EYE DETECTION BASED ON RECTANGLE FEATURES AND PIXEL-PATTERN-BASED TEXTURE FEATURES

Huchuan Lu, Wei Zhang, Deli Yang
Department of Electronic Engineering, Dalian University of Technology, Dalian Liaoning 116023, China, lhchuan@dlut.edu.cn

ABSTRACT
Eye detection can be used in intelligent human–computer interfaces, driver drowsiness detection, security, and biology systems. In this paper a new method for eye detection based on rectangle features and pixel-pattern-based texture feature (PPBTF) is proposed. First, Adaboost cascade classifier by rectangle features is constructed to do rough eye detection in a front facial image. Second, the result image patches are cropped and scaled to 24×12 to compute the features of PPBTF, then, put these features into an Adaboost and SVM classifier for an accurate detection. Some eye open front facial images from FERET Database are chosen to do the experiments, and the results show that the approach is real time and is effective for eye detection.

Index Terms—eye detection, rectangle feature, PPBTF, cascade classifier

1. INTRODUCTION
Eye detection is invaluable in determining the orientation of the face and also the gaze direction. Various schemes have been used to eye detection. For instance, since the pupils generally appear darker than the surrounding regions, the algorithms search the darker regions were proposed [1]. In [2] they used a generalized projection function (GPF) to locate the eye center and further the eye. However, such algorithms are highly sensitive to the lighting conditions, and are not sufficient for some facial features, such as eyebrows. Hough transform was also employed [3], which implies a preliminary robust edge detection procedure. Recently, Local Binary Patterns (LBP) as low-cost appearance features was also widely adopted for face detection and facial expression recognition [4, 5].

In this paper, a new method for eye detection based on rectangle features and pixel-pattern-based texture feature (PPBTF) is proposed. Rectangle features were firstly proposed by Paul Viola [6], in Paul Viola’s paper integral image and cascade classifier were proposed and these methods made the detection system real time. This paper uses Viola’s conception, and adds some new rectangle features to construct a cascade classifier for rough eye detection. The results from the classifier usually have some errors, such as eyebrows, mouth, nares, larger or smaller eyes, so pixel-pattern-based texture feature (PPBTF) is introduced to recognize the proper eye patches. PPBTF was proposed in [7], it is very fast and is applied to texture image segmentation successfully. The result image patches out from the cascade classifier are cropped and scaled to 24×12, and then PPBTF is computed on the scaled image patches, Adaboost and SVM classifier is used to classify the features and the result 96.79% shows that the PPBTF is effective to classify the eye and non-eye image patches.

This paper is organized as follows: In section 2, a cascade classifier is described, using the rectangle features. Section 3 introduces the PPBTF method and the Adaboost and SVM classifier. Experimental results are discussed in Section 4 and in Section 5 the conclusions are received.

2. RECTANGLE FEATURES AND CASCADE CLASSIFIER

2.1. Rectangle Features
Viola has used four features (Fig.1) for face detection and these features performed well on face. But for eye detection these features seem to insufficient, because eye has its unique characteristic different from face, and new features should be proposed to work with eye detection. Fasel [8] has proposed a new one (see Fig.2(F)), and in this paper some new rectangle features are also proposed (Fig.2).

These features are proposed according to the appearance of the eye. Since eye corners usually appear darker than the neighborhood, feature F could express this characteristic, so it should be introduced. Similarly, eyeballs are often darker than the other parts of the face, so G is involved. Feature K and L represent the edge around the corner of the eye. J shows the characteristic between the edge of eyeball and the eye corner. Examples of features express the eye can be seen in Fig.3, and these features will improve the precision of the detection and reduce the amount of the features, the detailed description will be discussed in section 4.

![Fig.1. Face detection features that were used by Viola.](image-url)
2.2. Cascade Classifier

Cascade of classifier achieves increased detection performance while radically reducing computation time. The key insight is that smaller, and therefore more efficient, boosted classifiers can be constructed which reject many of the negative sub-windows while detecting almost all positive instances. Simpler classifiers are used to reject the majority of sub-windows before more complex classifiers are called upon to achieve low false positive rates. The stage in cascade is constructed by Adaboost. (see Fig.4).

2.3. Problem analysis

After the detection of cascade classifier, some errors could come out, such as eyebrows, mouth, nares, larger or smaller eyes, to exclude these patches, some other method to classify the eye and non-eye patches that come out from the cascade classifier is necessary, and this method had better simple and fast to compute. So this paper involves PPBTF to solve this problem, this approach is only based on the texture of the image, and just meeting the demand that is simple and speedier.

This paper builds an Adaboost cascade classifier with only 7 stages, and facial images from FERET Database are put into the classifier to do a rough detection of eyes, then the result patches will be cropped, scaled and classified by Adaboost and SVM classifier using PPBTF described in the next section.

3. THE PPBTF AND ADABOOST AND SVM CLASSIFIER

3.1. PPBTF

PPBTF is firstly proposed in [7]. It is constructed from one pattern map. A gray scale image is first transformed into a pattern map in which edge and background pixels are classified by pattern matching with a given set of M pattern templates \( \{w_i\} \) that reflect the spatial features of images. For each pixel \((x, y)\) in a gray scale image \(I\), let \(z_i\) be the inner product of its \(S \times S\) neighbor block \(b\) with the \(i\)th pattern templates

\[
z_i = |b \bullet W_i|
\]

Then the pixel \((x, y)\) in the pattern map \(P\) is assigned a number \(k\) such that \(z_k = \max(z_1, z_2, \ldots, z_M)\). Therefore, the pixel values in a pattern map represent the pattern classes of pixels in the original gray scale image.

The feature model comes as follows: Suppose the number of patterns is \(M\), then the pixel value \(P(x, y)\) in the pattern map \(P\) is in a range of \([1, M]\). For each pixel \((i, j)\), the features in a window \(S1 \times S1\) can be generated by

\[
f_i(i, j) = \sum_{x=-S/2}^{S/2} \sum_{y=-S/2}^{S/2} h_i(x, y),
\]

where the function \(h\) is a binary function defined by

\[
h_i(x, y) = \begin{cases} 1 & \text{if } P(x, y) = i, \\ 0 & \text{otherwise}. \end{cases}
\]

Thus, the feature \(F_{PPBTF}\) gives the number of the pixels belonging to the pattern, and the feature vector is constructed with as components

\[
F_{PPBTF} = (f_1, f_2, \ldots, f_M).
\]

3.2. PCA basis function as pattern templates

3.2.1. Image coding by principal component analysis

Pattern templates represent the spatial features in an image and reflect that how the value of 1 pixel depend on those of its neighbors. PCA based pattern template is used in [7], in this paper this template is also been adopted, and has got a good result. The process of obtaining the templates is as follows: First for a given image, an \(S \times S\) image block is denoted as a neighbor vector, and \(N\) numbers of \(S \times S\)
image blocks are randomly created. If the image numbers are N, and an \((S \times S) \times (M \times N)\) matrix A would come out, then PCA will be computed using the matrix A. After PCA analysis eigenvectors are obtained, and some of the eigenvectors are selected as the template.

In the experiments 1000 block samples by the size of \(3 \times 3\) are generated randomly for each eye image, and 1000 eye images by the size of \(24 \times 12\) are used as template train samples, so the matrix \(9 \times 1000000\) is analyzed by PCA, and 9 basis functions sorted by decreasing order of eigenvalues are got. The first one corresponding to the largest eigenvalue is a Gaussian low-pass filter, and the others are derivative filters. Except the first basis function, the others can be used like gradient filters for pattern matching. In this paper except the first basis function, the following 5 basis functions are chosen as the pattern templates.

### 3.2.2. PCA pattern map

A PCA pattern map is obtained through the pattern matching process using the pattern templates mentioned in 3.2.1. In the transformation of a gray scale image \(24 \times 12\) eye image in this paper) to a pattern map, each pixel is assigned the index of the templates which best matches its \(3 \times 3\) neighbor block. Since the templates resemble derivative operators, the value of a pixel in the pattern map represents the edge pattern class of its neighbor block in the original gray scale image (Fig.5).

Fig. 5. The left image is the original eye image and the right is the corresponding pattern map

The features are then computed using the pattern map by the Eqs. (2) (3) (4). Here the window size \(S1\) is 5, so the total feature number is \(24 \times 12 \times 5 = 1440\).

### 3.3. Adaboost and SVM classifier

The classifier is constructed as follows:

First Adaboost is used to select features instead of being a classifier. A subset of the features is selected by Adaboost from 1440 PPBTF features, and in the experiment of this paper 100 subset of features are selected. At each step, the chosen feature is uncorrelated with the output of the previous features. Because the selected features are less than the total features, the speed is improved.

SVM classifiers are trained on the features selected by AdaBoost. This paper uses 5-fold cross-validate scheme in the operation. The best SVM parameters are gained by the method, which is called grid search. RBF kernel function is used for the SVM in this paper. The main parameters of SVM are penalty C and Sigma (parameter of RBF kernel). To improve the speed, both of them are chosen between 25~220 in the experiment.

### 4. THE EXPERIMENT

#### 4.1. Experiment set up

First, 8988 optimum features are pre-selected from all 53928 rectangle features. Then 5000 scaled eye images and 7000 scaled non-eye images by the scale size \(24 \times 12\) are used as train samples for the first stage classifier. The remaining samples for each stage are generated according to the detecting result of the stages before the current one. Stages in the cascade are constructed using Adaboost.

Second, a facial image is classified by the 7-stage cascade classifier, and the result patches include eyes and some other patches such as eyebrows, mouth, nares, larger or smaller eyes are got. All these image patches are scaled to \(24 \times 12\), and then PPBTF is computed to the scaled patches. Using the features of PPBTF these patches are classified by the Adaboost and SVM classifier. Here the Adaboost and SVM classifier is trained using 4000 scaled image patches with 2000 eye patches and 2000 non-eye patches selected manually from the results of the cascade classifier.

#### 4.2. Old rectangle features VS New added features

In the experiment two cascade classifiers are built with old rectangle features in Fig.1 and old add some new features in Fig.2. For both the old and the new cascade classifier the same train samples and parameters are used, and each cascade classifier has seven stages.

In the pre-selected time for old feature classifier 8244 features are selected from 32976 features, and for new 8988 are select from 53928. After the classifiers are built one can see that for the old 1126 features are selected in the seven-stage cascade classifier and for the new only 832. This indicates that the new feature could use fewer features to do the detection than the old. 896 eye-open front facial images in FERET Database are selected and the detection rate are computed, for the old 0.97935, and for the new 0.97210, so the new added features can reduce the feature numbers used without influence the detection rate. For the detected non-eye patches the new features also show its advantages. With the 896 images, 2226 patches are mis-detected but for the old features 3655 patches mis-detected. (Fig.6)

Fig.6. Images in the first row are the detecting results by old features and in the second are the new features.
4.3. PPBTF VS LBP

After the rough detection by the new cascade classifier, an accurate detection is performed. Here two methods PPBTF and Local Binary Pattern (LBP) are used to train the Adaboost and SVM classifier, and a simple comparison has been carried out.

The 59-bin operator [9] is select, and divided the images into 4×4 pixels regions. Thus eye images (24×12) are divided into 18 regions, and represented by the LBP histograms with length of 944(59×18).

The experiments use a training set including 4000 samples, with 2000 eyes and 2000 non-eyes. For PPBTF, 100 features are selected using Adaboost from the 1440 features and used to form a classifier by SVM. For LBP also 100 features are selected from 944 features and used to form a classifier by SVM. The two classifiers are then used to classify a test set including 1091 samples with 302 eyes and 789 non-eyes. The recognition rates with PPBTF and LBP are 96.79% and 95.69% respectively. Experimental results illustrate that the proposed PPBTF is better than LBP.

4.4. The computer time of PPBTF

Operated on a computer of Pentium(R) 4 CPU 2.4GHz and 256M memory, the PPBTF method costs about 2.5ms to calculate the 100 features. And it just the number of the features used to classify the eyes using the Adaboost and SVM classifier. Owing to the fast operation the eye detecting system is remaining real time.

4.5. The experiment results

896 eye-open front facial images in FERET Database are selected to do the experiment. First, the images come through the cascade classifier, and eye and some non-eye patches are all got. Second, all patches are scaled and put into the Adaboost and SVM classifier, this process would exclude the non-eye patches. Some detection results are shown in Fig.7, compared with the result image in Fig.6, most of the non-eye patches are excluded by PPBTF except one in the last image (Fig.7), this is because the block is much the same like an eye in the pattern map Fig.7 (b).

Although there may be some mistakes like this (Fig.7 (b)), most of the patches could be classified, it indicates that the method proposed in this paper is feasible.

5. CONCLUSION

In this paper a new method for eye detection based on rectangle features and pixel-pattern-based texture feature (PPBTF) is proposed. The facial image is put into the cascade classifier using rectangle features, and the result patches of the classifier are scaled and transformed into a pattern map using PCA basis functions to form the features which will be classified by Adaboost and SVM classifier. The experiment results on FERET Database are excellent and the method is real time.

Future work using the proposed method would be training the classifier using eye closed or rotated samples and detecting the eye closed or rotated in the facial image.

6. REFERENCES