LEARNING FEATURES FROM EXAMPLES FOR FACE DETECTION

Lu Xiaofeng, Zheng Songfeng, Zheng Nanning, Liu weixiang

The Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an, P.R.China 710049

ABSTRACT

In this paper, LPSVM algorithm are used to construct an over complete set of weak classifiers, and AdaBoost algorithm are adopted to select part of them to form a strong classifier. During the course of feature extraction and selection, the new method can minimize the classification error directly, whereas most previous works cannot do this. An important difference between this method and other methods is that the sparse features are learnt from the training set instead of being arbitrarily defined. Experiments demonstrate that the new algorithm performs well.

1. INTRODUCTION

Face detection has received an increasing amount of attention in the field of pattern recognition in recent years. It has direct relevance to the face recognition problem because locating faces in an unknown image is the first step of an autonomous human face recognition system. It also has potential applications in human-computer interfaces, census systems and surveillance systems, driver assistance systems, etc.

The current statistical features used to distinguish faces and nonfaces can be divided into two categories: local features [1,2,3] and global features [4,5,6]. Some previous global-feature-based face detectors [4,5,6] work very well for classifying frontal views of faces, but they are highly sensitive to translation and rotation of the face. Local-feature-based face detectors can avoid this problem by independently detecting parts of the face. For instance, the changes in the parts of the face are small compared to the changes in the whole face pattern for small rotations. So local features and global features are both important features.

When the non-zero components of a feature are not too much, we call it a sparse feature.

The number of non-zero components of a local feature is often largely smaller than the component number of the feature, so it is also a sparse feature. And when the number of non-zero components of a feature becomes larger, it evolves into a global feature. But we can still process it using the same model as the sparse feature. So although the non-zero components are not significantly less that the total number, we can consider it as a special case of sparse features.

In most previous works [1,2,3], spatial shapes of local features are subjectively defined instead of being learned from the training data set. So they use LNMF to learn part-based component, but there still exist some difficulties: 1) their method lacks a principled way for choosing the number of basis. 2) The selection of the basis cannot minimize the classification error in a direct way.

In order to minimize the classification error directly, and localize the features at the same time, we use Linear Support Vector Machine (LPSVM)[9] method to construct the over complete feature set. The LP SVM classifier is obtained only by solving a single system of linear equations in the usually small dimensional input space, so it is simple to implement .

Because of the good generalization of AdaBoost [8], we use it to select a small number of most discriminating features from the full feature set. The system has been compared with two face detection systems similar to the ones proposed in [1] and [2]

2. USING LPSVM TO EXTRACT SPARSE FEATURES[8]

Consider the problem of classifying l points in n-dimensional input space R, suppose the training data $(x_1, y_1),\ldots,(x_l, y_l)$, $x = (x^1,\ldots,x^n)^T \in R^n$, $y \in \{-1,+1\}$ can be separated by a hyper-plane: $w^T x_i + b = 0$,Where w is the weight and b is the bias of the hyper-plane, in this paper we consider them as a feature and a threshold, respectively. In order to make the features sparser, we should minimize the number of non-zero components of the feature, and at the same time minimize the classification error, so the optimal hyper-plane should satisfy the following condition:

$$\begin{aligned} &\min_{w,b} \quad \frac{1}{2}\|w\|_{L_0} + \frac{\nu}{2}\sum_{i=1}^{l} D(i)\xi_i^2 \\ &s.t. \quad y_i(w^T x_i + b) \geq 1 - \xi_i \end{aligned} \tag{1}$$

Where the L_0 norm of the feature counts the number of elements of the feature that are different from zero, and ν is a parameter that controls the trade-off between sparsity and the classification error, and $D(i)$ denotes the weight of the example x_i. and ξ_i are nonnegative slack variables. Observe that the smaller ν is in (1), the sparser the solution is.

Unfortunately, it can be shown that minimizing (1) is NP-hard because of the L_0 norm [10]. In order to circumvent this shortcoming, we use the L_2 norm as an approximation of the L_0 norm, and in order to further simplify the implementation, we modify (1) according to the main idea of LPSVM:

$$\min_{w,b} \ \frac{1}{2}\|w,b\|_{L_2} + \frac{\nu}{2}\sum_{i=1}^{l} D(i)\xi_i^2 \qquad (2)$$
$$s.t. \quad y_i(w^T x_i + b) = 1 - \xi_i$$

Substituting ξ_i into the objective function we get an unconstrained problem:

$$\min_{w,b} \frac{1}{2}(w^T w + b^2) + \frac{\nu}{2}\sum_{i=1}^{l} D(i)(1 - y_i(w^T x_i + b))^2 \qquad (3)$$

Set the gradient vector to zero, with respect to w, and b ,we have

$$\begin{cases} \nu A' DY(YD(Aw - eb) - Ye) + w = 0; \\ -\nu e' DY(YD(Aw - eb) - Ye) + b = 0; \end{cases} \qquad (4)$$

where e is a column vector of ones with arbitrary dimension, $A=[x_1, x_2, ..., x_l]^T$, $Y=Diag(y_1, y_2,...,y_l)$, $D=Diag(D(1),D(2)...,D(l))$.

After some simple algebra operations, we can get the following solution:

$$\begin{bmatrix} w \\ b \end{bmatrix} = (\frac{I}{\nu} + E'E)^{-1} E'YDe \qquad (5)$$

where $E=[DA \quad -De]$.

We can solve (5) directly by a linear system of $(n+1)$ variables where n is the number of attributes as well as the length of w.

In order to construct a overcomplete feature set for further selection in the AdaBoost step, we should train T weak LPSVM classifiers, and make each weak classifier as strong as possible. We adopt the follow steps to determine the T weak classifiers h_t:

For t=1,…,T

1 Set a working set to be empty;

2 Pick out two points randomly from the positive samples and the negative samples respectively, and add the two points into the working set;

3 Train a weak classifier on the working set using LPSVM algorithm;

4 Use this weak classifier to classify the whole data set, and add the correctly classified samples into the working set;

5 If the work set is not same as it was at last time, go to step 3;

End for

T should be large enough to make all the samples to be classified correctly at least once.

3. USING ADABOOST TO SELECT CLASSIFIERS

Obviously, using the full classifier set is infeasible in practice. So we use AdaBoost algorithm both to select some most discriminating weak classifiers from the full classifiers set and to train the final strong classifier. In its original form, the goal of AdaBoost is to get the good performance by combining a collection of weak classifiers to form a very effective classifier. In fact, a very small number of these classifiers are needed in our system, because the LPSVM weak classifiers we used are stronger than the weak classifiers described in [1] and [2].

Given training examples: $(x_1, y_1),\ldots, (x_N, y_N)$, where x_i represents image patterns and y_i = 1,-1 represent the labels for faces and nonfaces samples respectively. And N=m+n, where m and n represent the number of positives and negatives respectively.

The method of selecting weak classifiers and determining the final classifier can be described as following:

Step1. Initialize $D_t(i)=1/(2m),1/(2n)$ for y_i=1,-1, respectively.

Step2. for $t=1,...M,$:

1) Train T weak LPSVM classifiers to construct an overcomplete feature set using the method presented in section 2.

2) Choose the classifier h_t with the smallest error

$$\varepsilon_t = 0.5\sum_i D_t(i)\,|\,h_t(x_i) - y_i\,| \qquad (6)$$

and choose $\alpha_t = 0.5\ln\left((1-\varepsilon_t)/\varepsilon_t\right) > 0$ (7)

3). Update $D_t(i)$ as follows:

$$D_t(i) \to D_t(i)\exp(-\alpha_t y_i h_t(x_i)) \qquad (8)$$

and normalize them to $\sum_i D_t(i) = 1$, so that $D_t(i)$ will be a distribution.

Step 3. Output the final classifier:

$$H_{\text{final}}(x) = \text{sgn}\left(\sum_{t=1}^{M} \alpha_t h_t(x)\right) \qquad (9)$$

4.EXPERIMENTAL RESULTS

Training set and testing set preparation, speeding up method and performance comparison are given in this section.

4.1. Training Set and Testing Set Preparation.

Both the face training set and the face testing set are taken randomly from 2000 images downloaded from the World Wide Web. We first got 1,200 frontal face patches, each resized into 19×19, and processed

by masking an oval within the face rectangle, light correction and histogram equalization, and then the dataset is split into two subsets: 1000 face patches are used to create the face training set and 200 are used to create the face testing set. Face patches in the face training set were then aligned by the center point of the two eyes and the horizontal distance between the two eyes. In order to enlarge the training set, left-right flipped versions and slightly in-plane-rotated versions of the original faces were added into the original training set, so in total we got 6000 face samples.

1000 images containing no face were also collected from the Internet. 16,000 nonface patches were extracted from the images by randomly selecting a patch from an image and resized into 19×19. We randomly chosen 4000 of them to construct the nonface testing set, and the remains were taken as nonface training set.

Using the method proposed in the previous sections, we constructed a classifier set including about 2000 weak classifiers.

The first feature selected (see Figure 1) seems to focus on the property of a face that the region of the eyes is often darker than the surrounding region.

Fig. 1. The first feature selected by AdaBoost . The left image shows positive components of the weight *w*, while the right image shows the absolute value of the negative components.

4.2. Image Scanning and Face Detection

Our system detects faces by exhaustively scanning over the image for face-like patterns at many possible scales, so an image pyramid is constructed as Rowley did in [3]. The initial scale is 1.0,and the scale step is 1.2. We divide the original image into overlapping 19 by 19 pixels sub-images, and then classify them using our trained classifier to determine whether a face is present or not.

In order to detect face pattern correctly, we must preprocess each patch. First, in order to prune the useless background pixels, we try to mask an oval within the face rectangle. Second, we attempt to correct for unidirectional lighting effects as suggest- ed by [3] by fitting a single linear shading plane to the image and the plane can be subtracted out of the original image. Once the lighting direction is corrected for, the grayscale histogram can then be rescaled to span the minimum and maximum grayscale levels.

Results from all scales and locations are merged to get the final result. Figure 2 shows the output of our face detector on some test images. Although there are several detected faces around each face, only the window with the largest number of multiple detections is drawn to enclose each detected faces for clear presentation.

4.3. Speed up the detection process.

4.3.1 Classifier Cascade[1]

Usually most patches in an image do not include faces and they can be easily classified using less classifiers. So we use the similar feature selection framework with Viola's cascade method. A 30 layer cascaded detector was trained to detect frontal upright faces. And we use 1, 6, 22, 230 features in the 1^{st} , 2^{nd} ,3^{rd} ,20^{th} layer, respectively.

4.3.2. 2D-FFT[12]

To further reduce the computational complexity, we use 2D-FFT skill proposed by Yacoub[12] to efficiently calculate the activity of the classifier in the first layer. Since the activity of the first classifier over the whole image *I* can be formulated as follow:

$$h_1(I) = \mathrm{sgn}\left(w_1^{\,T} \otimes I + b_1\right) \qquad (10)$$

Where w_1 is the first sparse character, and $\otimes$ represents a cross-correlation operation. The first layer of our detector can process a 396 by 260 pixel image in about only 0.095 seconds, which is a little bit slower than Viola-Jones detector, but can reach a higher accuracy.

Table 1: detection rates for various numbers of false positives on the MIT+CMU

detector / False detections	Harr-like	LNMF	LPSVM
10	**75.2%**	**75.8%**	**77.3%**
28	**85.3%**	**84.7%**	**87.6%**
50	**90.1%**	**89.5%**	**91.7%**
62	**91.1%**	**91.4%**	**92.3%**
77	**91.2%**	**92.0%**	**93.1%**
95	**91.7%**	**92.1%**	**93.1%**
180	**92.8%**	**92.2%**	**93.2%**
300	**93.3%**	**92.5%**	**94.1%**

4.4 Performance Comparison

We tested our system on the images collected from the MIT+CMU test set. The harr-like feature based detection system proposed in [1] and the LNMF feature based detection system proposed in [2] were also tested for comparison. The results on this test set are shown in table 1 using the ROC (Receiver Operator Characteristic) of the final layer classifiers (500 features, for all detectors). To create the ROC the threshold is adjusted from $-\infty$ to $+\infty$.

Figure 3 shows another comparison, the images were downloaded from MIT CBCL, the training set consists of 6977 images (2429 faces and 4548 nonfaces), and the test set consists of 24,045 images

(472 faces and 23573 nonfaces).

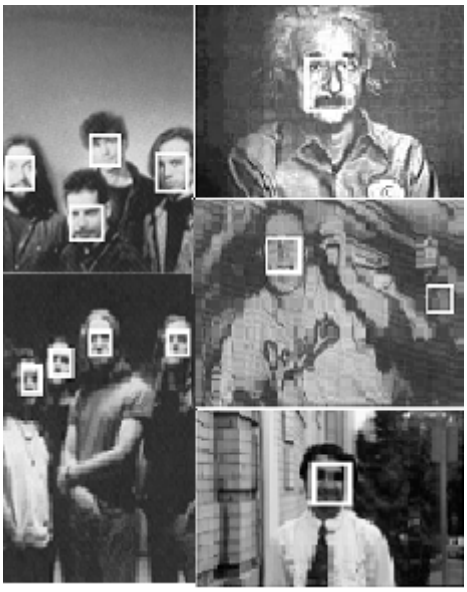

Fig.2. The output of our face detector on some test images

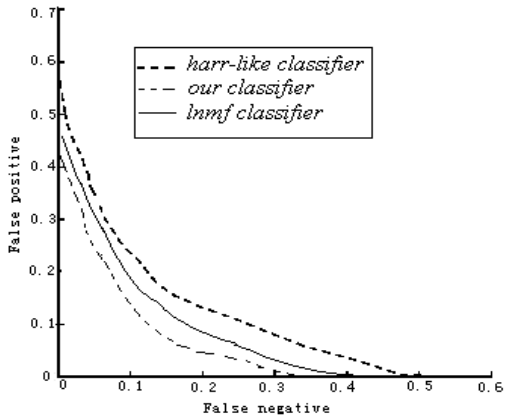

Fig. 3. ROC curves for comparison

5. CONCLUSION

We have presented a novel face detection method, which uses local features learnt from the training set instead of being arbitrarily defined. And during the course of feature extraction and selection, it can minimize the classification error directly, whereas most previous works cannot do this. Another character of the LPSVM feature extraction is that tuning the parameter ν in the algorithm will control the trade off between sparsity and the classification error. The performance of our face detection system is compared with the harr-like feature based detection system and the LNMF feature based detection system. Experimental result shows that the former is superior to the latter in the accuracy.

6. REFERENCES

[1] P. Viola and M.J. Jones, "Robust real-time object detection, technical report series," Compaq Cambridge research laboratory, CRL 2001 01,Feb. 2001.

[2] L.Gu, S.Z. Li, H.J. Zhang. "Learning Probabilistic Distribution Model For Multi-View Face Detection,"In Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition. Hawaii. December, 2001

[3] H. A. Rowley, S. Baluja, and T. Kanade. "Neural network-based face detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(1): 23–38, 1998.

[4] E.Osuna, R.Freund, F,Griosi, "Training support vector machine: an application to face detection," Proceedings of the CVPR (1997) 130-136.

[5] K .K. Sung and T. Poggio. "Example-based Learning for View-based Human Face Detection," IEEE PAMI.,20(1),39-51,1998.

[6] B. Moghaddam and A. Pentland. "Probabilistic visual learning for object detection," Technical Report 326,MIT Media Laboratory. June 1995.

[7] S.E.Palmer, "Hierarchical structure in perceptual representation," Cogn. Psychol. 9,441 -474,1977.

[8] Glenn Fung and Olvi L.Mangasarian. "Proximal support vector machine classifiers." Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Aug 26-29, 2001.

[9] Y. Freund and R.E.Schapire. "A decision- theoretic generalization of noline learning and an application to boosting." Computational learning theory: Eurocolt'95 pp23-37,1995.

[10] O.Chapelle and V.Vapnik. "Model selection for support vector machines." In Advances in Neural Information Processing Systems, 1999

[11] Stan Z.Li, Long Zhu, ZhenQiu Zhang. "Statistical Learning of Multi-View Face Detection." In Proceedings of The 7th European Conference on Computer Vision. Copenhagen, Denmark. May, 2002

[12] S. Ben-Yacoub, B. Fasel, and J. Luettin, Fast Face Detection using MLP and FFT, Proceedings of the Second International Conference on Audio and Video-based Biometric Person Authentication (AVBPA'99)", 1999