Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2016) 24: 5124 – 5134

doi:10.3906/elk-1409-10

A fast and accurate algorithm for eye opening or closing detection based on local maximum vertical derivative pattern

Marzieh TAFRESHI, Ali Mohammad FOTOUHI*
Department of Electrical Engineering, Tafresh University, Tafresh 39518 79611, Iran

Received: 01.09.2014 • **Accepted/Published Online:** 29.10.2015 • **Final Version:** 06.12.2016

Abstract: In this paper, a fast and accurate algorithm is proposed to recognize open and closed eye states. In the proposed algorithm, first a hierarchical preprocessing stage is used to detect eye areas. This stage employs Haar features to detect face area, color, and intensity mappings to extract eye candidate areas, and some simple geometrical relations for a final decision of the eye regions. In the second stage of the algorithm for detecting eye state, a new proposed descriptor based on a histogram of local maximum vertical derivative patterns in eye areas is extracted and applied to a support vector machine classifier. The proposed descriptor, while having low computational complexity, is defined well enough to describe eye features and hence can distinguish well between open and closed eyes. Experimental results from test images show that the proposed algorithm can correctly detect the eye state by the rate of 98.2%, which is higher than other similar algorithms.

Key words: Eye detection, eye state recognition, local maximum vertical derivative pattern, support vector machine classifier

1. Introduction

Eye state recognition, meaning detecting if an eye is open or closed, is very important in many applications, such as human–machine interfaces (HMI) and driver drowsiness detection. Different methods have been proposed in the field of eye state recognition.

Some researchers have used geometrical eye features to detect the state of the eye. For example, the distance between two eyelids [1] and the standard deviation of this distance [2] is used to detect if the eyes are open or closed. The curvature of the upper eyelid is also used for this purpose [3]. In this method, the curvature of the upper eyelid in a closed eye is negative, so the negativity of the curvature can be an indicator of a closed eye. One of the indicators of an open eye is the presence of the iris in the image of an eye [4]. This feature is used to detect the eye state, and since the iris is circular, its location is determined using a circular Hough transform. The state of the eye can be detected using color data of the image and the amount of exposed sclera [5]. Height to width ratio of the eye and its area are other state defining features of the eye [6]. The major advantages of these methods are their high detection speed and that they do not need any training in advance. However, these methods may have their own problems. For example, for some subjects, their irises may be largely occluded by their eyelids. As a result, it would be very hard to reliably detect irises in this situation [7].

Some other researchers have detected the state of the eye by defining a closed and an open eye template

*Correspondence: fotouhi@tafreshu.ac.ir

and matching it in proposed images [8,9]. The performance of these methods is heavily dependent on proper template definition.

In some papers, the state of the eye is recognized by a classifier that is trained by some extracted features from training samples. In these methods, instead of using visual data, like contours, circularity of the iris, and other geometrical features, textural or structural features of the eye are used. Although these methods need training of the classifier, their efficiency and robustness are significant. For example, a descriptor called LBIIP (local binary increasing intensity pattern) has been used to extract features and an AdaBoost classifier has been used to classify the eye images [10]. As another example, a 2-dimensional discrete cosine transform feature vector and an enhanced hidden Markov model classifier have been used to detect the state of the eye [11]. The study in [7] compared the performance of different classifiers and different methods of feature extraction. Local binary pattern, Gabor wavelet, and the histogram of oriented gradient have been used for feature extraction, and the nearest neighbor classifier, support vector machine, and AdaBoost have been used as classifiers.

In the proposed method of this paper, to increase the accuracy of eye state recognition, we use a novel descriptor, the local maximum vertical derivative pattern (LMVDP), for the feature extraction phase and the least squares support vector machine (LS-SVM) for classification. First, in order to limit the search space and to keep the computational cost low, the face area and then the eye areas are extracted by some simple preprocessing stages. We demonstrate that the proposed method of feature extraction (LMVDP) has a higher recognition rate and shorter feature vector than other methods that are used in eye state recognition.

2. Materials and methods

2.1. Materials

Because of the lack of existence of a standard database of open and closed eyes, 689 images of 109 different persons were provided in this research. These images contain faces with open and closed eyes with different facial expressions. Two hundred samples each of open and closed eye classes were used to train the classifier and the rest of them, 534 open eyes and 360 closed eyes, were used in the test phase. Some of the pictures used in this paper are shown in Figure 1.

2.2. Methods

The proposed algorithm for eye state recognition in this paper consists of 2 stages: a preprocessing stage to detect the face in the image and detect the eyes in the face area, and a second stage to detect the eye state. The preprocessing stage limits the search space, decreases the computational cost, and increases the accuracy of the second stage. In the second stage, a new feature descriptor called the LMVDP and the LS-SVM classifier are used to detect the state of the eye. In the following sections, the proposed algorithm will be studied in more detail.

2.2.1. First stage: locating eyes

This stage plays a key role in the second stage's speed and accuracy, because the eye state recognition process takes place only in the areas extracted in this stage. In this stage first the face area is detected and then eye detection is performed in this area.

Figure 1. Some images of the database and the results of eye detection stage.

2.2.1.1. Locating the face

In order to increase the accuracy and robustness of the algorithm against intensity changes, an intensity compensation process is performed on images using Eqs. (1)–(3).

$$AverageGray = \frac{R_{mean} + G_{mean} + B_{mean}}{3} \tag{1}$$

$$R_{ScaleValue} = \frac{AverageGray}{R_{mean}} \tag{2}$$

$$R_{new} = R_{ScaleValue} * R \tag{3}$$

In Eq. (1), R_{mean}, G_{mean}, and B_{mean} are the mean values of red, green, and blue components of all the image pixels, respectively. R_{new} is the new value of the red component of pixels in the new image. New blue (B_{new}) and green (G_{new}) components are calculated similarly.

After adjusting the intensity values of pixels, the Haar algorithm is used to locate the face area [12]. This algorithm properly handles head rotation, different hair and skin colors, and the existence of glasses, and it also has high detection speed.

2.2.1.2. Locating eyes in the face area

After locating the face area, the eyes should be located in this area. For this purpose, the initial eye region is defined using face region coordinates. Then eye candidates are identified by applying the color and intensity mappings. Finally, eye regions are chosen from previous candidates using some simple geometrical constraints. These 3 stages are explained as follows.

2.2.1.2.1. Determining initial eye regions

The initial eye region is determined in the face area, obtained in previous stage, using Eq. (4):

$$0.2 \times height \leq y\_eye_{area} \leq 0.5 \times height. \tag{4}$$

In this equation, y_eye_{area} is the y coordinate of initial eye area and $height$ is the face height. For example, the detected face area for a sample image, based on Eq. (4), is shown in Figure 2.

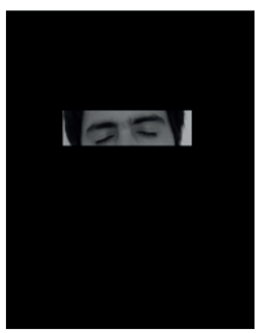


Figure 2. Detected initial eye region.

2.2.1.2.2. Extracting eye candidates using color and intensity mappings

In this step, eye candidate regions are extracted by applying the color and intensity mappings [13] on the regions obtained in the previous step. Color mapping is based on the fact that in the eye area and in the YCbCr color space, the blue component C_b has a greater value than the red component C_r. Based on this fact, intensity mapping is calculated by Eq. (5) [13], and it leads to prominent regions in the eye areas:

$$Eyemapc = \frac{1}{3}\left[C_b^2 + \widetilde{C_r}^2 + \left(\frac{C_b}{C_r}\right)\right]. \tag{5}$$

In Eq. (5), $\widetilde{Cr} = 255 - Cr$.

The eye area usually has both white and black pixels, so proper erosion and dilation morphological operators can be used to make the darker and brighter pixels around the eye region more prominent. Based on this, intensity mapping is calculated by Eq. (6) [13]:

$$EyemapY\left(x,y\right)=\frac{Y_{dilation}\left(x,y\right)}{1+Y_{erosion}\left(x,y\right)}. \tag{6}$$

In Eq. (6), $Y_{dilation}$ and $Y_{erosion}$ are results of applying dilation and erosion operators on the candidate eye area of the previous step in the YCbCr space, respectively. Then color and intensity mappings are multiplied together and a final mapping is obtained. Finally, using a proper threshold, the resulting mapping image is converted to a binary image. The result of this step for the sample candidate eye area of Figure 2 is shown in Figure 3. For the erosion process, a disk structure with radius 2, and for the dilation process, a disk structure with radius 5 and threshold value 66 have been used. The results of color mapping, intensity mapping, and multiplication of both mappings are shown in Figures 3a, 3b, and 3c, respectively. Each of the white regions in Figure 3d is a candidate for the eye region.

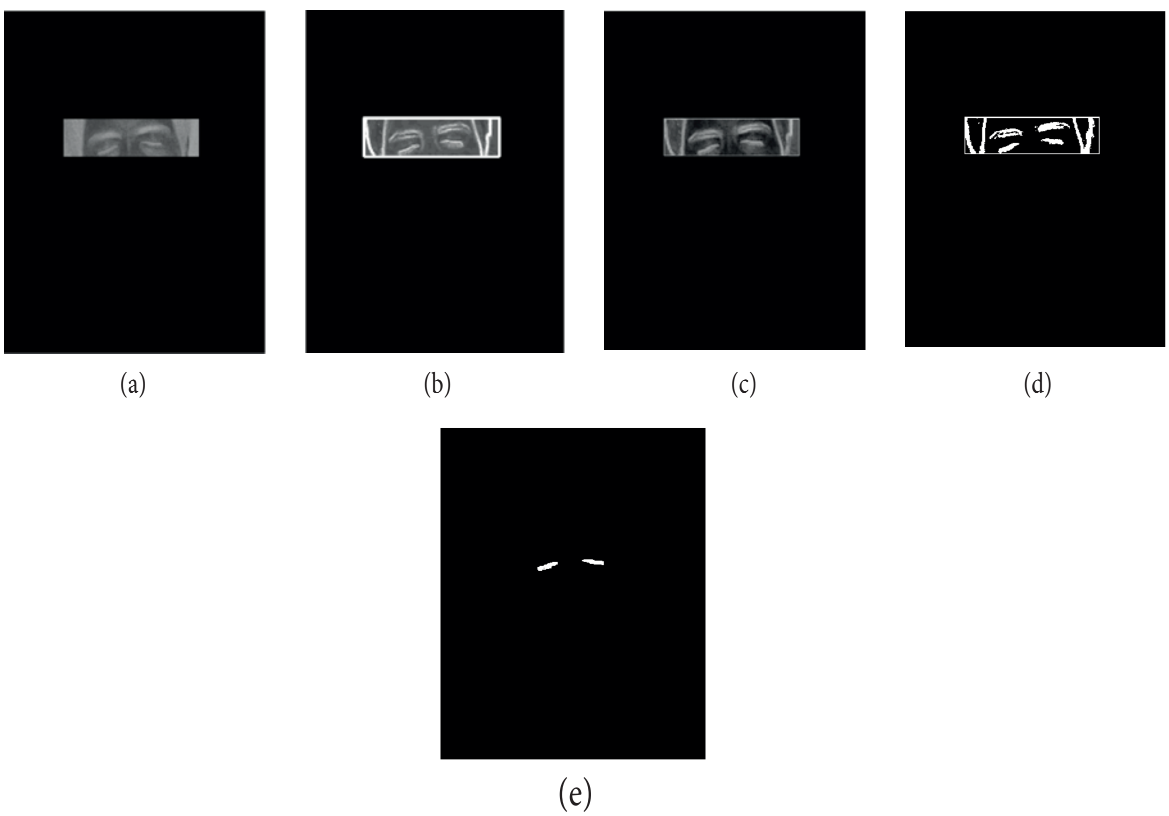


Figure 3. Results of eye candidates' extraction. (a) Color mapping. (b) Intensity mapping. (c) Multiplication of color and intensity mappings. (d) Eye candidate regions after thresholding. (e) Result after eye region confirmation steps.

2.2.1.2.3. Eye region confirmation

Final eye regions are chosen from candidate regions obtained in the previous steps by checking some simple but efficient geometrical constraints:

1. Each candidate area is modeled as an ellipse and the major to minor axis ratio is calculated. To remove noisy narrow areas, this ratio should be less than 10.

2. To remove extra-large areas, the candidate regions should have an area of less than 2500 pixels.

3. Since the eyes are positioned in a certain region of the face in the frontal images, the center of each confirmed eye candidate region must be placed within a distance that is expressed in Eqs. (7) and (8):

$$|y_{face} - y_{eye}| \leq \frac{height}{6}, \tag{7}$$

$$|x_{face} - x_{eye}| \leq \left(\frac{width}{4}\right) + 10. \tag{8}$$

   In these equations, (x_{face}, y_{face}) is the center of the face area and (x_{eye}, y_{eye}) is the center of the candidate eye region, *height* is the face height, and *width* is the face width.

4. If there are multiple candidate regions in a vertical line, the lower region is chosen. This condition can eliminate eyebrow regions from the candidate regions. From the remaining regions, the two regions that are more similar are chosen as final eye regions [14].

Figure 3e shows the results of eye region confirmation steps of this section based on the results of the previous step in Figure 3d.

2.2.2. Second stage: eye state detection

After locating the eyes in the first stage of the algorithm and thus reducing the search space, in the second stage of the algorithm, the eye state is determined in the remaining areas. In this section, we use a new feature descriptor called LMVDP in order to extract the feature vector, and the LS-SVM classifier with an RBF kernel function in order to make the decision about the eye state.

2.2.2.1. Local maximum vertical derivative pattern feature extraction

Since there are more horizontal edges in the eye region and they look different in a closed and an open eye, to classify open and closed eyes it is reasonable to use a texture feature based on vertical changes of intensity in the eye region. A descriptor of the first and higher order local derivative patterns in directions $0°$, $45°$, $90°$, and $135°$ was suggested in [15] in order to be of use in the face recognition application. Based on the idea of [15], in this paper, a new texture feature is proposed, which encodes the location and value of maximum vertical changes of intensity in the candidate eye area. This feature can distinguish well between open and closed eyes because the location and value of maximum vertical changes in the open and closed eyes are different. This new proposed feature will be called LMVDP, local maximum vertical derivative pattern, in the rest of this paper. The LMVDP feature reports the spatial information and the value of intensity changes of eye horizontal edges to the classifier. The reason for using the information of a pixel with maximum intensity changes is to ignore small and noisy edges.

2.2.2.2. LMVDP calculation

In order to calculate the LMVDP in a neighborhood, first we have to determine the first order derivative of pixel intensity in direction 90°. The corresponding calculation is demonstrated in Figure 4. For example, the intensity change for the 0th pixel, D_0, is obtained through Eq. (9), where $I(z_0)$ and $I(z_3)$ are the intensities of pixels z_0 and z_3, respectively.

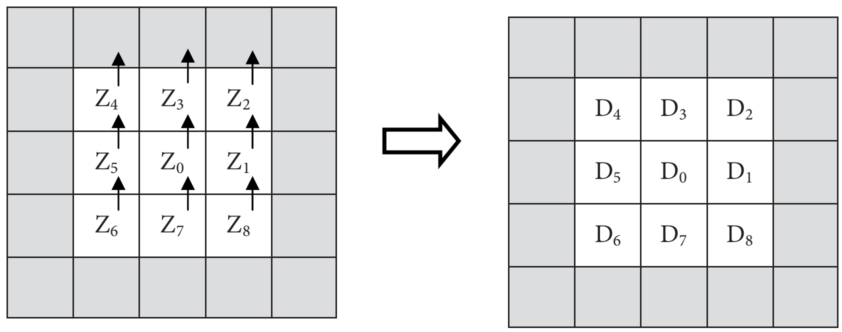


Figure 4. Calculation of first order derivative in direction 90°.

$$D_0 = I(z_0) - I(z_3) \tag{9}$$

After determining intensity derivatives for each pixel, the location of the maximum absolute value of derivatives is determined using Eq. (10). In this stage, we do not care about the sign of intensity changes (transition from dark to light or vice versa). If there are 2 maximums for the absolute values of derivatives, the one with the smaller index number is considered.

$$index = \arg_i \max \{ abs\,(D_i) | \, 0 \leq i \leq 8 \} \tag{10}$$

In Eq. (10), $abs\,(D_i)$ is the absolute value of intensity change in the $i^{th}(0 \leq i \leq 8)$ pixel and *index* is the location of the pixel with the highest absolute value of intensity change.

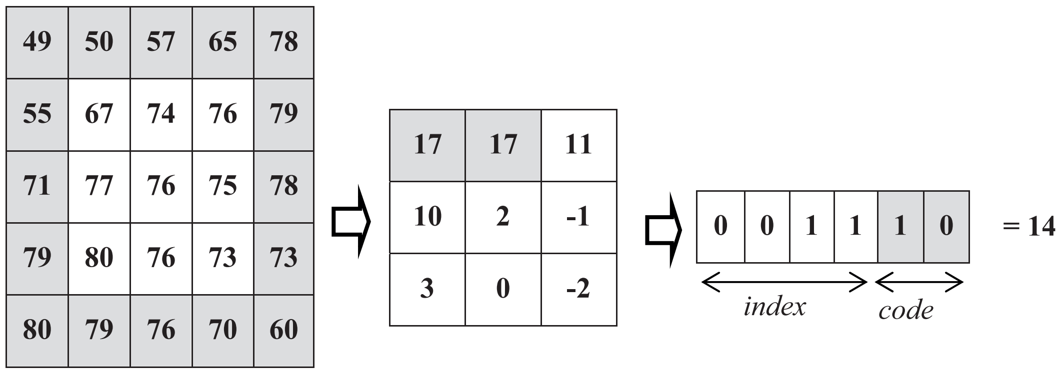
After determining the location of the maximum derivative, the value of its intensity change is encoded using the method suggested in [16] and defining a threshold, which in this paper is suggested to be 10. Eq. (11) shows the encoding procedure, where *code* is a number between 0 and 2 that is obtained according to the value and the sign of the maximum intensity change.

$$code = \begin{cases} 0; \; -10 \leq D_{index} \leq 10 \\ 1 \quad\; ; D_{index} < -10 \\ 2 \; ; D_{index} > 10 \end{cases} \tag{11}$$

Finally, the *LMVDP* code is obtained using Eq. (12) in each neighborhood. This value is placed in the central pixel of the neighborhood. Figure 5 shows different mentioned steps to calculate LMVDP for a 3 × 3 neighborhood. Figure 5a shows a 3 × 3 neighborhood of a pixel of an open eye image area. The first order derivative matrix in 90° and LMVDP binary code are shown in Figures 5b and 5c, respectively.

$$LMVDP = (4 \times index) + code \tag{12}$$

After determining LMVDP for each pixel, a histogram of obtained values in the eye candidate region is given to the LS-SVM classifier. According to Eq. (12), because the maximum value of the index is 8 and the maximum value of the code is 2, the maximum possible value for LMVDP is 34 and so the number of histogram bins is 35.



(a) (b) (c)

Figure 5. Calculation of LMVDP. (a) A 3 × 3 neighborhood of a pixel of an open eye image area. (b) First order derivative matrix in 90°; gray cells are the locations with maximum absolute changes. (c) LMVDP binary code, white-colored bits indicate number 3 in binary, which is the smaller location of maximum absolute changes according to the pattern of Figure 4, and gray bits indicate the value of code 2 obtained from Eq. (11), in binary.

3. Results

In the first stage of the proposed algorithm, in 99.6% of the images, the face region was detected correctly and the eye regions were extracted among those with 97.7% accuracy. In Figure 1, some of the first stage results have been shown. In the second stage of the proposed algorithm, the eye state was correctly detected with a detection rate of 98.2%. Thus, the overall correct detection rate of proposed algorithm is 95.6%.

4. Discussion

4.1. Comparison of proposed algorithm with other methods

To compare the feature vector proposed in this paper with other proposed feature vectors, such as the local binary pattern (LBP) [7] and local binary increasing intensity pattern (LBIIP) [10] used in eye state recognition applications, and other features including local principal texture pattern (LPTP) [16], local sign directional pattern (LSDP) [17], circular LBP (CLBP) [18], and mean LBP (MLBP) [19], used in facial expression recognition applications, and feature circular mean LBP (CMLBP), Table 1 shows the correct classification rate of eye state using all of these features for correctly detected eye areas.

Table 1. Results of different methods for eye state detection stage.

Methods	Open eye	Closed eye	Total
LBP [7]	95.7%	100%	97.4%
CLBP [18]	95.9%	99.7%	97.4%
MLBP [19]	94.4%	99.4%	96.5%
CMLBP	98.1%	98.9%	98.4%
LBIIP [10]	95.9%	98.3%	96.9%
LSDP [17]	96.1%	97.5%	96.6%
LPTP [16]	96.6%	98.1%	97.2%
LMVDP	97.4%	99.4%	98.2%

The performances of different methods can be compared based on their classification accuracy and the length of their extracted feature vector, which itself determines the computational complexity of algorithm. These 2 comparisons are shown in Figures 6 and 7.

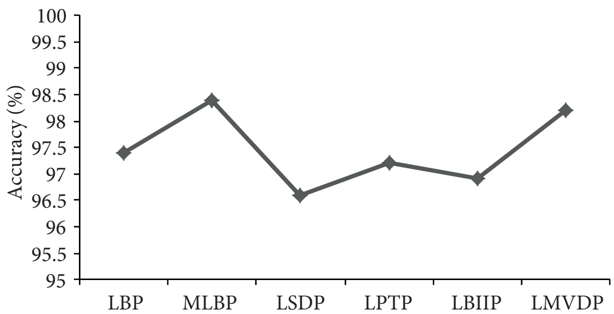


Figure 6. Classification accuracy of different methods for correctly detected eye areas.

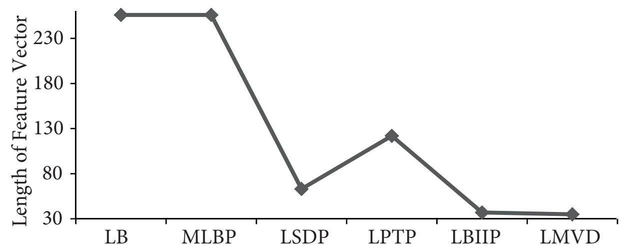


Figure 7. Length of feature vector of different methods.

The following results can be concluded from Table 1 and Figures 6 and 7:

- LMVDP, which has higher classification accuracy with respect to LBP, also has a much smaller feature vector than that of LBP (in the order of 7 times).
- LMVDP, which has a feature vector length close to that of LPTP, LSDP, and LBIIP, has a better classification rate.
- LMVDP classification accuracy is close to that of MLBP, but its feature vector length and thus its computational complexity are much smaller than those of MLBP (in the order of 7 times).

From the results above, it can be concluded that the proposed LMVDP method has a high classification accuracy as well as a short feature vector, which means low computational complexity. Implementing the proposed method and other methods in MATLAB software on the same hardware and software platforms shows a speedup ratio of 4.7 for the proposed method with respect to basic LBP, and a smaller run time for the proposed method with respect to other methods. Table 2 shows the run time of eye state detection using different methods.

Table 2. Run time of different methods for eye state detection stage.

Method	Run time (s)
LBP	2.93
MLBP	2.91
LPTP	0.93
LSDP	1.07
LBIIP	0.64
LMVDP (proposed method)	0.62

4.2. Some experiments on LMVDP and LS-SVM classifier parameters

In order to prove the advantage of using the derivative in direction 90° in LMVDP calculations, this method was implemented in some other directions, i.e. 0°, 45°, 90°, and 135°, and eye state classification accuracies based on them are shown in Figure 8. Comparing the results shows that direction 90° leads to more accurate results than other directions and it proves that features based on horizontal edges in the eye region can be more reliable and distinctive.

As mentioned in Section 2, in LMVDP computations for thresholding the maximum value of vertical intensity difference, a threshold value of 10 has been used. Classification accuracies of the proposed algorithm for threshold values 5, 10, 15, and 20 are compared in Figure 9. This figure shows that threshold 10 leads to better results, but the results of the proposed algorithm are not as sensible to this threshold value.

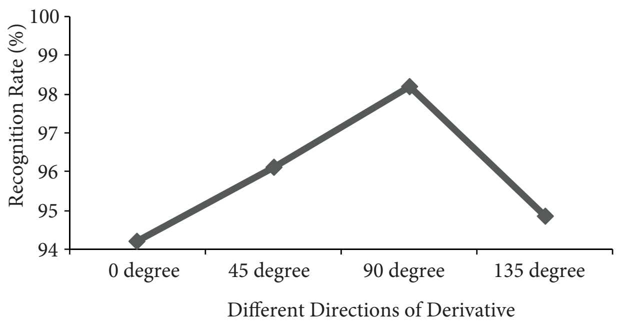


Figure 8. Detection accuracy comparison for different directions.

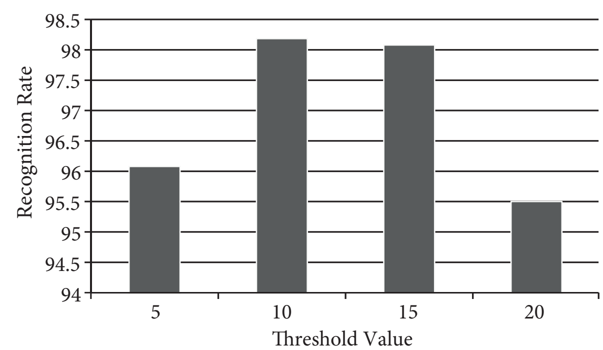


Figure 9. Detection accuracy comparison by using different threshold values.

In this paper, all experiments were implemented using the LS-SVM classifier with an RBF kernel function. Figure 10 illustrates the results for classification accuracy of the proposed algorithm by using different kernel functions including linear, polynomial, and RBF functions. It shows that classification accuracy for the classifier with an RBF kernel function is higher than the others, but it is obvious that the performance of the algorithm is not very dependent on the type of function used.

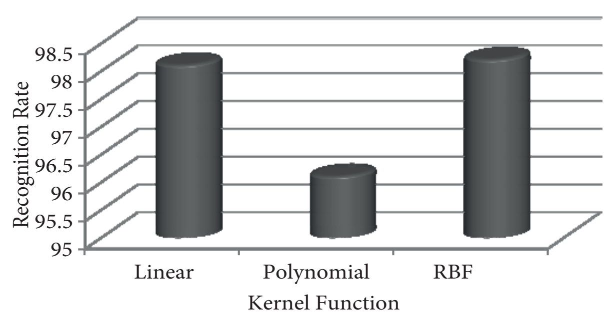


Figure 10. Detection accuracy using different kernel functions for SVM.

5. Conclusion

In this paper, a new algorithm was proposed to distinguish between open and closed eyes. The proposed algorithm first locates the eye regions in the detected face area by combining some previous and new ideas. This leads to limiting the search space for the second stage of the algorithm. Secondly, to detect the eye state, a novel local descriptor called LMVDP has been proposed. This descriptor reports, with low computational complexity, the spatial information and the intensity variations of eye horizontal edges to a SVM classifier. The proposed LMVDP feature has a higher detection accuracy and shorter feature vector than other features that are used in eye state detection. Improving the performance of the proposed algorithm for nonfrontal face images and eyes with glasses or images in the night is the future aim of this research. Using the proposed feature vector with other classifiers and comparing it with other feature vectors can be considered in future studies. Applying the proposed method in related applications can also be considered in future work.

References

[1] Dong W, Wu X. Fatigue detection based on the distance of eyelid. In: IEEE International Workshop on VLSI Design and Video Technology; 2005. New York, NY, USA: IEEE. pp. 365-368.

[2] Yunqi L, Meiling Y, Xiaobing S, Xiuxia L, Jiangfan O. Recognition of eye states in real time video. In: International Conference on Computer Engineering and Technology; 2009. pp. 554-559.

[3] Guo J, Guo X. Eye state recognition based on shape analysis and fuzzy logic. In: Intelligent Vehicles Symposium; 2009. pp. 78-82.

[4] Alioua N, Amine A, Rziza M, Aboutajdine D. Eye state analysis using iris detection based on circular Hough transform. In: International Conference on Multimedia Computing and Systems; 2011. pp. 1-5.

[5] Zoroofi RA, Tabrizi PR. Open/closed eye analysis for drowsiness detection. In: First Workshops on Image Processing Theory, Tools & Applications; 2008. pp. 1-7.

[6] Du Y, Ma P, Su X, Zhang Y. Driver fatigue detection based on eye state analysis. In: 11th Joint Conference on Information Sciences; 2008.

[7] Liu X, Tan X, Chen S. Eyes closeness detection using appearance based methods. In: 7th International Conference on IFIP Advances in Information and Communication Technology; 2012. pp. 398-408.

[8] Yutian F, Dexuan H, Pingqiang, N. A combined eye states identification method for detection of driver fatigue. In: International Communication Conference on Wireless Mobile and Computing; 2009. pp. 217-220.

[9] Fa-Deng G, Min-Xian H. Study on the detection of locomotive driver fatigue based on image. In: 2nd International Conference on Computer Engineering and Technology; 2010. pp. 612-615.

[10] Zhou L, Wang H. Open/closed eye recognition by local binary increasing intensity patterns. In: 5th International Conference on Robotics, Automation and Mechatronics; 2011. pp. 7-11.

[11] Qin H, Liu J, Hong T. An eye state identification method based on the embedded hidden Markov model. In: IEEE International Conference on Vehicular Electronics and Safety; 2012. New York, NY, USA: IEEE. pp. 255-260.

[12] Viola P, Jones M. Rapid object detection using a boosted cascade of simple features. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition; 2001. New York, NY, USA: IEEE. pp. 511-518.

[13] Hsu RL, Abdel-Mottaleb M, Jain AK. Face detection in color images. IEEE T Pattern Anal Mach Intell 2002; 24: 696-706.

[14] Beigzadeh M, Vafadoost M. Detection of face and facial features in digital images and video frames. In: Cairo International Biomedical Engineering Conference; 2008. pp. 1-4.

[15] Zhang B, Gao Y, Zhao S, Liu J. Local derivative pattern versus local binary pattern: face recognition with high-order local pattern descriptor. IEEE T Image Process 2010; 19: 533-544.

[16] Ramirez Rivera A, Rojas Castillo JA, Chae O. Recognition of face expressions using local principal texture pattern. In: 19th IEEE International Conference on Image Processing; 2012. New York, NY, USA: IEEE. pp. 2609-2612.

[17] Castillo JAR, Rivera AR, Chae O. Facial expression recognition based on local sign directional pattern. In: 19th IEEE International Conference on Image Processing; 2012. New York, NY, USA: IEEE. pp. 2613-2616.

[18] Huang D, Shan C, Ardabilian M, Wang Y, Chen L. Local binary patterns and its application to facial image analysis: a survey. IEEE T Syst Man Cy C 2011; 41: 765-781.

[19] Bai G, Zhu Y, Ding Z. A hierarchical face recognition method based on local binary pattern. In: Congress on Image Signal Process; 2008. pp. 610-614.