A Theoretical Approach to Construct Highly Discriminative Features with Application in AdaBoost

Yuxin Jin, Linmi Tao, Guangyou Xu, and Yuxin Peng

Computer Science and Technology Department, Tsinghua University, Beijing, China
jyx05@mails.tsinghua.edu.cn

Abstract. AdaBoost is a practical method of real-time face detection, but abides by a crucial problem of overfitting for the big number of features used in a trained classifier due to the weak discriminative abilities of these features. This paper proposes a theoretical approach to construct highly discriminative features, which is named composed features, from Haar-like features. Both of the composed and Haar-like features are employed to train a multi-view face detector. The primary experiments show promising results in reducing the number of features used in a classifier, which leads to the increase of the generalization ability of the classifier.

1 Introduction

In 1995, Freund and Schapire [1] introduced AdaBoost algorithm based on traditional boosting method. Thanks to their efforts, theoretical analysis on AdaBoost was pro-posed in the following years. They proved in [1] that the generalization error should be smaller if fewer training rounds are involved. Later, they gave a new theory of the generalization in terms of margins [2]: greater margins contribute to better results.

In early years, AdaBoost was, however, inapplicable in real-time case due to its great computational cost. Fortunately, the breakthrough occurred in 2001 when Viola and Jones proposed a novel real-time AdaBoost for face detection [4]. The keys to make real-time possible are the usage of the Integral Image, Haar-like features and Cascade Hierarchy. Based on this approach, two kinds of extensions were focused on the improvement of hierarchy and feature.

[12] extended the cascade hierarchy into the multi-view case - Detector Pyramid Architecture AdaBoost (DPAA). Later, [7] adopted Width-First- Search (WFS) Tree Structure to make a balance between high speed and robust.

[11] extended the Haar-like features with 45°rotated features. [8] proposed Asymmetric Rectangle Features and experiment showed improved performance.

However, the above methods are based on Haar-like features which are so weak that a large number of weak classifiers are used to train a strong classifier. Proven in [1], such a burdensome strong classifier increases the risk of overfitting.

Some efforts were taken to overcome the disadvantage of Haar-like features (their poor discriminative abilities). [9] used PCA approach to generate the

Y. Yagi et al. (Eds.): ACCV 2007, Part I, LNCS 4843, pp. 748–757, 2007.

global features which are included in the feature set in later layers of cascade. [10] used Gabor Features instead of Haar-like features. Although these features show superior discriminative abilities to Haar-like features, they are time-consuming in computation which may forbid real-time application. [14] used EOH (Edge Oriented Histogram) Features and gained good results.

In this paper, we propose a novel theoretical approach to construct highly discriminative features named composed features from Haar-like features whose computational load is small suitable for real-time tasks such as face detection. Thus, we can not only efficiently compute the highly discriminative features but also decrease overfitting.

In section 2, we will discuss features and their discriminative abilities. In section 3, highly discriminative features efficient in computation will be constructed by Haar-like features. Section 4 shows the promising experiment results.

2 Features

For AdaBoost, a strong classifier combined by many weak classifiers which are not much better than random guess can achieve any little error rate in training after sufficient loop rounds which has been proven in [1].

Each weak classifier contains two parts: feature and classification function. AdaBoost systematically chooses different features and then builds weak classifiers based on them and finally outputs a strong classifier. We will focus on the features.

Currently, in order to be real-time, Viola and Jones [4] introduced Haar-like features. With the help of Integral Image, they are efficient to compute. Only about six additions are involved to compute their value. Moreover, they are in essence linear features because their value can also be calculated as the dot product of the feature and the image vector. Fig. 1 show the vector representation.

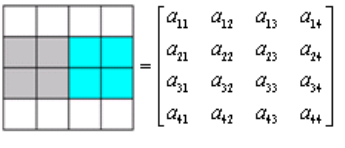


Fig. 1. Vector representation of Haar-like feature

Image could be represented as:

$image = [a_{11}, a_{21}, a_{31}, a_{41}, ...a_{34}, a_{44}]^T$

Similarly, the Haar-like feature also could be represented in vector form:

$\mathbf{w} = [0, -1, -1, 0, 0, -1, -1, 0, 0, +1, +1, 0, 0, +1, +1, 0]^T$

Feature value x_w is just the dot product of these two vectors. (The module of $\mathbf{w}$ does not affect the discriminative ability.)

$$x_w = \mathbf{w} \cdot image = \mathbf{w}^T image \tag{1}$$

Unlike many known linear features which are highly discriminative such as PCA (Principal Component Analysis), LDA (Linear Discriminant Analysis), Gabor features etc, a single Haar-like feature is so simple that its discrimination ability is weak. With respect to the discriminative abilities, features can be categorized into two classes: weak features and strong features shown in Fig. 2. *Categorization is not strictly defined. Strong features only mean the features with relatively higher discrimination abilities (or the classification error may be lower than some threshold).*

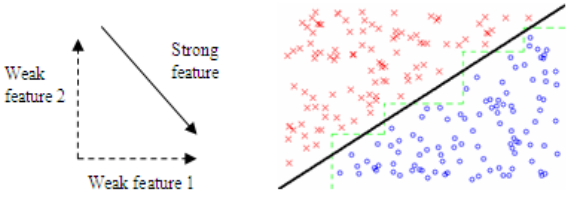


Fig. 2. A 2-Class case. Left: 2 weak feature vectors' direction and 1 strong feature vector's direction. Right: dash lines: the separation plane in weak feature space; solid line: the separation plane in strong feature space.

In this example, two classes are drawn in circles and crosses. Suppose there are only two weak features whose directions are horizontal and vertical. In order to fully separate these two classes, about eight weak classifiers should be trained by AdaBoost. However, only one weak classifier with the strong feature can fully separate them shown in Fig. 2. As to the generalization theory in [1], the strong classifier combined with fewer weak classifiers performs better.

Although strong features are superior to weak features in discriminative abilities, they are always time-consuming in computation which involves n multiplications for an n dimensional image vector. That is exactly the reason to keep them from application in real-time cases.

In the example illustrated by Fig. 2, all features are vectors. The strong feature can be constructed by two weak features $\mathbf{f}_{W1}$ and $\mathbf{f}_{W2}$:$\mathbf{f}_S = \alpha\mathbf{f}_{W1} + \beta\mathbf{f}_{W2}$. More generally, any linear feature vector in can be constructed by at most linearly independent vectors (linear features) in . In light of this, we will propose a novel approach to reduce the computational load of strong feature by constructing strong features from Haar-like features.

3 Composed Features

3.1 Definition of Composed Features

The computation cost of a strong feature value arises from the numerous multiplications of dot product between feature vector and image vector. Thus, in order to reduce such cost, the number of multiplications should be decreased. We can use Haar-like features to construct a strong feature. Due to the low computation

cost for Haar-like features' value, strong feature's value can be calculated fast. A strong feature is de-noted as vector $\mathbf{f}_j$ in R^d.

For a 16*16 image, there exist more than 50,000 Haar-like features. These features are definitely linearly dependent because the dimension is just 256 far fewer than the number of features. Actually, there are just $M * N$ linearly independent feature vectors for $M * N$ image size. We define one group of $M * N$ independent feature vectors from the set of Haar-like features as base features.

$\mathbf{w}_0, \mathbf{w}_1, \mathbf{w}_2, ...\mathbf{w}_{d-1}, d = M \times N$

By using these base features, $\mathbf{f}_j$ can be constructed as follows:

$$\mathbf{f}_j = \sum_{i=0}^{d-1} p_{ij}\mathbf{w}_i \tag{2}$$

Then the feature value can be computed as:

$$x_j = \sum_{i=0}^{d-1} p_{ij}\mathbf{w}_i^T image \tag{3}$$

In this way, we can compute the strong feature's value. However, such representation requires d multiplications, too. To reduce the computing time, we want to use fewer Haar-like features to construct a strong feature.

We choose k linearly independent Haar-like features $\mathbf{w}_0, ...\mathbf{w}_{k-1} \in W, k \ll d$ to construct a feature $\mathbf{q}_j$, W is the complete set of Haar-like features, then:

$$\mathbf{q}_j = \sum_{i=0}^{k-1} q_{ij}\mathbf{w}_i = \mathbf{W}\alpha_q, \mathbf{W} = [\mathbf{w}_0, ...\mathbf{w}_{k-1}], \alpha_q = [q_{0j}, ...q_{k-1j}]^T \tag{4}$$

We define such features as Composed Features:

Definition 1. *Composed features: linear and constructed by some linearly independent features - base features so that the computation of the features' value can be implemented indirectly by calculating the base features' value. Usually, the computational cost is reduced.*

3.2 Approximation Measurement

Composed features are not necessary to be strong features. Some of them may be strong features while others may not. We have two ways to find strong features:

Firstly, we can exhaustively search a proper set of to construct a com-posed feature and then check its discrimination ability. If the classification error is less than some predetermined threshold, it can be viewed as a strong feature.

Secondly, as some strong features (PCA, LDA, or Gabor) are known, we can con-struct a composed feature to approximate these features.

The first way would be feasible only if k is very small(just 2-3). Making exhaustive exploration is equivalent to making comparison among all the possible combination of all Haar-like features. Even fixing the coefficients, eg. $\alpha_q = [1, 1]^T$,

the searching space will still be $C(|W|, k)$, where $|W|$ is the number of Haar-like features. When $k > 3$, the computational cost is too large:$C(50000, 4) \approx 2.6e^{17}$. Such a naïve approach will crash when k is a bit larger.

The second way is more practical. In this case, $\mathbf{f}_j$ is assumed to be known(they can be PCA, LDA, or Gabor features). Our task is to find a proper set of $\mathbf{w}_0, ...\mathbf{w}_{k-1}$ and construct a composed feature which can approximate $\mathbf{f}_j$. Thus, a measurement to evaluate the approximation should be introduced as follows:

Definition 2. *Approximation Measurement: The smaller the angle θ between vector $\mathbf{f}_j$ and $\mathbf{q}_j$ is, the better $\mathbf{q}_j$ approximates $\mathbf{f}_j$.*

Thus, given set $\mathbf{W}$, the coefficients α_q can be uniquely determined according to the Approximation Measurement. It is equivalent to maximize $\cos\theta$.

$$\cos\theta = \frac{\mathbf{q}_j \cdot \mathbf{f}_j}{|\mathbf{q}_j||\mathbf{f}_j|} = \frac{\sum_{i=0}^{k-1} q_{ij}\mathbf{w}_i^T\mathbf{f}_j}{\sqrt{\sum_{i=0}^{k-1} q_{ij}^2\mathbf{w}_i^T\mathbf{w}_i}|\mathbf{f}_j|} = \frac{\alpha_q^T\mathbf{F}_W}{\sqrt{\alpha_q^T\mathbf{W}^T\mathbf{W}\alpha_q}|\mathbf{f}_j|} \tag{5}$$

where $\mathbf{F}_W = [\mathbf{w}_0^T\mathbf{f}_j, ..., \mathbf{w}_{k-1}^T\mathbf{f}_j]^T$

Then the problem becomes solving the maximum problem given below.

$$\begin{cases} \max \alpha_q^T\mathbf{F}_W \\ s.t. \quad \alpha_q^T\mathbf{W}^T\mathbf{W}\alpha_q = 1 \end{cases} \tag{6}$$

Here, we restrict the composed vector $\mathbf{q}_j$ to be unit vector without losing generosity. Such problem could be easily solved by implementing Lagrangian method.

$$f(p_{0j}, p_{1j}, ..., p_{k-1j}, \lambda) = \alpha_q^T\mathbf{F}_W - \lambda(\alpha_q^T\mathbf{W}^T\mathbf{W}\alpha_q - 1) \tag{7}$$

$$\begin{cases} \lambda = \dfrac{\sqrt{\mathbf{F}_W^T(\mathbf{W}^T\mathbf{W})^{-1}\mathbf{F}_W}}{2} \\ \alpha_q = \dfrac{(\mathbf{W}^T\mathbf{W})^{-1}\mathbf{F}_W}{\sqrt{\mathbf{F}_W^T(\mathbf{W}^T\mathbf{W})^{-1}\mathbf{F}_W}} \end{cases} \tag{8}$$

As selected Haar-like features are linearly independent, $\mathbf{W}^T\mathbf{W}$ is invertible.

The remaining problem is how to find a proper set W. It is infeasible to implement an exhaustive search, therefore, we will introduce a novel algorithm based on Simulated Annealing to achieve it.

3.3 Proper W Searching Algorithm

To find a proper set W is an optimization problem described as follows:

$$\begin{cases} \min\theta = \min f(\mathbf{W}) \text{ for given } k \\ s.t. \quad \mathbf{W} \in D \end{cases} \tag{9}$$

D is the configuration space. Each configuration is a set of k Haar-like features chosen from W. Because the number of Haar-like features denoted as $|W|$ is finite, so that the number of configurations in D ($|D|$) is finite. However, it is an NP problem because $|D| = C(|W|, k), k \ll d$.

In order to find a proper set of Haar-like features, Simulated Annealing algorithm is implemented. We take θ as the energy function, if current θ_i is larger than θ_j calculated from configuration through (5), i will be transited into j, otherwise the transition occurs at some probability.

From the current configuration i to the next one j, $i, j \in D$, we only exchange one Haar-like feature in i with the transition probability.

$$p_{ij} = G_{ij}(t)A_{ij}(t), \quad \forall j \in D \tag{10}$$

Where t is the temperature. $G_{ij}(t)$ and $A_{ij}(t)$ are Generalization and Acceptance probability, respectively.

$$G_{ij}(t) = \begin{cases} 1/|N(i)|, & j \in N(i) \\ 0, & j \notin N(i) \end{cases} \tag{11}$$

$$A_{ij}(t) = \begin{cases} 1, & f(i) \geq f(j) \\ \exp -\frac{f(j)-f(i)}{t}, & f(i) < f(j) \end{cases} \tag{12}$$

$N(i)$ is the neighbor of configuration i. Only one feature by randomly selection in i can be exchanged with any other feature in W which is linearly independent with the remained features. Thus, the neighbor number of i is $k|W| \ll |D|$. In this way, the problem is feasible for solving. The Proper Searching Algorithm is given in Fig. 3.

Research in feature extraction lasts for several decades. Thanks to these efforts, PCA, LDA, Gabor features or other possible linear features can be used

Step 1: Randomly choose an Initial configuration $\mathbf{W}^{(s)}$, $s = 0$, $t = t_0$

Step 2: Current configuration $\mathbf{W}^{(s)} = [w_{(0)}^{(s)}, \ldots w_{(k-1)}^{(s)}]$, $w_{(0)}^{(s)}, \ldots w_{(k-1)}^{(s)} \in W$

Randomly choose i from $\{0, 1, \ldots, k-1\}$

Choose j from $\{0, 1, \ldots, |W|-1\}$, satisfying linear independent

let $\mathbf{W}' = [w_{(0)}^{(s)} \ldots w_{(i-1)}^{(s)}, w_j, w_{(i+1)}^{(s)}, \ldots, w_{(k-1)}^{(s)}]$

$\Delta f = f(\mathbf{W}') - f(\mathbf{W}^{(s)})$

if $\Delta f \leq 0$ or $\exp(-\Delta f / t) > rand(0,1)$, goto Step 3;

else repeat Step 2;

Step 3: $s := s+1$, $\mathbf{W}^{(s)} = \mathbf{W}'$

if $s \bmod ck = 0$, $t := \frac{K - s/ck}{K} t$, c is a constant

if $t > \varepsilon_t$ *or* $f(\mathbf{W}^{(s)}) > \varepsilon_g$, goto Step 2,

ε_t and ε_g are sufficiently small constant

else end procedure;

Fig. 3. Proper Searching Algorithm

here as $\mathbf{f}_j$. With the construction, complicated strong features are composed of several simple Haar-like features. As a result, the computation of strong features' value is faster with some insignificant loss in accuracy. Then the Composed Features can be used in AdaBoost for real-time application.

4 Experiment

A total of 10,000 faces are collected from variant sources and categorized into 5 views, [-90, -60], [-60, -15], [-15, 15], [15, 60] and [60, 90] (view 1∼5) with 2000 of each view. Each face example's size is 16*16. We adopt DPAA ([12], [8]) as Train-ing Structure under AdaBoost Scheme. The ratio of the number of positive examples to the number of negative examples is 1.0 for each layer.

We implement our experiments by using a P4 3.0GHZ, 512 RAM computer.

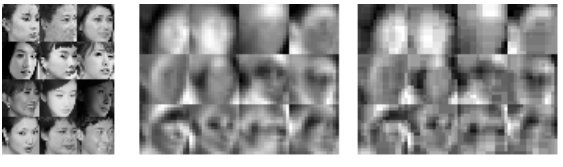


Fig. 4. Left: faces data in view 4; Center: extracted PCA features of group 4; Right: approximated PCA features

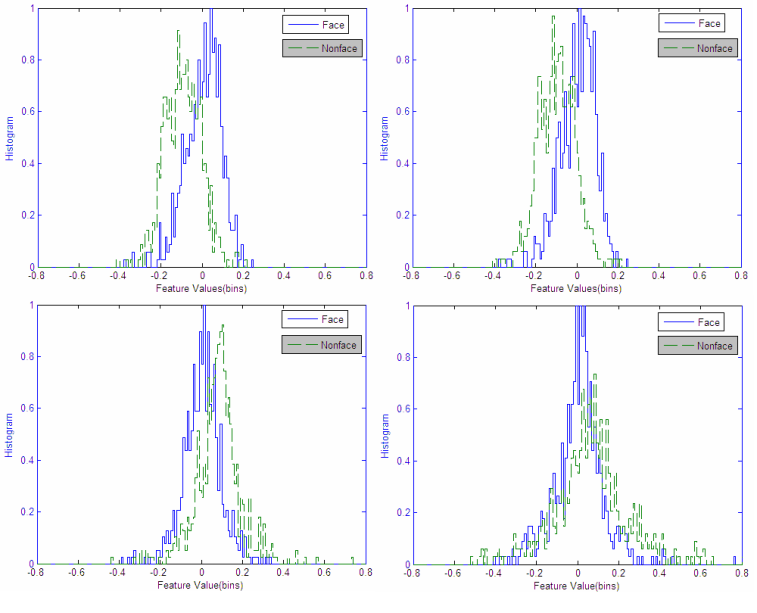


Fig. 5. Feature values' distribution for features. Top-Left: PCA feature. Bottom-Right PCA- CF. Bottom-Left: 1st Haar-like features. Bottom-Right: 5th Haar-like features.

In our experiment, we only construct PCA features. We extract PCA from 9 groups, each of which may include one or more views data. Groups 1 5 include 5 views data respectively; group 6 include view 1 and 2; group 7 include view 2, 3 and 4; group 8 include view 4 and 5; group 9 include all views. We choose first 100 PCA features from each group to be approximated.

Fig. 4 shows the PCA features and PCA-CFs (PCA Composed Features) each of which is combined with 15 Haar-like features for view [30, 60]. The angles between PCA features and PCA-CFs are less than $20^o (\cos 20^o \approx 0.939693)$.

In Fig. 5, top row shows that in layer 7 the feature values' distribution is similar for the PCA feature and PCA-CF. Their error rates are 26.75% and 27.58% respectively. Obviously, although PCA-CF is not exact PCA features, their discrimination abilities are similar. In layer 7, it is difficult to depart faces from nonfaces. Bottom row shows the distribution for the 1st and 5th Haar-like features selected with error 30.17% and 38.17%. They are worse than PCA-CF.

We tested our system on the CMU profile data set (208 images and 441 faces). Fig. 6 are the comparison of training error curves and margin distribution in the test. As to the results, the training error rate converges faster to zero if PCA-CFs are used. Furthermore, fewer features are selected and higher margins are gained. Generalization error's up bound must be reduced according to [1].

Scaling the scanning window from 16*16 to 256*256 with scale ratio 1.2, we can process about 14 frames per second for 320*240 images on average. Some of the results are given in Fig. 7. We compare ROC curve of our approach with those

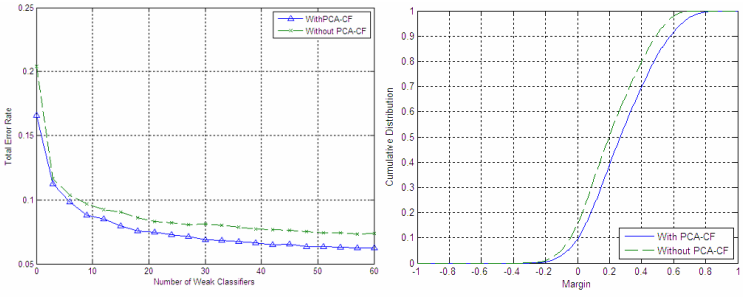


Fig. 6. Left: Training Error Curve; Right: Margin distribution

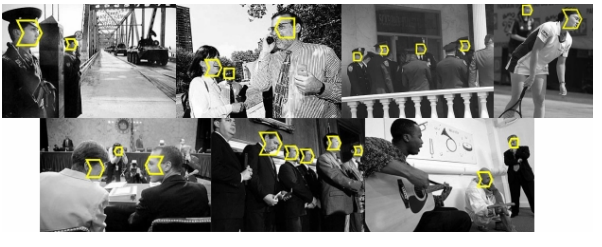
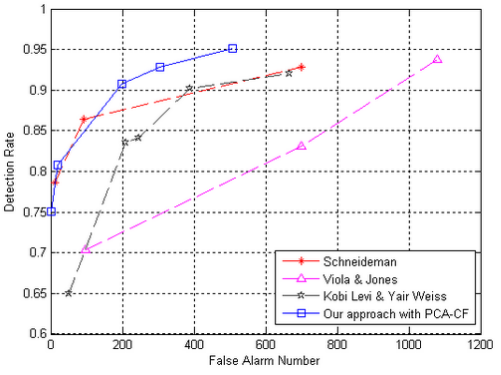


Fig. 7. Some results on CMU profile test set



Approach	Detection Rate	False Alarm #
Schneideman	78.60%	12
	86.40%	91
	92.80%	700
Viola & Jones	70.40%	98
	83.10%	700
	about 93.7%	about 1080
Levi & Weiss	84.10%	246
	90.20%	389
	about 92.10%	about 660
Our approach with PCA-CF	80.80%	20
	92.80%	306
	95.10%	507

Fig. 8. ROC comparison on CMU profile test set

pro-posed by Viola, Jones [5] and Schneideman [6] (these approaches reported results in CMU profile data set), the curves are shown in Fig. 8. Apparently, our approach is the best among them and it is real-time. (The results in [8] were given on some unknown dataset. [9][10] gave results on other dataset. [11] only gave the false alarm rate instead of false alarm number).

5 Conclusion

In this paper, we propose a theoretical approach to construct linear strong features so that we improve generalization ability and efficiency.

Haar-like features are too weak to discriminate classes, which results in serious overfitting. Strong features may be used to gain highly discriminative ability but they are inefficient in computation.

Composed features proposed in this paper inherit advantages from both Haar-like features and strong features. By using Proper W Searching Algorithm, composed features can be constructed and approximate strong features so that efficiency and better generalization ability could be achieved.

Experiment shows our method is better than Viola, Jones, Schneiderman and Levi, Weiss. We can build up real-time AdaBoost system on composed features.

Our approach could be extended to construct any linear strong features. This approach's application should not be limited in AdaBoost, it can be used in any cases where need fast computation of some strong features.

Acknowledgements

This work is supported by National Science Foundation of China under grant No 60673189 and No 60433030.

References

1. Freund, Y., Schapire, R.E.: A Decision- Theoretic Generalization of On-Line Learning and an Application to Boosting. Journal of Computer and System Science 55, 119–139 (1997)
2. Schapire, R., Freund, Y., Bartlett, P., Lee, W.S.: Boosting the Margin: A New Explanation for the Effectiveness of Voting Methods. The Annals of Statistics 26(5), 1651–1686 (1998)
3. Schapire, R., Singer, Y.: Improved Boosting Algorithms Using Confidence-rated Predictions. Machine Learning 37, 297–336 (1999)
4. Viola, P., Jones, M.: Rapid Object Detection Using a Boosted Cascade of Simple Features. In: IEEE Conf. Computer Vision and Pattern Recognition, IEEE Computer Society Press, Los Alamitos (2001)
5. Jones, M., Viola, P.: Fast Multi-view Face Detection. In: TR (2003)
6. Schneiderman, H., Kanade, T.: A statistical method for 3D object detection applied to faces and cars. In: IEEE Conf. Computer Vision and Pattern Recognition, IEEE Computer Society Press, Los Alamitos (2000)
7. Huang, C., Ai, H., et al.: Vector Boosting for Rotation Invariant Multi-view Face Detection. In: IEEE International Conf. on Computer Vision (2005)
8. Wang, Y., Liu, Y., Tao, L., Xu, G.: Real-Time Multi-View Face Detection and Pose Estimation in Video Stream. In: IEEE International Conf. on Pattern Recognition (2006)
9. Zhang, D., Li, S.Z., et al.: Real-Time Face Detection Using Boosting in Hierarchical Feature Spaces. In: IEEE International Conf. on Pattern Recognition, IEEE Computer Society Press, Los Alamitos (2004)
10. Yang, P., Shan, S., et al.: Dong Zhang: Face Recognition Using Ada-Boosted Garbor Features. In: IEEE International Conf. on Automatic Face and Gesture Recognition (2004)
11. Lienhart, R., Maydt, J.: An Extended Set of Haar-like Features for Rapid Object Detection. In: IEEE International Conf. on Image Processing (2002)
12. Li, S.Z., Zhu, L., et al.: Statistical Learning of Multi-View Face Detection. In: Tistarelli, M., Bigun, J., Jain, A.K. (eds.) ECCV 2002. LNCS, vol. 2359, Springer, Heidelberg (2002)
13. Fleuret, F.: Fast binary feature selection with conditional mutual information. Journal of Machine Learning Research 5, 1531–1555 (2004)
14. Levi, K., Weiss, Y.: Learning object detection from a small number of examples: the importance of good features. In: IEEE Conf. Computer Vision and Pattern Recognition, IEEE Computer Society Press, Los Alamitos (2004)