Learning Local Descriptors for Face Detection

Hongliang Jin[1,2], Qingshan Liu[1,2], Xiaoou Tang[1], Hanqing Lu[2]
[1] Department of Information Engineering, The Chinese University of Hong Kong
[2] National Lab of Pattern Recognition, Institute of Automation,Chinese Academy of Sciences
hljin, qsliu, luhq@nlpr.ia.ac.cn; xtang@ie.cuhk.edu.hk

Abstract

In this paper, we propose a realtime face detection approach based on local structure and texture of the objects in gray-level images. Our strategy is to map the local spatial structures and image textures of face class into binary patterns, and use these binary patterns as local descriptors. Boosting based face detector is constructed using these local descriptors, and cascade scheme is employed to further improve the efficiency of the face detector. Compared to the existing face detection approaches, our proposed method has two advantages: (1) it is robust to illumination changes to some extend, for the features use the information of local relationship instead of the original gray values; (2) the computational cost is very low, both in training procedure and evaluation step. The experimental results show that the proposed method can meet the demand of realtime applications with a satisfied detection performance.

1. Introduction

Face recognition is attracting more and more attention due to its potential applications. For an automatical face recognition system, face detection is the first and one of the most important steps. Many face detection approaches have been developed in recent years[1]. Representative approaches include the distribution-based model [8], support vector machine(SVM) [9], neural network [10] and the adaboost [5]. Sung et al [8] develop distribution models of face class and non-face class, and used a multilayer perceptron (MLP) classifier for classification. The SVM approach [9] aims to find an optimal hyperplane between the face and non-face class. The optimal hyperplane is defined by a weighted combination of a small set of the training samples. Rowley et al [10] present a set of neural network-based filters on an image, and used an arbitrator to combine the filters outputs. The most popular algorithm is the adaboost approach by Viola [5]. He uses simple harr-like features for face representation and adaboost to select features. Integral image and cascade structure are adopted to accelerate the face detector. This approach can meet the demand of realtime applications, but it often takes weeks of time training on a single PC.

In this paper, we propose a new realtime face detection approach based on the local structure and texture descriptors of the objects. Our strategy is to map the local structure and texture of face images to binary patterns, called the improved local binary patterns (ILBPs) [2], which successfully preserve the local information of the object. ILBP features have the virtues of simple, multiresolution, grayscale and rotation invariant. Taking a $3 * 3$ patch for example, it compares the pixels in the local area with the mean of this area and maps them to a binary pattern. Since ILBP features are relatively primitive compared to some holistic features [7], we use adaboost to combine the ILBP features to form a strong classifier. Cascade scheme is adopted to further improve the efficiency of the face detector.

The proposed approach has two advantages: (1) ILBP features are simple but efficient and discriminative. It is illumination invariant by nature. (2) The computation of ILBP features is very easy and fast. Because it only needs some ADD, $MINUS$, AND and OR operations, it costs less time than most face detection approaches. The training procedure could also be finished in a short time. We propose a realtime face detection algorithm. Our face detector can process a frame of $320 * 240$ in about 0.1 second using a PIV2.8 GHz PC(The speed can be further improved with coding optimization).

2. Local Texture and Structure Preserved Descriptors

2.1. Description of Improved LBP

Local binary pattern (LBP) is a texture operator with powerful discrimination. It can detect microstructures such as edges, lines, spots, flat areas, corners [3]. Compared to the original grayscale features, LBP features are invariant against monotonic transform of grayscale by definition. With a circular neighborhood P with radius R, we could compute the difference between the central pixel g_c and its neighborhood $\{g_0, ..., g_{P-1}\}$ to generate the operator $LBP_{P,R}$, which is given as follows:

0-7803-9332-5/05/$20.00 ©2005 IEEE

$$LBP_{P,R} = \sum_{i=0}^{P-1} s(g_i - g_c)2^i, \quad (1)$$

$$s(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{if } x \leq 0. \end{cases} \quad (2)$$

Though the original LBP features have great discriminative power, they can not deal with some special cases. For example, Figure 1.a contains two different microstructures, but the original LBP gives the same representation shown in Figure 1.b.

In this paper, we use the improved version of LBP in our previous work [2]. The definition of ILBP is shown as equation (3),(4) and (5). The central information of the structure is encoded in ILBP, which actually plays an important role in discriminating microstructures. So ILBP can represent $2^{P+1}-1$ different local structures, while original LBP produces $2^P - 1$. Figure 1.c gives the representations of the above two microstructures based on ILBP. We can see that the two microstructures are separated successfully.

$$ILBP_{P,R} = \sum_{i=0}^{P-1} s(g_i - m)2^i + s(g_c - m)2^P, \quad (3)$$

$$s(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{if } x \leq 0, \end{cases} \quad (4)$$

$$m = \frac{1}{P+1}\left(\sum_{i=0}^{P-1} g_i + g_c\right). \quad (5)$$

a. Original patch

X	20	X
20	20	20
X	20	X

X	1	X
1	20	20
X	20	X

b. Original LBP

X	0	X
0	*	0
X	0	X

X	0	X
0	*	0
X	0	X

c. Improved LBP

X	0	X
0	0	0
X	0	X

X	0	X
0	1	1
X	1	X

Figure 1: Comparison of $LBP_{4,1}$ and $ILBP_{4,1}$

2.2. Multi-Scale Strategy

ILBP also have the virtue of multiscale analysis. The performance could be enhanced if larger scale descriptors are introduced, since larger scale descriptors contain more information about the adjacent structure of texture. Of course, down-sample can also provide multiscale information about the structure of the texture, but the training samples in face detection are already small and down-sample will cause loss of the texture information. In this paper, we use different neighborhoods of ILBP features to extract multi-scale information. In a sense, the censes transform [6] can be seen as a special case of our multi-scale strategy.

In this paper, four variations of ILBP features are adopted. Figure 2 gives the illustration. We can see that $ILBP_{8,2_2}$ is a re-scale version of $ILBP_{8,1}$. $ILBP_{8,1}$ operator represents the local structure in a $3*3$ patch, and $ILBP_{8,2_2}$ expresses the relationship in a $5*5$ patch, which contains larger scale information. Certainly other ILBP features could also be introduced to express more detail texture information.

i =

7	0	1
6	8	2
5	4	3

ILBP 8,1

		0		
	7		1	
6		8		2
	5		3	
		4		

ILBP 8,2 1

7		0		1
6		8		2
5		4		3

ILBP 8,2 2

	7		0	
6				1
		8		
5				2
	4		3	

ILBP 8,2 3

Figure 2: $ILBP_{8,1}, ILBP_{8,2_1}, ILBP_{8,2_2}, ILBP_{8,2_3}$

3. Learning Local Descriptors of face Pattern

Based on ILBP, we get a large number of simple features. We construct a lookup table for each feature, and use adaboost to select a small set of discriminative features from them. Cascade structure is adopted to further enhance the performance of the face detector.

3.1. Learning the Cascade

Adaboost [4] is an efficient learning technique to combine weak hypotheses to form a strong classifier. At each step, training data is re-weighted so that misclassified training samples can get larger weights. By that, the next loop will focus on the misclassified samples in the last loop, and actually this maximizes margins between training samples. In this paper, we use adaboost to combine ILBP features to make a strong classifier for face detection.

In order to improve the computational performance of the face detector, cascade structure is adopted [5]. Cascade structure can speed up the face detector greatly with very small sacrifice in detection rate. The ILBP features are more discriminative than rectangle features in [5], for they contain detailed local structure information. We can use only 10 layers to obtain similar performance of [5] with 32 layers. In our experiment, the first layer has generally no more than twenty features. But it can reject most negative examples while retaining almost all the positives. The performance of each layer should meet the demand that the detection rate is no less than 99%. So the total detection rate is no less than $0.904(0.99^{10})$. We use bootstrap strategy to collect negatives [10]. The latter layers in cascade usually have more features than the former because the negatives get "harder" to be classified. The algorithm can be summarized as Table 1.

---

1. Given training samples (x_1,y_1),...,(x_n,y_n) where y_i=0,1 for negative and positive examples respectively.
2. Initialize weight of the training sample (x_i,y_i): $w_{t,i} = \frac{1}{2m}, \frac{1}{2l}$, for $y_i = 0, 1$ respectively, where m and l are the number of the negatives and positives respectively.
3. $t = 0$ and Loop
   - $t = t + 1$
   - Compute weighted error of all the weak classifiers under the distribution w_t (see table 2).
   - Select one weak classifier $\mathbf{p_t}$ with the lowest error ϵ_t and let $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$, $\alpha_t = -\frac{1}{2}ln\beta_t$
   - Update the distribution: $w_{t+1,i} = w_{t,i}\beta_t^{1-e_i}$ where $e_i = 0, 1$ for example x_i is classified correctly and incorrectly respectively.
   - Normalize the weights
     $w_{t+1,i} = \frac{w_{t+1,i}}{\sum w_{t+1,i}} \quad i = 1, ..., n$
   - Evaluate the performance using the obtained t features, if the performance meets the demand , then $T = t$ and break this loop
4. The final strong hypothesis of this layer is:
   $h(I) = \sum_{t=1}^{T} \alpha_t h_{\mathbf{p_t}}(ilbp_t) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t$
   where I is an image patch to be classified.
5. Refresh the negative samples using bootstrap and goto Step 2 to get another classifier to fill in the cascade until all the layers are obtained.

Table 1: Algorithm of AdaBoost

3.2. Weak Classifiers

In this subsection, we will describe how the weak classifiers are constructed. During each iteration in adaboost, we set up two statistic weighted histogram of each ILBP feature for positive and negative samples respectively using the sample weights in last loop. We count all the indexes and sum them up into bins. Then we use the established histograms to build a look-up table based on weighted majority vote. Each weak classifier in the final form will be a look-up table, which has $2^{P+1} - 1$ bins with only $1/0$, and the weighted error can be computed in the same way. We map the selected local structures into binary patterns for evaluation, and look it up in the look-up table to get the classification result. So it is very fast for evaluation. Details of the weak classifier is given in Table 2:

---

1. The distribution weight of the sample is w_i where $i = 1, ..., n$ is the sample label.
2. Create two weighted histograms for positives and negatives respectively.
   $\begin{cases} histp(\mathbf{p}, ilbp) = \sum_{x(\mathbf{p})=ilbp}^{y_i=1} w_{t,i} \\ histn(\mathbf{p}, ilbp) = \sum_{x(\mathbf{p})=ilbp}^{y_i=0} w_{t,i}. \end{cases}$
   where $histp, histn$ are the weighted histogram of positives and negatives respectively.
   $\mathbf{p} = \{a, b, c\}$ contains the local structure information. a, b are the coordinations of the ILBP feature, and c is the type of the ILBP feature used in this paper.
3. Calculate the error of this weak hypothesis:
   $\epsilon(\mathbf{p}) = \sum_{ilbp} min\{histp(\mathbf{p}, ilbp), histn(\mathbf{p}, ilbp)\}$
4. Finally the weak hypothesis is:
   $h_{\mathbf{p}}(ilbp) = \begin{cases} 1, & histp(\mathbf{p}, ilbp) \geq histn(\mathbf{p}, ilbp); \\ 0, & else. \end{cases}$
   where $ilbp$ is the mapped value of an improved local binary pattern $\mathbf{p}$.

Table 2: Computation of the weak classifier and error

4. Experiments

In our experiment, we use 5,000 face patches and their mirror images as positive samples, which are manually cropped by fixing the centers of the eyes. 10,000 non-face samples are randomly collected from natural scene images as negatives. The training samples are all resized to the size 24*24. The feature number of $ILBP_{8,1}$ is 484($22*22$) (not including the pixels on the border). The feature numbers of $ILBP_{8,2_1}$, $ILBP_{8,2_2}$, and $ILBP_{8,2_3}$ are 400(20*20) respectively. Therefore, the total number of features for selection is 1684. Keeping the positive samples, we use bootstrap method to recollect negative samples for each layer. We perform 10 times on more than 10,000 natural images containing no face with the size of 360*270. Training every layer in the cascade will use the false detected patches which pass through all the previous layers.

In evaluation step, one patch which passes all the layers in the cascade would be considered as face. In order to get face patches in all scales and positions, we scan each image using a fixed down-sample factor of 1.2. Multi-detections will usually appear around the true face position. We integrate the final results by averaging the adjacent results.

We investigate the performance of the proposed approach using two sets. One includes images chosen from the MIT-CMU set, which is widely used in the evaluation of face detection algorithms. There are 80 images containing 227 faces [7]. We compare our approach with the methods of Schneiderman-Kanade [11] and Liu [7]. The results are reported in Table 3. The evaluation of S-K which is publicly available at $http://www.vasc.ri.cmu.edu/cgi-bin/demos/findface.cgi$. There are two parameters, the frontal detection threshold and the profile detection threshold, which control the number of face detected and false alarms. We can see that we get comparable results on the MIT-CMU set to their approaches. But our approach is much faster than theirs, for we do not need any PCA projection, which is time consuming. Our approach only needs to map local structures of a patch to binary patterns, and then looks them up in the tables. Only some ADD, $MINUS$, AND, and OR operations are needed. Our face detector can process a frame of $320 * 240$ in about 0.1 second using a PIV2.8 GHz PC (The speed can be further improved with coding optimization).

method	face detected	false alarms	detection rate
S-K(1.0,1.0)	218	41	96.0%
S-K(2.0,2.0)	214	5	94.3%
S-K(3.0,3.0)	208	1	91.6%
BDF in [7]	221	1	97.4%
Our approach	211	3	93.0%

Table 3: Experimental Results on the MIT-CMU Set

In order to evaluate the robustness to illumination variation of the proposed method, we construct another testset from the YaleB face database, which consists of 10 subjects. Each subject has 576 images in (9 poses and 64 illumination conditions). We select 252 images, with the camera axis less than 50 degrees in azimuth and 50 degrees in elevation. We exclude the images with extreme illumination, since local structures of these images are lost. Thus we have 2520 images with pose and illumination variations. The results are shown in Table 4. We can see that our approach also achieves an encouraging performance under illumination variation.

face detected	false alarms	detection rate
2385	12	94.6%

Table 4: Experimental Results on the YaleB set

In our experiment, it should be mentioned that we could increase the detection rate and save computational time by using less layers in cascade, but this will also increase the false alarms and vice versa. There is a compromise between the detection rate, computational time, and false alarm rate.

5. Conclusion and discussion

In this paper, we use improved local binary pattern as descriptor in face detection instead of original grayscale value. Adaboost algorithm is used to select the local descriptors and cascade structure is adopted in our classifier to make it more efficient. Our method has two advantages. Firstly, it is robust to illumination change using ILBP features. Secondly, the computational cost of our face detector is very low, both in the training procedure and in evaluation. It can meet the demand of realtime applications. The experimental results show that our approach has encouraging performance.

Acknowledgments

This work described in this paper was fully supported by grants from the Research Grants Council of the Hong Kong SAR and a Hong Kong RGC/NSFC joint project N-CUHK 409/03 (NSFC No. 60318003), and partially supported by NSFC Grant No.60405005.

References

[1] M. Yang, D. Kriegman, and N. Ahuja: "Detecting faces in images: A survey" IEEE TPAMI 2002 vol.24(1), pp.34-58

[2] H. Jin, Q. Liu, H. Lu, and X. Tong: "Face Detection Using Improved LBP Under Bayesian Framework" ICIG 2004.

[3] T. Ojala, M. Pietikainen, and T. Maenpaa: "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns" IEEE TPAMI 2002 vol.24(7), pp.971-987

[4] R.E. Schapire, Y. Freund, and W.S. Lee: "Boosting the Margin: A New Explanation for the Effectiveness of Voting Methods" ICML 1997

[5] P. Viola and Michael J. Jones: "Robust real-time object detection" International Journal of Computer Vision, 2001

[6] B. Froba, and A. Ernst: "Face Detection with the Modified Census Transform" AFGR 2004

[7] C. Liu: "A Bayesian Discriminating Features Method for Face Detection" IEEE TPAMI 2003 vol.25(6), pp.725-740

[8] K-K. Sung, and T. Poggio: "Example-Based Learning for View-Based Human Face Detection" IEEE TPAMI 1998 vol.20(1), pp.39-51

[9] E. Osuna, R. Freund, and F. Girosi: "Training Support Vector Machines: an Application to Face Detection" CVPR 97

[10] Henry A. Rowley, S. Baluja, and T. Kanade: "Neural Network-Based Face Detection" IEEE TPAMI 1998 vol.20(1), pp. 22-38

[11] H. Schneiderman and T. Kanade: "Probabilistic Modeling of Local Appearance and Spatial Relationships for Object Recognition" CVPR 98.