Neurocomputing 187 (2016) 19–26

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Face detection using representation learning

Shu Zhan*, Qin-Qin Tao, Xiao-Hong Li

Hefei University of Technology, Hefei, China

ARTICLE INFO

Article history:
Received 1 May 2015
Received in revised form
29 July 2015
Accepted 30 July 2015
Available online 3 December 2015

Keywords:
Face detection
Convolutional neural network
Deep learning
Support vector machine
Adaboost

ABSTRACT

Face representation is a crucial step of face detection system. In this paper, we present a fast face detection algorithm based on representation learnt using convolutional neural network (CNN) so as to explicitly capture various latent facial features. Firstly, in order to improve the speed of detection in the system, we train an Adaboost background filter which can remove the background most quickly. Secondly, we use the CNN to extract more distinctive features for those face and non-face patterns that have not been filtered by Adaboost. CNN can automatically learn and synthesize a problem-specific feature extractor from a training set, without making any assumptions or using any hand-made design concerning the features to extract or the areas of the face pattern to analyze. Finally, support vector machines (SVM) are used to detect instead of using the classification function of CNN itself. Extensive experiments demonstrate the robustness and efficiency of our system by comparing it with several popular face detection algorithms on the widely used CMU + MIT frontal face dataset and FDDB dataset.

1. Introduction

Face detection is the foundation of computer vision and pattern recognition technology [1,2]. It plays an important role in the face recognition, facial point detection, facial expression analysis and other topics [3]. However, because of the illumination, head pose, partial occlusion, facial expressions and other reasons, the face detection problem remains a challenge.

The first step in face detection system is to represent the face images as feature vectors. After obtaining the representation, various learning algorithms can be applied to perform the classification task [4]. Therefore, the performance of face detection algorithm mainly depends on the selected features. As for features, many studies proposed numerous hand-crafted features. The encoding methods of these hand-crafted features are designed manually based on the prior knowledge of face images (e.g., LBP or SIFT). For example, after Viola and Jones [5] proposed the first real-time face detector, Haar-like features have been adopted as the standard feature representation for face detection. Ahonen et al. [6] proposed to use the LBP features to describe the microscopic structure of the face. In addition to using a single feature, many researches use heterogeneous feature types come together to describe the human face [7–9]: Pan et al. [7] used heterogeneous feature types, including Haar feature, LBP feature, SURF feature, to represent face patterns from various aspects, which greatly improves the performance.

These articles all used hand-crafted characteristics to represent the human face; although these features also achieved good results, considerable room for improvement still exists. On one hand, Chen et al.'s [10] experiments showed that most hand-crafted features only gave similar results under the high-dimensional learning framework. It claimed that traditional hand-crafted representations suffered from a visible performance bottleneck and most of them were making different tradeoffs between discriminative ability and robustness. On the other hand, manually acquiring the optimal feature from data is very difficult. To avoid the drawbacks of handcrafted encoding methods, a lot of deep learning algorithms start to look for a new type of feature. For example, CNN could be employed to obtain simple and effective facial features [11,12].

In this paper, we propose a novel and effective face detection system based on CNN learning facial features automatically. First, we train an Adaboost classifier which can roughly find the position of faces, filter most background regions quickly, and consequently improve the detection speed of the system. Then, a feature extractor called CNN is trained to learn and extract features automatically. Making use of the obtained features, we train a SVM classifier for the final classification. The powerful and complex SVM makes the classification better than the CNN itself, which can carefully remove those remaining complex non-face patterns that cannot be ejected by Adaboost.

The rest of this paper is organized as follows: Section 2 presents an overview of some popular techniques applied to face detection. Section 3 describes the proposed face detection system. This is followed by the experimental results and performance

* Corresponding author.

http://dx.doi.org/10.1016/j.neucom.2015.07.130
0925-2312/© 2015 Elsevier B.V. All rights reserved.

analysis, presented in Section 4. Finally, Section 5 presents concluding remarks.

2. Related works

There are some significant previous studies about face detection. These studies can be grouped into four categories: knowledge-based methods, feature invariant methods, template matching methods, and appearance-based methods. Among various face detection approaches, appearance-based methods are able to learn distinctive face characteristics, so these methods have attracted much attention. In appearance-based face detection methods, the general practice is to collect a large set of face and non-face examples, and adopt certain machine learning techniques to learn a face model for classification. The key issues are what feature to extract and what learning algorithm to apply.

The boosting cascade framework by Viola and Jones [5] is a milestone in face detection. The amazing real-time speed and high detection accuracy of the face detector can be attributed to three factors: the integral image representation, the cascade framework, and the use of Adaboost to train cascade nodes. But it still has several limitations: First, the number of Haar features is too large, which is usually in hundreds of thousands level for a typical 20 × 20 sample. Selecting several effective weak classifiers takes a long time in so many features. Second, the feature representation capacity of Haar feature is very limited. It cannot well handle viewpoint, pose and illumination variations. Li et al. [13] proposed a boosting cascade based face detection framework using SURF features to outperform Viola and Jones' work. SURF feature is more distinctive and the number is smaller, so that the feature selection time is shortened and the performance is improved. Shih et al. [14] presented a novel face detection method by applying discriminating feature analysis (DFA) and SVM. DFA derived a discriminating feature vector by combining the input image, its 1-D Haar wavelet representation, and its amplitude projections. In addition to the above hand-crafted features, there are some learning-based features. For example, unlike many previous manually encoding methods, Cao et al. [15] used unsupervised learning techniques to learn an encoder from the training examples. And then they applied PCA to get a compact face descriptor. Although this scheme upgrades the performance, the careful tuning of each individual module is very labor-intensive. More important, it is unclear how to ensure the performance of the whole system by optimizing each module individually.

Usually, how to use these features to achieve best performance is a process to constantly correct errors and regulate parameters. In addition, these features are usually effective only when they are high-dimensional. And the algorithm is relatively complicated. So in order to extract effective features simply, CNN began to be widely studied. In fact, before the Viola and Jones's [5] detector was published, neural network had been a very popular approach and achieved state-of-the-art performance at that time [16]. Garcia et al. [11] applied CNN to face detection. CNN performed self-driven feature extraction and classification of the extracted features in a single integrated scheme. Chen et al. [17] added a pre-processing step and a single convolutional feature map based on Garcia's work, which can quickly filter more than 75% of backgrounds. The rest of the complex patterns were passed to CNN to deal with. Tivive et al. [18] applied Shunting inhibitory convolutional networks to face detection, which used shunting inhibitory neurons as feature detectors. It showed the proposed face detector based on a hierarchical neural network that can classify in-plane rotated faces in an image, regardless of their orientation. A few research works have been reported to apply CNN on face related problems. For instance, Zhang et al. [19] built a CNN that can simultaneously learn face/non-face decision, the face pose estimation problem, and the facial landmark localization problem. Sun et al. [12] used CNN to extract the global high-level characteristics and detect facial landmarks.

About the research on learning algorithm, AdaBoost has been proven to be an effective algorithm in the area of face detection since the milestone work of Viola and Jones. After that, variants of AdaBoost are proposed for improving the performance of face detector, such as OtBoost [20] . Recently, SVM is an effective classifier with high accuracy, which is commonly-adopted. It can determine the best discriminative support vectors for face detection and classification. However, the detection could not be executed in real time when only a single SVM-based detector was used. So, Pan et al. [7] adopted a coarse-to-fine classifier: in early stage of the system, it employed GH features to remove simple non-face patterns as soon as possible. In the middle stage, MB-LBP descriptors were applied to filter out as many as non-face patterns efficiently. More discriminative and slower SVM classifier used SURF descriptors performing the final detection in the last stage of cascade classifiers to separate face patterns from the remaining difficult non-face patterns that are similar to each other. Base on the above method, we also adopt a coarse-to-fine classifier. Firstly, Adaboost removes most backgrounds quickly. Then only a small part of candidate faces are passed to the CNN and SVM. This framework can ensure the speed of the whole system and the detection rate at the same time.

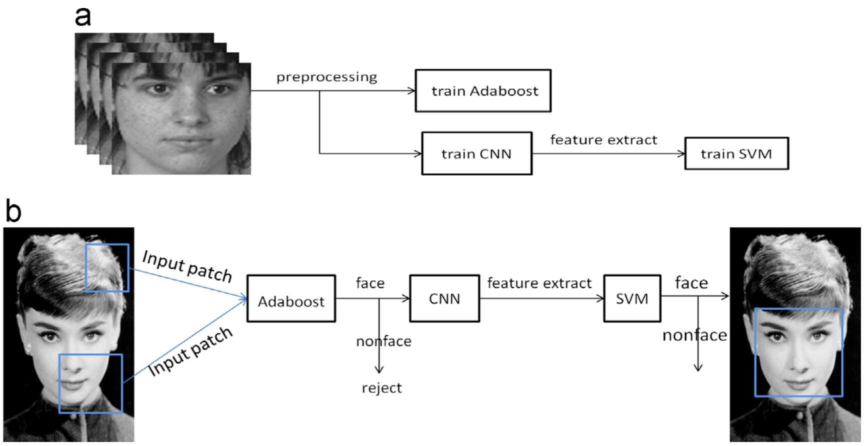


Fig. 1. (a) Training of face detection framework and (b) testing of face detection framework.

3. The proposed method

Fig. 1 shows the basic process of training and detecting human face. All training samples were scaled to a base resolution of 28 × 28 pixels, then histogram equalization was performed to decrease the variation caused by illumination changes. In the training stage, we train the three models: Adaboost, CNN, and SVM. In the testing stage, the input image was multi-scaled scanned and a lot of input patches were obtained. And the same pretreatments were carried out for these patches. These patches passed the trained Adaboost classifier first. If Adaboost classifier judged it as a face, then the patch would be passed to the next stage for further classification. If the judgment is no, the patch would be discarded directly. CNN extracted features of the received patches, and then the features were passed to SVM to make the final classification.

3.1. The Adaboost based background filter

Giving an image containing faces, most of it are backgrounds, and faces account for only a small part. Most backgrounds are obvious and simple, and these backgrounds can be easily filtered by classifier. However, for those complex backgrounds, we need more powerful classifier.

The boosting cascade framework by Viola and Jones is a great breakthrough in the field of face detection. Basically, there are three key ideas that make it able to build real-time detector: the integral images for efficient Haar feature computing, the boosting learning of weak classifier, and the cascade structure for fast negative rejection. So we employ the detector as our background filter in the first stage to filter out those obvious backgrounds, increasing the speed of system, and ensuring the detection rate at the same time.

It is important to set the threshold of the Adaboost appropriately. Setting the threshold too high may cause too many positive examples to be rejected, reducing the overall detection rate, and setting it too low may lead to too many pass-through patches that need to be classified further by the CNN, slowing down the overall detection process. So a minimum detection rate of 99.8% and a maximum false positive rate of 50% were set as the training parameters.

3.2. CNN feature extractor

3.2.1. Feature learning

Feature is a prerequisite of face detection. Its impact on the final result is of no doubt. If the face data is well described into facial features, we usually can get a satisfactory result. For face image, the pixel-level feature has no value. Studies found that for complex graphics that consist of some basic structures, the structural-level feature may work. So facing more structured and more complex graphics, we need much higher level features to represent them. A high-level expression is a combination of low-level expressions. More features mean more reference information, the accuracy will be improved. However, it can also increase the computational complexity. So how many layers of features are the most appropriate? Using deep learning approach to solve this issue is an ideal solution. The essence of deep learning is to learn more useful features through a large number of training data to construct a machine learning model that has a number of hidden layers, thus to enhance the accuracy of the classification or prediction. Compared with the manually designed encoding methods, using the depth model to learn features can portray the rich internal information of data better.

Table 1
Comparison of detecting time.

	Image size	Environment	Processing time
CNN + SVM	864 × 890	MATLAB	130 s
Adaboost + CNN + SVM	864 × 890	MATLAB	3.97 s

Recent years have seen many significant improvements in the area of representation feature learning by introduction of many depth models such as Deep Boltzman Machines (DBM), Deep Belief Networks (DBN), CNN, Recurrent Neural Networks(RNN), Autoencoders and others. The reason behind the success of these models is the learning of feature representation which is capable of capturing more intelligent features from the input data. Most of the models combine low-level representation into high-level representation, which is abstract, complex, and non-linear. Among these models, CNN is a powerful bioinspired hierarchical multilayered neural network that combines three architectural ideas to ensure some degree of shift, scale, and distortion invariance: local receptive fields, shared weights, and spatial subsampling. Besides, it reduces the number of parameters that need to be learned through the local receptive field and shared weights, cutting down the complexity of the model. So we adopt the CNN as the depth model of learning characteristics.

3.2.2. Convolutional neural network structure

CNN is a multilayer neural network, each layer is composed of multiple two dimension planes, and each plane is composed of multiple independent neurons. As shown in Fig. 2, CNN consists of 8 layers, including input layer, convolution layer, sampling layer, full connected layer and output layer. Layers C1 through C5 contain a series of planes where successive convolutions and subsampling operations are performed. These planes are called feature maps as they are in charge of extracting and combining a set of appropriate features. Each unit in a layer receives input from a set of units located in a small neighborhood in the previous layer. The small neighborhood is called local receptive fields. With local receptive fields, neurons can extract elementary visual features such as oriented edges, end-points, or corners. These features are then combined by the subsequent layers in order to detect high-level features.

Layer C1 is composed of six feature maps. Each neuron in each feature map is connected to a 5 × 5 neighborhood into the input. The step is 1, so the size of the feature map is 24 × 24. Layer S2 is composed of six feature maps, one for each feature map in C1. The receptive field of each unit is a 2 × 2 area in the previous layers corresponding feature map. Contiguous units have nonoverlapping contiguous receptive fields. So the size of the feature map is 12 × 12.

Convolution process: Each feature map unit computes a weighted sum of its input x by a 5 × 5 convolution kernel that can be learned, adds a trainable bias b_x , and then passes the results through Rectified Linear Units (ReLU), obtaining a convolution layer C_x. We adopt the ReLU as the activation function here. CNN with ReLU trains several times faster than the traditional CNN with tanh units, which has a great influence on the performance of large models trained on large datasets [21].

$$C_x = max(0, K \ast x + b_x) \tag{1}$$

Sampling process: Each unit computes the average of its four inputs, multiplies it by a trainable coefficient w_{x+1}, adds a trainable bias b_{x+1}, and passes the results through the ReLU. Thus produce a feature map S_{x+1}.

$$S_{x+1} = max(0, \Sigma C_x \times w_{x+1} + b_{x+1}) \tag{2}$$

Layer C3 is a convolutional layer with 16 feature maps. Each unit in each feature map is also connected to a 5 × 5 neighborhood in a subset of the feature maps of S2. The feature maps size is

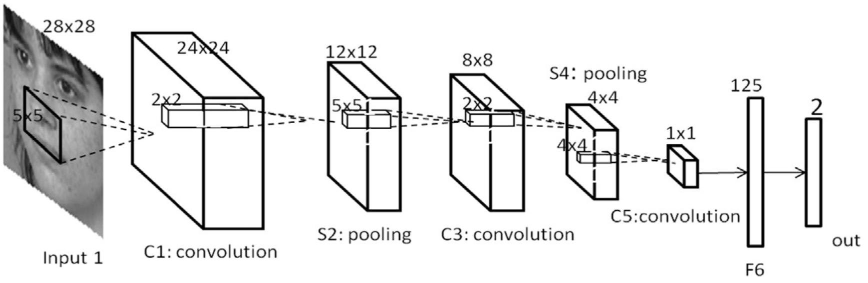


Fig. 2. The architecture of Convolutional Neural Network. Sizes of input, convolution, and pooling layers are illustrated by cuboids whose width and height denote the size of each map. Local receptive fields of neurons in different layers are illustrated by small squares in the cuboids.

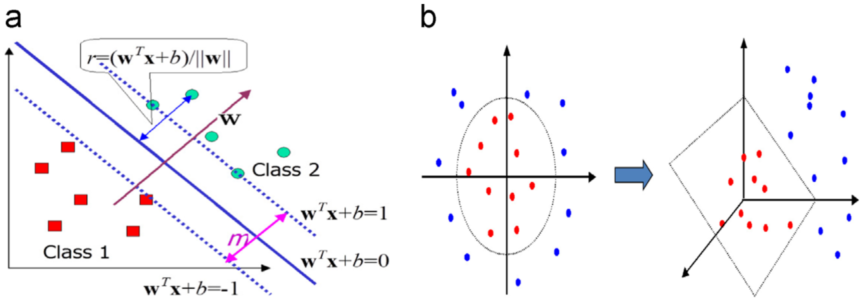


Fig. 3. (a) Linear separable and (b) linear inseparable.

8×8. Layer S4 is a subsampling layer. The receptive field of each unit is a 2×2 area in the previous layers corresponding feature map in C3, like for S2 and C1. Therefore, layer S4 has 16 feature maps of size 4×4. Layer C5 is a convolution layer, containing 120 feature maps. Each unit in each feature map is connected to a 4×4 neighborhood in a subset of the feature maps of S4. So the size of the C5 layer feature map is 1×1.

In layer C5, a series of features are extracted and fed to F6. F6 is the full connected layer with 125 units. Each of the 125 units is connected to all features maps in C5. The units in full connected layer compute the dot product between their input vector and weight vector, to which a bias is added. Then the result is passed to the ReLU function for nonlinear transformation. We take the output of the full connected layer as the extracted features of a sample. Finally, the two neurons of output layer are fully connected to the full connected layer, producing the output which is used to classify the input image as face or non-face.

The last two layers act as a classifier, the previous ones acting as feature extractors. In our scheme, instead of using the preliminary classification of CNN, we train the SVM classifier for more accurate judgment.

3.3. *The final classification based on SVM*

SVM is a statistical study theory based supervised learning method. The main idea of the learning machine is to find a hyperplane to separate the classes while minimizing the experience error and maximizing geometric edge area so as to keep balance between the complexity and the generalization ability of the model.

In linear separable case, all the samples of the same class are on the same side of the hyperplane, such that a simple line can separate them as shown in Fig. 3(a).

The training sets are denoted by $(x_i, y_i)^T, i = 1, 2, \ldots, n$, where $x_i \epsilon R^d$ stands for the ith sample of the training sets, and $y_i \epsilon \{+1, -1\}$ stands for the ith desired output. The optimal classification surface is as follows:

$$f(x) = w^T x + b = 0 \tag{3}$$

The distance between the classification surface and the most neighboring sample is $r = 1/\|w\|$. Thus, the learning problem for SVM classifier can be formulated as:

$$max_{w,b}\frac{2}{\|w\|} \doteq min_{w,b}\frac{1}{2}\|w\|^2 \tag{4}$$

Subject to the constraint: $y_i g(x_i) = y_i(w^T x_i - b) \geq 1, i = 1, 2, \ldots, n$

However, the optimal separating plane discussed earlier is too strict in many practical situations. When the samples in a classification problem can or can almost be separated linearly, the optimal separating hyperplane can be constructed by solving a relaxed quadratic programming problem which introduces several slack variables ξ_i and a penalization C for cases that cannot be classified correctly.

$$min_{w,b}\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i \tag{5}$$

Subject to the constraint: $y_i g(x_i) = y_i(w^T x_i - b) \geq 1 - \xi_i, i = 1, 2, \ldots, n$ $, \xi_i > 0$, $C > 0$. C is the punish coefficient for the incorrect cases.

Eq. (5) is a typical quadratic programming problems which can be solved by the Lagrange multiplier method. Specific derivation process is not repeated herein. At last, the optimization problem of objective function can be transformed into following equation:

$$L(\alpha) = \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j x_i^T x_j \tag{6}$$

Constraint is: $\sum \alpha_i \alpha_j y_i y_j = 0, 0 \leq \alpha_i \leq C$. The training data x_i associated with nonzero coefficients α_i are called support vectors. The output of SVM is thus defined by

$$f(x) = sgn\left(\sum_{i=1}^{n}\alpha_i^* y_i (x_i^T x) + b^*\right) \tag{7}$$

where $\alpha_i^*, i = 1, 2, \ldots, n$ and b^* are obtained by Formula (8).

However, in practice, most problems are nonlinear. Nonlinear case has been a difficult problem in the field of classification, which mainly because of the difficulty of constructing nonlinear discriminant function. In nonlinear case, SVM uses kernel function to map the original feature into a higher dimension feature space where they can be separated using a linear hyperplane. As shown in Fig. 3(b), a kernel function is used here. So Eq. (8) can be written as follows:

$$L(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \tag{8}$$

$K(x_i, x_j) = exp\left(\frac{|x_i - x_j|^2}{\sigma^2}\right)$ is a RBF kernel. Constraint is: $\sum \alpha_i \alpha_j y_i y_j = 0, 0 \le \alpha_i \le C$. Therefore the final decision function is

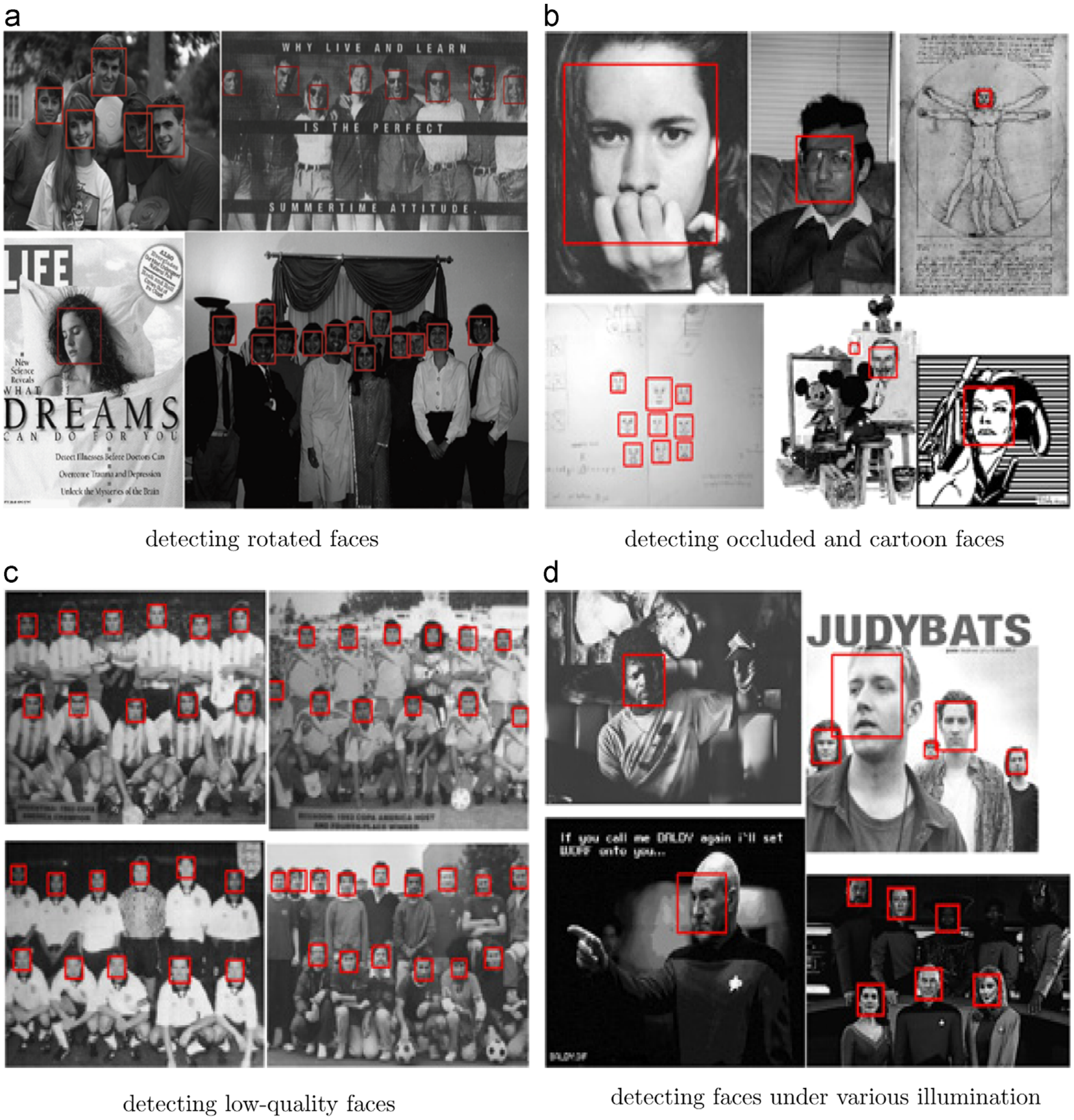
$$f(x) = sgn\left(\sum_{i=1}^{n} \alpha_i^* y_i K(x_i, x) + b^*\right) \tag{9}$$

CNN can perform feature extraction, as well as classification in a single integrated scheme. Its own classification function uses sigmoid function to map the extracted features into two dimensional outputs. If the first dimension of output is larger, then classify it as face, otherwise classify it as non-face. In complex background, if only simple sigmoid function is utilized for classification, it cannot detect the faces correctly usually. And the excellent performance of SVM solving linear inseparable problem makes us choose it as the final classifier to do a more accurate judgment.

4. Experimental results

For training and testing, a set of 10,000 frontal face images were collected from various sources. These face images cover ± 15° up–down out-of-plane rotation (Pitch) and ± 20° left–right out-of-plane rotation (Yaw). All face images were scaled to a base resolution of 28 × 28 pixels, and then histogram equalization and intensity normalization were performed. Additionally 20,000 non-face images were collected as negative samples. The same pre-treatment was performed to these samples.

We train the Adaboost background filter, setting the minhitrate as 0.998, maxfalsealarm as 0.5, stage as 3, so Adaboost classifier can quickly remove more than 80% of backgrounds. Then we train our CNN, using stochastic gradient descent to upgrade the network parameters, setting the momentum as 0.9, learning rate as 0.005. The trained CNN is utilized to extract the characteristics of the training samples. Finally, we use the normalized features to train the SVM classifier. Training SVM needs to set many parameters. The most important two parameters are −c and −g. In order to seek the optimal parameters, we employ the PSO algorithm for SVM parameters optimization.


a
WHY LIVE AND LEARN
IS THE PERFECT
SUMMERTIME ATTITUDE
LIFE
WHAT DREAMS CAN DO FOR YOU
b
c
d
JUDYBATS
If you call me DADDY again I'll set WOFF onto you...

detecting rotated faces

detecting occluded and cartoon faces

detecting low-quality faces

detecting faces under various illumination

Fig. 4. Examples of detecting faces.

In this section, we present several experiments to evaluate the performance of our detector. For evaluation we use two challenging public datasets: FDDB [22] and CMU+MIT [16]. They are widely used to evaluate the face detection methods.

4.1. Evaluation on CMU+MIT data set

The CMU+MIT data set contains test set A, B, C (test, test-low, newtest) and rotated test set. We use the three test sets (test A, B, C), without the rotated one, containing 130 images with 511 faces. We first show some detection results by our detector on the CMU+MIT data set.

The faces in CMU+MIT dataset have different sizes, poses, expressions, and lighting conditions, but the proposed method can handle them well, as shown in Fig. 4. For example, Fig. 4 (a) shows some examples of detecting rotated faces. It proves that our method can detect not only frontal faces, but also some rotated faces. Because our training examples cover ±15° up-down out-of-plane rotation (Pitch) and ±20° left-right out-of-plane rotation, so our detector can efficiently detect faces within this angle range. If the rotation angle is larger, the performance will decrease significantly. The proposed method can detect the occluded faces and hand-drawn faces as shown in Fig. 4 (b). The proposed method can detect these faces in the data set effectively because CNN can learn a good representation of facial features. What is more, the combination of CNN and SVM makes sense. Our detector can detect the faces in low quality images correctly, as shown in Fig. 4 (c). For those faces under the dim light as well as the different scale faces, we can also detect them effectively, as shown in Fig. 4 (d). However, when the illumination is too dark and the characteristic is not obvious, the detector may fail to detect those faces. In Fig. 4 (d) there are two very dark faces missed by the proposed method.

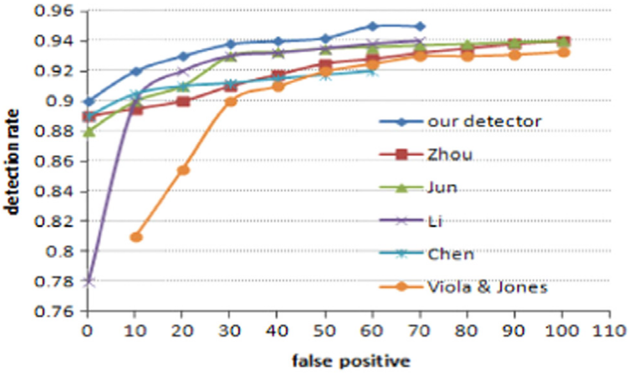


Fig. 6. ROC curves of different algorithms on CMU+MIT data set.

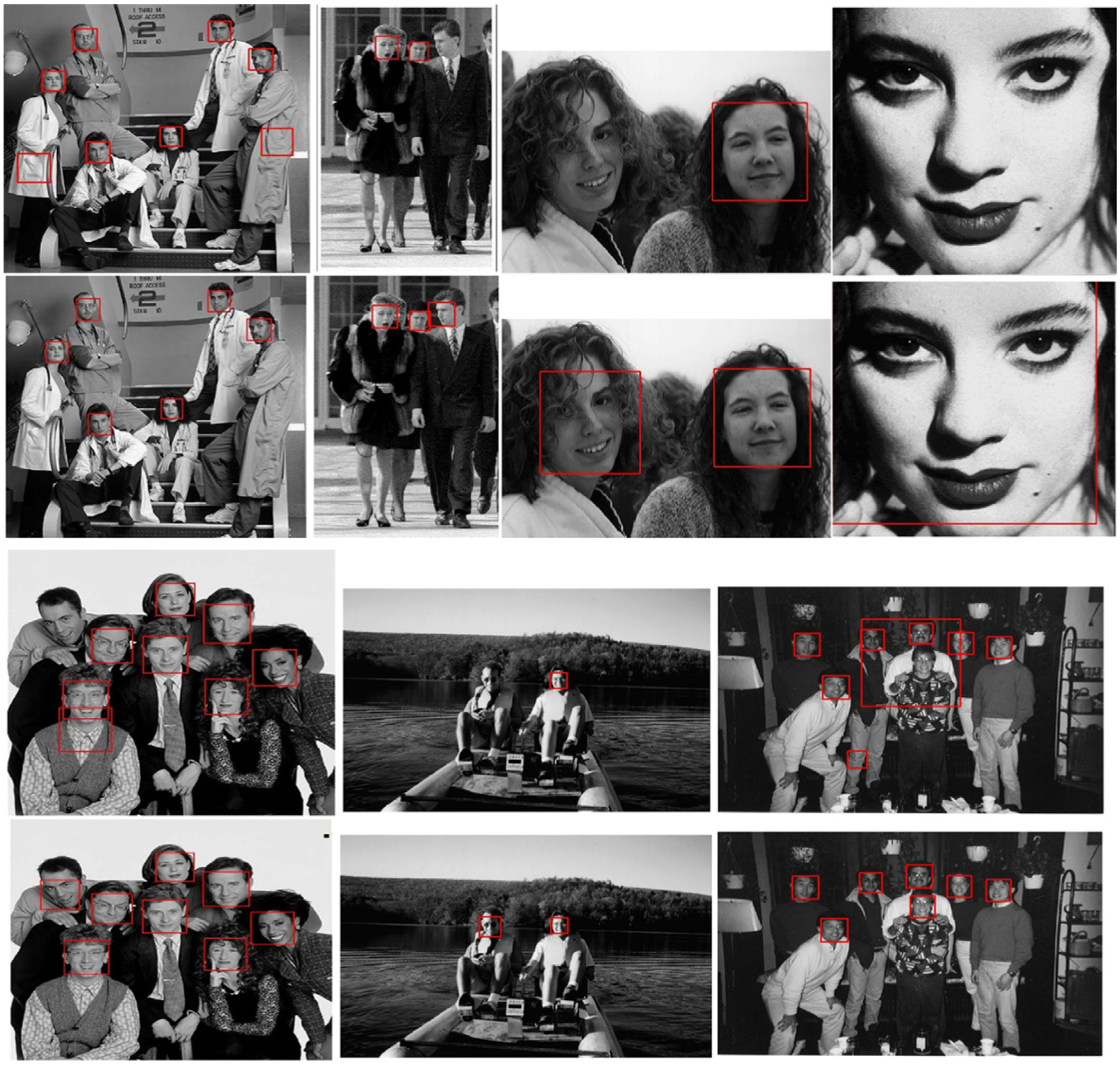


Fig. 5. Comparison detection examples using our detector and Viola and Jones' detector on CMU+MIT dataset (examples at the first and third row are detection results from Viola and Jones' detector, whereas examples at the second and fourth row are detection results from our face detector).

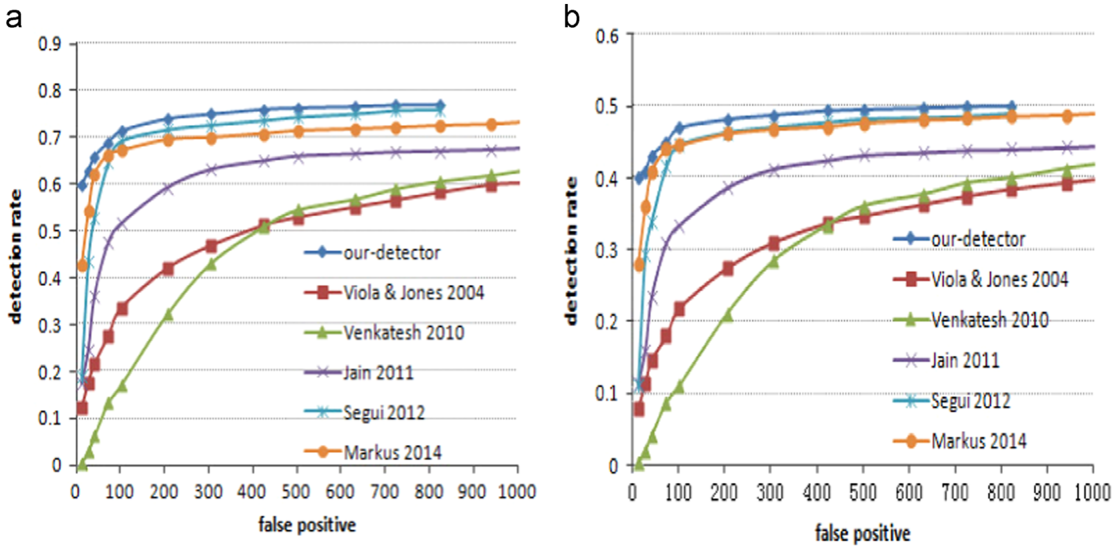


Fig. 7. (a) Discrete score ROC curves and (b) continuous score ROC curves for different methods on UMass FDDB dataset.

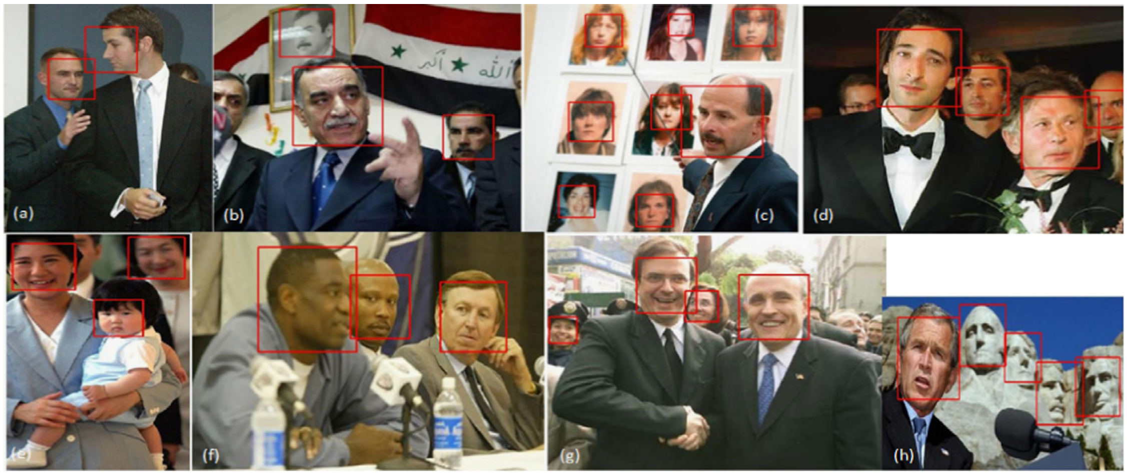


Fig. 8. Some examples of face detection results on FDDB dataset.

To illustrate the superiority of our detector against the famous Viola and Jones' detector, we compare the detection results from both detectors, as shown in Fig. 5. In particular, we implemented Viola and Jones' detector using OpenCV2.3.1 with the default frontal face classifier configuration (i.e. haarcascade-frontalface-default.xml). As can be seen, because of feature extraction using CNN and classification using SVM, some examples that are successfully detected by our detector failed in Viola and Jones' detector. In addition to, our detector can effectively reduce false positives.

Finally, we compare our algorithm with several related algorithms on CMU+MIT data set. Fig. 6 plots the Receiver Operating Characteristics (ROC) curves of our method as well as other popular face detection algorithms including Viola and Jones [5], Li et al. [13], Jun et al. [23], Zhou et al. [24], Chen et al. [17]. Li et al. [13] presented a boosting cascade face detection framework which used SURF features. Jun et al. [23] proposed a face detection method which used local gradient patterns (LGP) to represent face. Zhou et al. [24] employed Multi-Block local gradient patterns (MB-LGP) as the features, and used SVM to perform the classification. Chen et al. [17] proposed a face detector on a modified CNN.

As shown in Fig. 6, our detector is more efficient than other algorithms. Especially for cases at low false positives, our detector can still achieve good results. The comparison with Viola and Jones [5], Li et al. [13], Jun et al. [23], Zhou et al. [24] proves that compared with the hand-crafted features, the features extracted by CNN improve the detection rate in a certain extent. Additionally, the comparison with Chen et al. [17] shows that the combination of SVM and CNN in our approach outperforms the traditional CNN.

4.2. Evaluation on FDDB dataset

The CMU+MIT dataset is a little out-of-date as it only contains gray, relative low-resolution images, and the size of the data set is too small to reflect nowadays data explosion status. Hence, the UMass face detection dataset and benchmark (FDDB) is introduced [22]. It contains 2845 images with a total of 5771 faces under a wide range of conditions. Besides, it provides a systematic protocol to evaluate performance of face detection system. We use the "ROC.txt" and the evaluation code files from FDDB website to generate the discrete score and continuous score ROC curve for comparison to some available results on the benchmark [5,25–28] as shown in Fig. 7.

In the discrete setting, a detection window is considered correct if its intersection-over-union ratio with respect to an annotated face region is larger than 0.5. This criterion is commonly used in object detection evaluation. In the continuous setting, the overlapping ratio is used as a weight for every detection window. This criterion is much stricter. So we can see in Fig. 7, the detection rate in continuous score

ROC curve is much lower than discrete score ROC curve. But it is obvious that our detector outperforms others under both protocols.

FDDB dataset is very challenging because the faces cover various poses, rotation, occlusion and illumination. Facing the challenging data set, our method still achieves good results, as shown in Fig. 8. However, when the facial points are occluded deeply, due to the lack of sufficient features, these faces will be rejected by our detector. Such as (b), (d) and (g), there are several faces that are occluded excessively which cannot be detected by our detector.

5. Conclusion

This paper puts forward an effective face detection system based on the combination of CNN and SVM. Characteristic is very important to face detection, choosing which kind of features to represent human face is a difficult problem. In this paper, we employ multilayer CNN as a feature extractor to acquire problem-specific features automatically. In order to quickly filter out most backgrounds, we train a background filter based on Adaboost classifier. Thus even if the CNN is very complex, the speed of the whole system is still fast. In the first stage, all the possible human faces are roughly captured by Adaboost classifier. Then SVM is used to further filter non-face, and accurately locate the face region. In the proposed system, the extracted features by CNN can be a good representation for face detection, coupled with the advantage of SVM for classification, which makes the proposed approach further improve the detection rate. The experiments on the dataset also prove the validity of our method.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grants 61371156 and 61371155, Anhui Province Science and Technology Research Programs under Grant 1401B042019.

References

[1] J. Yu, R.C. Hong, M. Wang, J. You, Image clustering based on sparse patch alignment framework, Pattern Recognit. 47 (11) (2014) 3512–3519.
[2] J. Yu, Y. Rui, D.C. Tao, Click prediction for web image reranking using multimodal sparse coding, IEEE Trans. Image Process. 23 (5) (2014) 2019–2032.
[3] C.Q. Hong, J. Yu, D.C. Tao, M. Wang, Image-based 3d human pose recovery by multi-view locality sensitive sparse retrieval, IEEE Trans. Ind. Electron. 62 (6) (2015) 3742–3751.
[4] J. Yu, Y. Rui, Y.Y. Tang, D.C. Tao, High-order distance based multiview stochastic learning in image classification, IEEE Trans. Cybern. 44 (12) (2014) 2431–2442.
[5] P. Viola, M. Jones, Robust real-time face detection, Int. J. Comput. Vis. 57 (2) (2004) 137–154.
[6] T. Ahonen, A. Hadid, M. Pietikainen, Face recognition with local binary patterns, Lecture Notes in Computer Science, (2004) 469–481.
[7] H. Pan, Y.P. Zhu, L.Z. Xia, Efficient and accurate face detection using heterogeneous feature descriptors and feature selection, Comput. Vis. Image Underst. 117 (1) (2013) 12–28.
[8] J. Yu, Y. Rui, B. Chen, Exploiting click constraints and multiview features for image reranking, IEEE Trans. on Multimed. 16 (1) (2014) 159–168.
[9] J. Yu, D.C. Tao, Y. Rui, Learning to rank using user clicks and visual features for image retrieval, IEEE Trans. Cybern. 45 (4) (2015) 767–779.
[10] D. Chen, X. Cao, F. Wen, J. Sun, Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification, In: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, 2013, pp. 3025–3032.
[11] C. Garcia, M. Delakis, Convolutional face finder: a neural architecture for fast and robust face detection, IEEE Trans. Pattern Anal. Mach. Intell. 26 (11) (2004) 1408–1423.
[12] Y. Sun, X.G. Wang, X.O. Tang, Deep convolutional network cascade for facial point detection, In: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, 2013, pp. 3476–3483.
[13] J. G. Li, T. Wang, Y. M. Zhang, Face detection using surf cascade, In: IEEE International Conference on Computer Vision Workshops, 2011, pp. 2183–2190.
[14] P. Shih, C.J. Liu, Face detection using discriminating feature analysis and support vector machine, Pattern Recognit. 39 (2) (2006) 260–276.
[15] ZM. Cao, Q. Yin, XO. Tang, J. Sun, Face recognition with learning-based descriptor, In: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, 2011, pp. 2707–2714.
[16] H. Rowley, S. Baluja, T. Kanade, Neural network-based face detection, IEEE Trans. Pattern Anal. Mach. Intell. 20 (1) (1998) 23–38.
[17] Y.N. Chen, C.C. Han, C.T. Wang, B.S. Jeng, K.C. Fan, A CNN-based face detector with a simple feature map and a coarse-to-fine classifier, IEEE Trans. Pattern Anal. Mach. Intell. 99 (2009) 1–13.
[18] F. Tivive, A. Bouzerdoum, A hierarchical learning network for face detection with in-plane rotation, Neurocomputing 71 (16–18) (2008) 3253–3263.
[19] C. Zhang, Z.Y. Zhang, Improving multiview face detection with multi-task deep convolutional neural networks, Appl. Comput. Vis. (2014) 1036–1041.
[20] J.B. Wen, Y.S. Xiong, S.L. Wang, A novel two-stage weak classifier selection approach for adaptive boosting for cascade face detector, Neurocomputing 116 (2013) 122–135.
[21] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Adv. Neural Inf. Process. Syst. 25 (2012) 1097–1105.
[22] V. Jain, E. Learned-Miller, Fddb: A Benchmark for Face Detection in Unconstrained Settings, University of Massachusetts, Amherst (UM-CS-2010-009).
[23] B. Jun, D. Kim, Robust face detection using local gradient patterns and evidence accumulation, Pattern Recognit. 45 (9) (2012) 3304–3316.
[24] S.H. Zhou, J. Yin, Face detection using multi-block local gradient patterns and support vector machine, J. Comput. Inf. Syst. 10 (4) (2014) 1767–1776.
[25] BS. Venkatesh, S. Marcel, Fast bounding box estimation based face detection, In: Proceedings of the Workshop on Face Detection of the European Conference on Computer Vision, 2010.
[26] V. Jain, E. Learned-Miller, Online domain adaptation of a pre-trained cascade of classifiers, In: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, 2011, pp. 577684.
[27] S. Segui, M. Drozdzal, P. R. J. Vitria, An integrated approach to contextual face detection, In: International Conference on Pattern Recognition Applications and Methods, 2012, pp. 9097.
[28] N. Markus, M. Frljak, I. Pandzic, A method for object detection based on pixel intensity comparisons organized in decision trees, (2013). arXiv:1305.4537.

Shu Zhan received the BE and ME degrees in Electronic Engineering from the Hefei University of Technology in 1990 and 1993, and the PhD degree in Electronic Engineering from University of Science and Technology of China in 2000. He is an associate professor in the School of Computer and information at Hefei University of Technology. He was a postdoctoral research associate at the University of Tokyo from 2002–2004. His research interests include pattern recognition, computer vision and medical imaging, with a focus on biometrics, medical image segmentation and applied machine learning. He is a member of the IEEE.

Qinqin Tao received the BE degrees in Electronic Engineering from the Hefei University of Technology in 2013, she is a master degree student in Hefei University of Technology; her research interests include pattern recognition and medical image analysis.

Xiaohong Li received the BE and ME degrees in Electronic Engineering from the Hefei University of Technology in 1993 and 1998, she is an associate professor in the School of Computer and information at Hefei University of Technology. Her research interests focus on image coding and applied machine learning.