extracted from the images using the parameters  $pCellSize = 32 \times 32$  and  $pComponentNum = 20$ .

Figure 8 demonstrates that the classifier SVM-RBF performed the best on three of the four performance measurements of the fatigue detection problem. Although SVM-L shares the same framework of support vectors with SVM-RBF, SVM-L performed much worse than SVM-RBF. XGBoost performed similarly well with SVM-RBF, with a slight decrease (4.39%) in accuracy. XGBoost achieved the best sensitivity  $Sn = 99.11\%$ , but its specificity  $Sp$  was worse than SVM-RBF. KNN generated the third best fatigue detection model, but the two tree-based classifiers RForest and DTree did not achieve satisfying classification accuracies. So SVM-RBF was chosen as the classifier for the final model.

3.5. Optimizing the Parameters of the Classification Module. We further refined the best classifier SVM-RBF by screening for the best values of the two parameters C and Gamma, as shown in Figure 9. Twenty values between 0.125 and 3.000

TABLE 3: Four recent studies on the fatigue detection problem. The column “Year” gives the publication year of each study. And the column “Algorithms” summarizes the techniques used in each study. The column “Accuracy” gives the accuracy achieved by the best model on the complete or largest test dataset in each study. The column “Time (ms)” is the processing time of the algorithm on one image.

Authors	Year	Algorithms	Accuracy	Time (ms)
Ma et al.	2017	Two-stream CNN	91.57%	—
Wathiq and Ambudkar	2017	VJ method, SVM	94.32%	70
Wang et al.	2017	Iris tracking, EBF, PERCLOS	80.75%	—
Gao and Wang	2017	Eye feature extracting, SVM	88%	—

with step size 0.125 were evaluated for the parameter  $C$ . Seven values {0.100, 0.178, 0.316, 0.562, 1.00, 1.334, and 1.778} were evaluated for the other parameter Gamma. A grid search was conducted to find the best choices of these two parameters. The classifier SVM-RBF achieved the best accuracy of 96.07% when  $C = 2.125$  and Gamma = 0.100.

**3.6. Summarization of the Best Model.** The best model to detect fatigue using the facial images was achieved after the above optimization procedure, as shown in Figure 10. The classifier SMV-RBF with the parameter  $C = 2.125$  and Gamma = 0.1 achieved a satisfying performance on our dataset. The sensitivity  $S_n = 98.75\%$  and overall accuracy  $Acc = 96.07\%$  were very high, while the specificity  $S_p = 90.71\%$  might need further improvements.

**3.7. Comparison with the Existing Studies.** Four recent studies were chosen for a comparison with our work, as shown in Table 3. Ma et al. refined the background removal of the two-stream video and trained a depth video-based convolutional neural network (CNN) model [35]. They achieved an overall fatigue detection accuracy of 91.57%. Wathiq and Ambudkar proposed a fatigue detection model based on the texture features of facial components and achieved the accuracy 94.32% on the largest dataset [36]. Wang et al. focused on the patterns of the eye activities and implemented an Android-based fatigue detection system with an accuracy of 80.75% [37]. Gao and Wang described the fatigue status by the percentage of eye closure time, the average eye closure time, and the eye blinking frequency and achieved the fatigue detection accuracy 88% [38].

The algorithm REDE in this study achieved the best accuracy 96.07% using the SVM-RBF algorithm and took 21 milliseconds to process one image. So REDE outperformed all the existing four studies on the fatigue detection accuracy. Only the Wathiq and Ambudkar study discussed the processing time of one facial image and ran about 3.33 times slower than REDE proposed in this study. The calculation of REDE was also fast enough to support the real-time fatigue monitoring since the video capturing technology used in this study generated 30 images per second.

## 4. Conclusions and Future Directions

The facial image-based fatigue detection system REDE consists of three major steps. Firstly, the regions of two eyes and one mouth were extracted from the facial image. Secondly, the LBP algorithm was used to extract features from

the three regions, and the principal components were calculated using the PCA algorithm. Lastly, the SVM model with the RBF kernel was trained to do the classification. REDE outperformed the four existing fatigue detection algorithms and ran fast enough to do the real-time fatigue monitoring.

We will focus on improving the first two steps of REDE in the future. To facilitate the fatigue detection community, we also released the images for feature extractions in this study at <http://www.healthinformatics.org/supp/resources.php>. The readers and fatigue detection researchers may also directly use our LBP feature matrix extracted from the images for classification model refining.

## Data Availability

To further facilitate the research of fatigue detection, the raw data and the feature matrix were also released at <http://www.healthinformatics.org/supp/resources.php>.

## Conflicts of Interest

The authors declare that there are no conflicts of interest.

## Authors' Contributions

FZ, XF, and KL conceived the project, designed the experiments, and drafted the manuscript. KL, SW, CD, and YH collected the data and carried out the experiments. FZ, KL, and XF proofed and polished the manuscript and organized this project.

## Acknowledgments

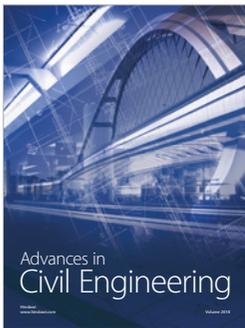
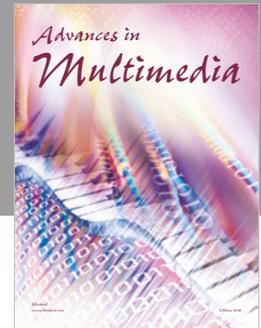
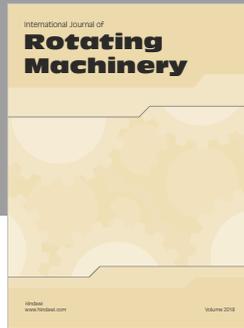
This work was supported by the Strategic Priority Research Program of the Chinese Academy of Sciences (XDB13040400), Jilin Provincial Key Laboratory of Big Data Intelligent Computing (20180622002JC), the Education Department of Jilin Province (JJKH20180145KJ), and the startup grant of the Jilin University. This work was also partially supported by the Bioknow MedAI Institute (BMCPP-2018-001), and the High Performance Computing Center of Jilin University, China. We appreciate the constructive comments from the anonymous reviewers.

## References

- [1] L. S. Arnold and B. C. Tefft, *Prevalence of Self-Reported Drowsy Driving*, United States: 2015, 2015.

- [2] R. Fu, H. Wang, and W. Zhao, "Dynamic driver fatigue detection using hidden Markov model in real driving condition," *Expert Systems with Applications*, vol. 63, pp. 397–411, 2016.
- [3] C. Sun, J. H. Li, Y. Song, and L. Jin, "Real-time driver fatigue detection based on eye state recognition," *Applied Mechanics and Materials*, vol. 457–458, pp. 944–952, 2013.
- [4] J. Jo, S. J. Lee, K. R. Park, I.-J. Kim, and J. Kim, "Detecting driver drowsiness using feature-level fusion and user-specific classification," *Expert Systems with Applications*, vol. 41, no. 4, pp. 1139–1152, 2014.
- [5] W. Kong, L. Zhou, Y. Wang, J. Zhang, J. Liu, and S. Gao, "A system of driving fatigue detection based on machine vision and its application on smart device," *Journal of Sensors*, vol. 2015, Article ID 548602, 11 pages, 2015.
- [6] G. M. Bhandari, "Yawning analysis for driver drowsiness detection," *International Journal of Research in Engineering and Technology*, vol. 3, no. 2, pp. 502–505, 2014.
- [7] X. Lan, A. J. Ma, and P. C. Yuen, "Multi-cue visual tracking using robust feature-level fusion based on joint sparse representation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1194–1201, Columbus, OH, USA, June 2014.
- [8] X. Lan, A. J. Ma, P. C. Yuen, and R. Chellappa, "Joint sparse representation and robust feature-level fusion for multi-cue visual tracking," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5826–5841, 2015.
- [9] X. Lan, M. Ye, S. Zhang, H. Zhou, and P. C. Yuen, "Modality-correlation-aware sparse representation for RGB-infrared object tracking," *Pattern Recognition Letters*, 2018, In press.
- [10] X. Lan, S. Zhang, P. C. Yuen, and R. Chellappa, "Learning common and feature-specific patterns: a novel multiple-sparse-representation-based tracker," *IEEE Transactions on Image Processing*, vol. 27, no. 4, pp. 2022–2037, 2018.
- [11] J. Vicente, P. Laguna, A. Bartra, and R. Bailón, "Drowsiness detection using heart rate variability," *Medical & Biological Engineering & Computing*, vol. 54, no. 6, pp. 927–937, 2016.
- [12] F. C. Lin, L. W. Ko, C. H. Chuang, T. P. Su, and C. T. Lin, "Generalized EEG-based drowsiness prediction system by using a self-organizing neural fuzzy system," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 59, no. 9, pp. 2044–2055, 2012.
- [13] R. N. Khushaba, S. Kodagoda, S. Lal, and G. Dissanayake, "Driver drowsiness classification using fuzzy wavelet-packet-based feature-extraction algorithm," *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 1, pp. 121–131, 2011.
- [14] D. M. Morris, J. J. Pilcher, and F. S. Switzer III, "Lane heading difference: an innovative model for drowsy driving detection using retrospective analysis around curves," *Accident Analysis & Prevention*, vol. 80, pp. 117–124, 2015.
- [15] J. Krajewski, D. Sommer, U. Trutschel, D. Edwards, and M. Golz, "Steering wheel behavior based estimation of fatigue," in *Proceedings of the 5th International Driving Symposium on Human Factors in Driver Assessment, Training, and Vehicle Design: Driving Assessment 2009*, Big Sky, MT, USA, 2009.
- [16] R. Ge, M. Zhou, Y. Luo et al., "McTwo: a two-step feature selection algorithm based on maximal information coefficient," *BMC Bioinformatics*, vol. 17, no. 1, p. 142, 2016.
- [17] V. E. Wilkinson, M. L. Jackson, J. Westlake et al., "The accuracy of eyelid movement parameters for drowsiness detection," *Journal of Clinical Sleep Medicine*, vol. 9, no. 12, pp. 1315–1324, 2013.
- [18] D. E. King, "Dlib-ml: a machine learning toolkit," *Journal of Machine Learning Research*, vol. 10, pp. 1755–1758, 2009.
- [19] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1867–1874, Columbus, OH, USA, 2014.
- [20] H. Farid, "Blind inverse gamma correction," *IEEE Transactions on Image Processing*, vol. 10, no. 10, pp. 1428–1433, 2001.
- [21] F. Pedregosa, G. Varoquaux, A. Gramfort et al., "Scikit-learn: machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [22] M. Van Schalkwyk, *System for Capturing and Displaying Digital Images*, Google Patents, 2010.
- [23] P. Guo, Y. Luo, G. Mai et al., "Gene expression profile based classification models of psoriasis," *Genomics*, vol. 103, no. 1, pp. 48–55, 2014.
- [24] C. Xu, J. Liu, W. Yang et al., "AnOMIC biomarker detection algorithm TriVote and its application in methylomic biomarker detection," *Epigenomics*, vol. 10, no. 4, pp. 335–347, 2018.
- [25] C.-C. Yu, J.-C. Yang, Y.-C. Chang et al., "VCP phosphorylation-dependent interaction partners prevent apoptosis in Helicobacter pylori-infected gastric epithelial cells," *PLoS One*, vol. 8, no. 1, article e55724, 2013.
- [26] Z. Zhang, Z. Wang, G. Mai, Y. Luo, M. Zhao, and F. Zhou, "Evolutionary optimization of transcription factor binding motif detection," *Advances in Experimental Medicine and Biology*, vol. 827, pp. 261–274, 2015.
- [27] F. F. Zhou, Y. Xue, G. L. Chen, and X. Yao, "GPS: a novel group-based phosphorylation predicting and scoring method," *Biochemical and Biophysical Research Communications*, vol. 325, no. 4, pp. 1443–1448, 2004.
- [28] B. W. Matthews, "Comparison of the predicted and observed secondary structure of T4 phage lysozyme," *Biochimica et Biophysica Acta (BBA) - Protein Structure*, vol. 405, no. 2, pp. 442–451, 1975.
- [29] R. Zhao, R. Zhang, T. Tang et al., "TriZ-a rotation-tolerant image feature and its application in endoscope-based disease diagnosis," *Computers in Biology and Medicine*, vol. 99, pp. 182–190, 2018.
- [30] K. F. Phillips, "Power of the two one-sided tests procedure in bioequivalence," *Journal of Pharmacokinetics and Biopharmaceutics*, vol. 18, no. 2, pp. 137–144, 1990.
- [31] T. C. Pataky, "One-dimensional statistical parametric mapping in Python," *Computer Methods in Biomechanics and Biomedical Engineering*, vol. 15, no. 3, pp. 295–301, 2012.
- [32] X. Cai, F. Nie, H. Huang, and F. Kamangar, "Heterogeneous image feature integration via multi-modal spectral clustering," in *CVPR 2011*, Colorado Springs, CO, USA, USA, 2011.
- [33] Y. Jia, E. Shelhamer, J. Donahue et al., "Caffe: convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia.*, pp. 675–678, Orlando, FL, USA, November 2014.
- [34] T. Chen and C. Guestrin, "Xgboost: a scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794, San Francisco, CA, USA, 2016.
- [35] X. Ma, L.-P. Chau, and K.-H. Yap, "Depth video-based two-stream convolutional neural networks for driver fatigue detection," in *2017 International Conference on Orange*

- Technologies (ICOT)*, pp. 155–158, Singapore, Singapore, December 2017.
- [36] O. Wathiq and B. D. Ambudkar, “Optimized driver safety through driver fatigue detection methods,” in *2017 International Conference on Trends in Electronics and Informatics (ICEI)*, pp. 68–73, Tirunelveli, India, May 2017.
- [37] F. Wang, X. Chen, D. Wang, and B. Yang, “An improved image-based iris-tracking for driver fatigue detection system,” in *2017 36th Chinese Control Conference (CCC)*, pp. 11521–11526, Dalian, China, July 2017.
- [38] Y. Gao and C. Wang, “Fatigue state detection from multi-feature of eyes,” in *2017 4th International Conference on Systems and Informatics (ICSAI)*, pp. 177–181, Hangzhou, China, November 2017.



**Hindawi**

Submit your manuscripts at  
[www.hindawi.com](http://www.hindawi.com)

