AIVP ADVANCES IN IMAGE AND VIDEO PROCESSING

SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM

VOLUME 2, ISSUE 5
ISSN: 2054 - 7412

An Efficient and Effective Framework for Eye Region Detection and Eye State Recognition

Cheng-Chieh Chiang
Department of Information Technology, Takming University of Science and Technology
Taipei, Taiwan
kevin@csie.ntnu.edu.tw

ABSTRACT

This paper proposes a framework to treat the eye information in a face image including the eye region detection and the eye state recognition. In order to make it possible to employ our approach to a real application in practice, our goal in this paper is to design not only a fast enough but also a high performance framework for the eye region detection and the eye state recognition. Our proposed framework mainly contains two parts: the first is to locate the eye regions in a face image, and the second is to recognize the states, either open or closed, of the eye regions. When a frame is captured from a video sequence, a face detection method is first performed to determine the positions of face regions. Next, face regions are converted into binary images and then we perform the horizontal and the vertical projections to locate the eye regions. Two visual features containing the intensity values of pixels and local binary pattern (LBP) are extracted from eye regions to classify the eye states with the support vector machine (SVM) approach. This paper also demonstrates a several experiments to present the efficiency and effectiveness of our proposed framework.

Keywords: Eye Region Detection, Eye State Recognition, Local Binary Pattern, Support Vector Machine.

1 Introduction

Computer vision has been widely applied to many domains in our life. For example, the eye tracking system [1][2] can automatically detect eye regions and then track the user gaze in order to understand what targets the users are looking at. Several important issues, including eye region detection/tracking/state recognition and eye as bioinformatics for identification, to treat eye images/videos in computer vision have been active in recent. This paper presents our effort to propose a framework that cannot only locate eye regions but also recognize the eye states in a video frame captured from a common webcam or PTZ camera.

Two main tasks are considered in this paper: (i) eye region detection and (ii) eye state recognition. The most fundamental task is to locate the eye regions when a human face is determined. Therefore, the eye state recognition can indicate some evidences of the subject (human). Let us consider an example in car driving. Camera systems have been installed in cars to monitor drivers' status in order to provide enough assistances if necessary. When the system finds that the driver's eye status keeps closed in a predefined time period, it can strongly imply that the driver is in drowsing. Similar functions can be also applied to our classrooms. The eye state recognition can automatically count the drowsing students to understand the attraction in course contents. This paper proposes a real-

DOI: 10.14738/aivp.25.479
Publication Date: 17th October 2014
URL: http://dx.doi.org/10.14738/aivp.25.479

time framework to efficiently locate the eye regions in a face image and determine the corresponding eye states.

Figure 1 draws the brief flowchart of our work in this paper. When a video frame is captured, a face detection method is first employed to demonstrate the face regions in image. Here we only consider a single frame of video, and the advanced applications should concatenate the multiple decisions of frames from a video sequences. We do not limit the number of faces appeared in image, but only request the resolution of eye regions. That is to say, the system has to guarantee the eye regions large enough (at least 50 pixels in width in our experiments) such that more information included for well recognition. Given a face region, all of pixels are converted to gray levels and then converted to binary. The binarization process for a face region can enhance the contrast between the pupil and its surrounding areas. Then, the horizontal and vertical projection that is simple but efficient is designed to locate the eye regions in the face region. Two kinds of visual features are extracted from eye regions, containing the intensity values of pixels and the local binary patterns (LBP) [3]. We tried to employ the two visual features individually with the support vector machine (SVM) [4] to recognize the eye states. Our experiments in this paper demonstrate the comparison of the efficiency of the two visual features used in the eye state recognition.

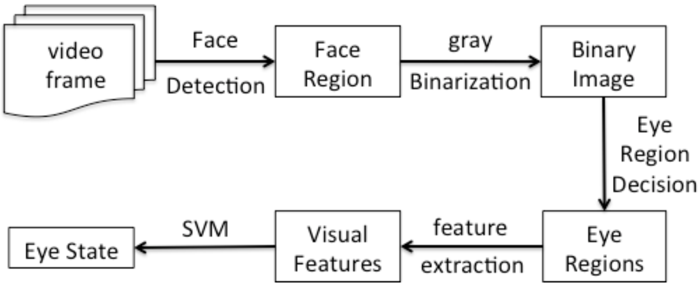

Figure 1: Flowchart

Our contribution in this paper mainly involves two parts. First, we propose a simple but efficient framework for eye state recognition. Our approach can achieve the real-time requirement such that less limitation is given in a real eye-related application. The proposed framework Moreover, each component in Figure 1 can be replaced to other more proper methods. For example, we can employ other visual features such as entropy or other informatics features to represent the eye regions for recognition. Second, this paper compares the efficiency of the two visual features (the intensity values of pixels and LBP) with the SVM in the eye region recognition. In our experiments, the visual representation using the intensity values of pixels can achieve a better performance than that using LBP in the eye state recognition.

The remainder of this paper is organized as the follows. Section 2 provides a brief review of the related works for the eye region detection and the eye state recognition. Section 3 presents our approach to detect eye regions in a face image, including face detection, image binarization, and eye region decision. Section 4 introduces the details of the eye state recognition used in our proposed framework. Section 5 first describes the experimental setup and data and then demonstrates the experimental results to show the performance of our proposed method. Section 6 makes a conclusion for this paper and provides possible extensions in the future.

2 Related Works

Many practical systems for eye tracking have been developed in market such as Tobii [1] and SMI [2]. How to well localize the eye regions in a face and then to identify the eye states are the two most fundamental issues in an eye tracking system.

Eyes locate at the fixed region on human face, where the eye regions are often darker than the surrounding skin mainly due to the low intensity of the pupil. Hence, an intuitive method to detect the eye regions is to employ the horizontal and vertical projection to locate the low intensity area in a binary face image [5]. Our proposed method to locate the eye regions basically follows the idea in [5] with some necessary modifications. R. Valenti and T. Gevers employed [6] the shape of the eye that is a significant feature in face to locate the eye regions. The pupil looks almost circle when an eye is open, so their approach sampled the pupil edges to identify the circle center that is also the eye center in face. Y. Wu et al. extracted the ellipse edges of eye as visual features for template matching to locate the eye regions [7]. Also, the OpenCV library included a template matching approach for eye region detection by using Haar-like features [8].

The eyes in general contain two states: open and closed, with the assumption of the eye regions available. J. Wu and M. M. Trivedi integrated the features in face to determine the eye states and to track the pupil movement [9]. H. Ta and Y.-J. Zhang recognized the eye states by looking for the iris of eye: iris found indicates the eye open [10]. Considering the variation of eye heights in open and closed, M. Dehnavi et al. performed the vertical project in the binary eye image to determine the eye open and closed with maximum and minimum thresholds, respectively [11]. C. Xu et al. adopted the LBP approach [12] for representing eye images and then construct an adaboosting classifier for eye state recognition [13]. Similarly, Y.-S. Wu et al. also employed the LBP approach but the SVM classification was adopted [14]. These two works presented the good performance of using the LBP feature for eye image representation.

3 Detection for Eye Regions

The first main issue in this paper is to detect the eye regions given a face image. This component shown in Figure 1 can be divided into three steps: face detection, conversion of the binary image, and eye region decision. This section describes the details of the three steps in the follows.

3.1 Face Detection

Face detection is a key technology to locate or identify human in image. Many researchers have paid attention to treating the face detection in either image or video under different conditions of the real world [15][16].

This work adopted the adaboosting approach [17], which was first proposed by Viola and Jones in 1999, to automatically detect face regions in a video frame. This approach collects a large number of Haar-like features that are fast computed by the integral image. Then, an adaboosting approach [18] is employed to select most significant Haar-like features of face regions. In general, many Haar-like features may be contained to construct an efficient classifier in the adaboosting approach, but that should also need more computational time for face detection. In order to achieve the real-time requirement, Viola and Jones designed a cascading structure, the basic concept is shown in Figure 2, to fast filter most of trivial non-face regions out using fewer Haar-like features in the several starting stages and then accurately to detect face regions using more features in the ending stages.

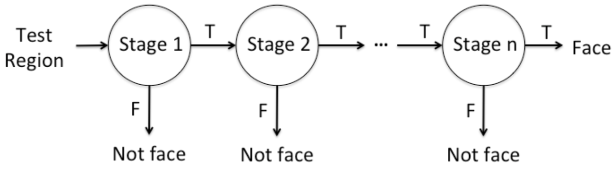

Figure 2: The cascade structure

The adaboosting approach of face detection was implemented in the openCV library and was widely used in many applications. Although this approach may be failed if the orientation of face regions is skew, it is still appropriated for our framework due to skewed human faces can be simply ignored in eye-related applications such as the eye state monitoring for car drivers.

3.2 Binarization

In face, the eye region contains a strong characteristic that there is high contrast between the eyes and the surrounding areas. Hence, the gray channel is more appropriate that color channels in representing eye regions. The other advantage of using the gray channel is to reduce the computational complexity in process.

This step of binarization contains two parts. The first is to convert the color to the gray image by using the simplest averaged method: gray = (R+G+B)/3, where R, G, and B are the pixel values in the red, green, and blue channels, respectively. Next, the gray image of face is thresholded to a binary image by using the Otsu method [19]. Figure 3 demonstrates an example from a color face image to a binary image.

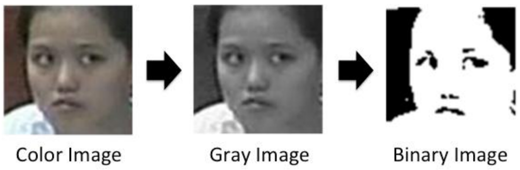

Figure 3: A color face image to a binary image

3.3 Eye Region Decision

Many state-of-the-art technologies have been developed to eye region detection, stated as the above in Section 2. Since our work aims to design an eye state recognition for real-time applications, it is necessary to look for an efficient but fast approach to treat the eye region decision.

When a face has been located without skew, we can observe that the eye area is almost fixed on the upper-to-middle of the face. Hence, researchers defined a rough area in face to reduce the search space for the eye regions [20]. Let the width and the height of the face image be *w* and *h*, respectively. We averaged the position of eye in face in the training data, which is described in Section 5, to define that the left and right eyes individually locate at a rectangle region ((0.2, 0,23), (0.42, 0.5)), and ((0.58, 0.23), (0.8, 0.5)), respectively, in a face, where each of coordinates contains the left-upper and right-down corners. The candidate of eye regions defined in this work is drawn in Figure 4.

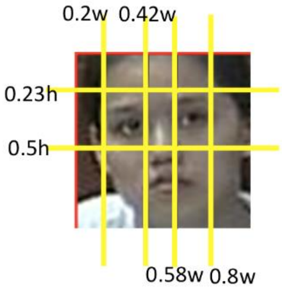

Figure 4: Candidate of two eye regions in a face

More advanced tasks are necessary to accurately extract the eye regions, such as ignoring the eyelashes and the shadow of eye. Due to our real-time requirement, we adopted the horizontal and vertical projection method [5] to fast but efficient capture the eye regions. Assume that *I*(*x*, *y*) is the pixel value at (x, y) in a candidate of eye region with size *m* by *n* pixels. Because the image is converted to binary, the pixel value is either 0 or 1. The horizontal (H_{porj}) and vertical (V_{proj}) projection on the binary image can be defined as the follow,

$$H_{proj}(x) = \sum_{x=1}^{n} I(x,\ y),\ \text{where } x = 1 \text{ to } m$$

$$V_{proj}(y) = \sum_{x=1}^{m} I(x,\ y),\ \text{where } y = 1 \text{ to } n.$$

An example of horizontal projection to align the horizontal center of an eye region is drawn in Figure 5. The distribution of the horizontal projection in Figure 5(c) contains two obvious peaks, where the upper peak corresponds to the eyelashes and the lower one to the eye region. Thus, we take the lower peak to align the horizontal center of the eye region. Moreover, the vertical projection to align the vertical center of the same eye region is drawn in Figure 6. Both in the closed and open eyes in this example, the peak of the vertical projection in Figure 6(c) can indicate the center of the eye region. Hence, the intersection of the horizontal and the vertical projection can exactly indicate the eye center.

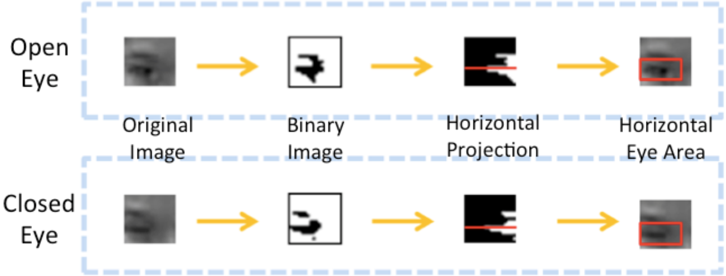

Figure 5: Example of the horizontal projection in an eye region

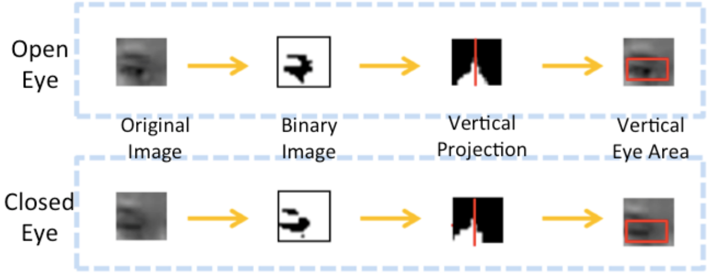

Figure 6: Example of the vertical projection in an eye region

4 Eye State Recognition

The second main issue in this paper is to recognize the eye state given an eye region. To this end, a visual feature is extracted for representing the eye region and then a classification method is applied to recognize the eye state. Here only two eye states are considered: closed and open. In this paper, we adopted two kinds of visual features: pixel values and the LBP [3] in an eye region, and we employed the SVM approach [4] to the eye state classifier. In order to have the same length of feature vectors for the SVM classifier, we resize all eye regions (both training and test) to the equal size.

In our experiments described in the next section, we compare the performances of two visual features in the eye state recognition. The first one is to simply collect all values of row-major pixels in the eye region as a feature vector. In this case, all raw information in pixels is preserved without any encoding.

The second visual feature used in our experiments is the LBP method. The details of the LBP method are described in [12], and we only briefly introduce the concept in this section using the example in Figure 7. Assume that Figure 7(a) is a 3x3 cell in an image. Except the center value, all pixels are assigned a binary value shown in Figure 7(b): set 1 if the pixel value is greater than the center value, and set 0 otherwise. That is to say, the corresponding binary cell in Figure 7(b) remarks an intensity change of the cell associated with the center pixel. Moreover, a fixed weighted mask in Figure 7(c) encodes the binary cell to a LBP value in Figure 7(d). An image can then be converted to a series of LBP values by sliding 3x3 cells.

53	31	22
57	41	38
21	66	43

(a)

1	0	0
1		0
0	1	1

(b)

2^0	2^1	2^2
2^7		2^3
2^6	2^5	2^4

(c)

The LPB value of cell (a) is: $2^0+2^4+2^5+2^7=1+16+32+128=177$

(d)

Figure 7: LBP (a) a 3x3 cell, (b) the corresponding binary, (c) the fixed weights, and (d) the LBP value of (a)

Collecting visual features of the training eye regions, an SVM classifier is learnt to recognize test eye regions. SVM is a supervised learning model and is well-known to achieve a good performance in classification. In implementation, we adopted LIBSVM library [21] with a radial basis kernel. LIBSVM library was developed by Machine Learning and Data Mining Group, National Taiwan University, and can support the main functions of SVM classifiers. The SVM approach designs a linear classifier to best separate positive and negative data by the support vectors. However, a linear method is not able to well classify data in a complex problem, hence we adopt a radial basis kernel to project data into a very-high dimensional space such that the classification problem is separatable.

5 Experiments

5.1 Setup and Data Set

Figure 8 demonstrates the practical environments in our experiments that contain short and long distances to capture human eyes. In the both two cases, each of face images has to be larger than 60x60 pixels in order to catch enough information in eyes. All experiments are performed at a desktop that is with Intel(R) Core(TM) i7-3770: CPU @ 3.4GHz, RAM 8.0G, and windows 7.

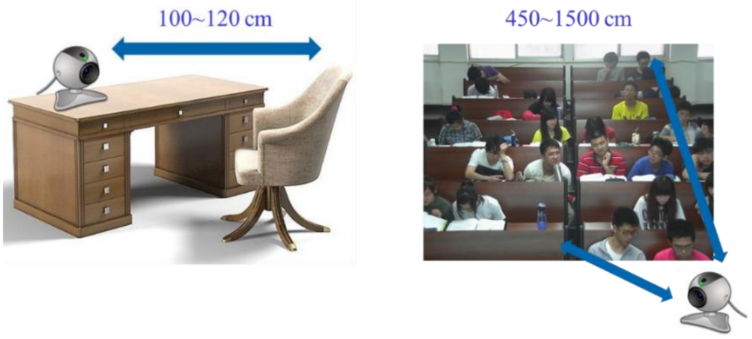

Figure 8: Practical environments in our experiments. Left: shot distance; right: long distance.

Two data sets are collected in our experiments. The first data set is called D1 that asked for 17 people without glasses looking at the screen in the short distance in the left of Figure 8. There are six types of gazes in D1: front, up, down, right, left, and eye closed. In total, D1 contains 2260 eye images where 1500 for open and the other 760 for closed. The second data set is called D2 that was captured in classroom to record students' eye states in the right of Figure 8. This data set covers 39 people, 28 with glasses and the other 11 without glasses. Due to a long distance, the eye regions in D2 are low resolution. Table 1 summarizes the information of D1 and D2.

Table 1: Summary of our data sets D1 and D2

Data Set: D1	Open Eye					Closed Eye
Gaze	front	up	down	left	right	
# for training	100	100	100	100	100	500
# for test	200	200	200	200	200	260
total	300	300	300	300	300	760
Data Set: D2	Open Eye					Closed Eye
# for training	250					250
# for test	750					500
total	1000					750

5.2 Results

Our experiments mainly aim to evaluate the performance of our framework that applies the SVM approach as a classifier with two kinds of visual features: pixel values and the LBP. We adopted the LIBSVM library [21] with a radial basis kernel that is developed by Machine Learning and Data Mining Group, National Taiwan University. The number of training and test sets are listed in Table 1.

In the data set D1 that considers a short distance to capture eye images, we took 1000 images of the open and closed eye for training, and the others 1260 images for test. The recognition results using the two visual features are drawn in Table 2, which presents the pixel values can obviously achieve a better performance than LBP. Moreover, the LBP with SVM is faster than the pixel values with SVM.

Table 2: Recognition rates of the two visual features in data set D1.

	Open Eye (%)	**Closed Eye (%)**	**Avg. Exec. Time (sec/per frame)**
Pixel Values + SVM	99.8	100	0.0610
LPB + SVM	94.5	100	0.0401

Table 3: Recognition rates of the two visual features in data set D2

	With glasses			**Without glasses**		
	Open Eye (%)	**Closed Eye (%)**	**Avg. Exec. Time (sec/per frame)**	**Open Eye (%)**	**Closed Eye (%)**	**Avg. Exec. Time (sec/per frame)**
Pixel Values + SVM	94.53	88.26	0.0636	96.6	95.4	0.0542
LPB + SVM	82.6	86.9	0.0404	84.4	88.8	0. 0404

In the data set D2 that treat a long distance to capture low quality eye images, we took 500 images of the open and closed eye for training, and the others 1250 images for test. The recognition results using the two visual features are drawn in Table 3, which consists of with and without glasses. The results in Table 2 and 3 clearly present the better recognition rates using the feature of pixel values in both short and long distances.

6 Conclusion and Future Work

This paper presents a framework based on computer vision methods to treat eye region detection and eye state recognition. In the eye region detection, a simple but effective method is proposed to fast locate the eye regions in a face image. Then we employ two kinds of visual features, pixel values and LBP values, with SVM approach to classify the eye states. We also perform several experiments to demonstrate the performance of our proposed framework using these two visual features. The potential extension of this work can employ other efficient features including color, texture, and shape to improve the representation for eye regions. Another important task is to apply our proposed method to a practical system such as monitoring the driver status for car driving and recording the student status in classroom.

REFERENCES

[1]. Tobii, http://www.tobii.com/

[2]. SMI, http://www.smivision.com/en.html

[3]. Ojala, T., Pietikäinen, M., and Harwood, D., A comparative study of texture measures with classification based on feature distributions. Pattern Recognition, 1996. 19(3): p. 51-59.

[4]. Cortes, C. and Vapnik, V., Support-vector networks. Machine Learning, 1995. 20(3): p. 273-297.

[5]. Zhou Z.-H. and Geng X., Projection functions for eye detection. Pattern recognition, 2004. 37(5): p. 1049-1056.

[6]. Valenti R. and Gevers T., Accurate eye center location and tracking using isophote curvature. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2008. p. 1-8.

[7]. Wu Y., Liu H., and Zha H., A new method of detecting human eyelids based on deformable templates. Proceedings of IEEE Conference on Systems, Man and Cybernetics, 2004. p. 604-609.

[8]. OpenCV, http://opencv.org/

[9]. Wu J. and Trivedi M. M., An eye localization, tracking and blink pattern recognition system: algorithm and evaluation. ACM Transactions on Multimedia Computing, Communications, and Applications, 2010. 6(2): p. 599-601.

[10]. Tan H. and Zhang Y.-J., Detecting eye blink states by tracking iris and eyelids. Pattern Recognition Letters, 2006. 27(6): p. 667-675.

[11]. Dehnavi M., Attarzadeh N., and Eshghi M., Real time eye state recognition. Proceedings of the 19th Iranian Conference on Electrical Engineering, 2011. p. 1-4.

[12]. Ojala T., Pietikainen M., and Maenpaa T., Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002. 2497): p. 971-987.

[13]. Xu C., Zheng Y., and Wang Z., Efficient eye states detection in real-time for drowsy driving monitoring system. Proceedings of International Conference on Information and Automation, 2008. p. 170-174.

[14]. Wu Y.-S., Lee T.-W., Wu Q.-Z., and Liu H.-S., An eye state recognition method for drowsiness detection. Proceedings of IEEE Vehicular Technology Conference, 2010. p. 1-5.

[15]. Hsu R.-L., Abdel-Mottaleb M., and Jain A. K., Face detection in color images. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002. 24(5): p. 696-706.

[16]. Liao S., Jain A. K., Li S. Z., Partial face recognition: alignment-free approach. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013. 35(5): p. 1193-1205.

[17]. Viola P. and Jones M. J., Robust real-time face detection. International journal of computer vision, 2004. 57(2): p. 137-154.

[18]. Freund, Y., Schapire, R., and Abe, N., A short introduction to boosting. Journal-Japanese Society For Artificial Intelligence, 1999. 14: p. 771-780.

[19]. Otsu N., A threshold selection method from gray-level histograms. IEEE Transactions on Systems, Man and Cybernetics, 1979. 9(1): p. 62-66.

[20]. Królak A. and Strumiłło P., Eye-blink controlled human-computer interface for the disabled. Advances in Intelligent and Soft Computing, 2009. 60: p. 123-133.

[21]. [LIBSVM] LIBSVM: http://www.csie.ntu.edu.tw/~cjlin/.