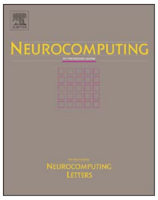
Neurocomputing 235 (2017) 10–14

Contents lists available at ScienceDirect

Neurocomputing

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



Data augmentation for unbalanced face recognition training sets

Biao Leng[a], Kai Yu[a], Jingyan QIN[b,*]

[a] *School of Computer Science, Beihang University, Beijing 100191, PR China*
[b] *School of Mechanical Engineering, University of Science & Technology Beijing, Beijing 100083, PR China*

ARTICLE INFO

Communicated by Zidong Wang

Keywords:
Data Augmentation
Face Recognition
Convolutional Neural Network
Support Vector Machine

ABSTRACT

Face recognition remains a challenging problem. While one-to-one face verification has been largely tackled, verification-based classification problem still demands effort. To further enhance the verification models, one solution is to fully utilize the unbalanced training sets, where, while abundant samples are provided for some subjects, there are often so few samples available for the rest. These subjects with too few samples can contribute little to the model learning. Therefore, before training a model, algorithms usually perform data augmentation on the whole dataset, especially on subjects with insufficient samples. In this paper, a new augmentation method is proposed, targeting on data augmentation for face classification algorithms. Instead of directly manipulating the input image, we perform virtual sample generating on feature level. The distribution of feature maps is first estimated, then random noise consistent to the distribution is applied to the feature vectors of training samples. Our method is based on Joint Bayesian Face Analysis, and we also develop an algorithm to boost the whole procedure. We conduct experiments based on high dimensional LBP features and features extracted by a shallow Convolutional Neural Network, and succeed to verify the effectiveness of this method, using image data from benchmark dataset LFW.

1. Introduction

Face recognition is a long-studied integrated task to locate the face and distinguish to whom the face belongs. The latter sub-task can be categorized into two types: face verification and face classification. Verification models take two images as input, and determine whether they belong to the same person. A popular routine is first extracting feature vectors of both faces, then calculating the similarity between them by directly using cosine distance or applying metric learning methods such as Bayesian Face Recognition [1]. In recent years, with the rise of deep learning algorithms, the one-to-one verification problem has been largely solved. A famous benchmark dataset for face verification, i.e. the LFW [2], has reported many algorithms with accuracy even higher than human performance.

However, the classification task remains a difficult problem. Different from object classification, the number of classes in face classification is not fixed, since each subject correspond to a class. If we simply adopt conventional classifiers such as Softmax or Multi-Class SVMs, the model trained on one dataset cannot be applied on another dataset or practical scenes, since they may share little or no subjects. A popular work-around is to use similarity comparing models for classification task, considering its successful application in existing verification algorithms. For a target subject, this method compares its similarity with each candidate class, and the most similar one is assigned as the target's identity. This method, however, exposes the weakness of some models, since difficult negatives only harms the accuracy curve during evaluation of the verification, but affect almost every result in the classification task. In no doubt, a perfect model (100% accurate) can guarantee perfect classification accuracy, but it is still a long way to go. Therefore, it is still in great demand to promote the accuracy of similarity comparing models.

Besides effective algorithms and neural network architecture designs, training data is also a key factor to the accuracy of models. However, as the size of face datasets has reached a large amount, further enlarging the datasets has increasingly larger cost while less benefits on the model performance. Also, many datasets suffer from a long tail phenomenon, that is, many subjects in the datasets have very few samples, making contribution of them thus trivial in the learning process. Intuitively, these subjects occupies so small a volume in the feature space, that they are easily viewed as outliers by classifiers and has limited influence to force the model to fit a better or higher-dimensional feature space. On the other hand, many algorithms focus on single sample training conditions [3–5]. They indeed achieve higher accuracy than traditional algorithms on this specified extreme condition. However, directly applying them on situations where subjects with only one sample and others with multiple samples both exist in

[*] Corresponding author.
E-mail address: qinjingyanking@foxmail.com (J. QIN).

http://dx.doi.org/10.1016/j.neucom.2016.12.013
Received 24 December 2014; Received in revised form 29 October 2015; Accepted 6 December 2016
Available online 10 December 2016

the datasets (see LFW [2]).

In this work, we present a novel method of data augmentation, aiming at solving the unbalancing problem of existing datasets. Different from popular data augmentation methods, which directly manipulate the images to produce extra sample images, our method generate virtual samples on feature level, that is, the training samples remain as origin during the feature extraction stage, and random noise is added to the extracted features to form extra samples. The classifier is trained on a union of the original feature set and the augmented feature set. This is based on the intuition that direct manipulation of the image might introduce incorrect information to the training data. For example, fitting a completely different 3D face model to one's 2D face image may result in an unreasonable virtual sample, and directly analyzing the distribution and principle components on pixel level is also difficult, which is right the reason why we classify faces with the extracted features but not raw images. On the contrary, if we assume our feature extraction model is powerful and discriminative, the inter-class and intra-class variance can be easily analyzed from the extracted features. Therefore, adding noises with respect to the distribution of intra-class variance can enrich the datasets without harming discrimination of the models. We also design a shallow Convolutional Neural Network (CNN) to extract features, compared to a high dimensional Local Binary Patterns (LBP) feature extraction proposed in [6], then perform sample augmentation on feature level for classes with insufficient samples. We estimate the improvement brought by the augmentation to the training process of SVM on the LFW dataset.

This paper is organized as follows. In section 2, we give a brief review of some related works on face verification and face identification. In section 3, we introduce the detailed implementation of our algorithm. We show our experiments in section 4, and make a conclusion in section 5.

2. Related work

Parallel with the rapid development of deep learning algorithms featuring high accuracy, many efforts are also made on addressing insufficient training sample problems. J. Lu, Y.P. Tan and G. Wang proposed their method based on partition and multifold analysis focusing on single training sample situations [7]. There are also attempts to apply conventional algorithms which originally require multi-samples on simple sample problems. Q.X. Gao, L. Zhang and D. Zhang tried to apply Fisher Linear Discriminant Analysis (FLDA) on single sample classes by using Singular Value Decomposition (SVD) to decompose the face image in order to get a difference image for evaluating the within-class scatter matrix [8]. Y. Su, S. Shan, X. Chen, W. Gao also make attempts to apply FLDA but instead propose an Adaptive Generic Learning (AGL) method to preprocess the training set [9].

The idea of extending sample sets in this work is not unprecedented. There have been already a number of data augmentation methods. The most simple while popular ones are 2D geometric transformations such as mirroring (generating horizontally flipped image of an original training sample), rotating (rotating an image around the center on z-axis at random or preset angles and zero-padding the resulting vacancy if necessary) and random cropping (also called oversampling, cropping an image with randomly selected bounding boxes). Meanwhile, there are also augmentation methods targeting on pixel appearance, such as photometric transformations and color jittering. Besides 2D methods, some data augmentation based on 3D models have also been proposed, such as [3]. It extends samples by trying to rotate the face in the space to create virtual facial images. However, the effectiveness of 3D methods are based on the assumption that 3D model fitting is a tackled question, but anyway, 3D mapping itself is a challenging problem. On the contrary, we do sample augmentation on feature level, based on the hypothesis that same illumination or pose changes to a face share similar effects on extracted feature vectors.

The method we use to extend sample sets is derived from [10]. The original method is used for verification. After analyzing the covariance matrices of the identity and in-class variation components of images using an EM-like algorithm, it use a log likelihood ratio to determine whether the two face images are of the same person. We believe this method holds a strong ability of discriminating identity and in-class variation, thus can also be applied in other applications.

The design of our shallow CNN is derived from some successful examples. The setting of keeping the same the feature map sizes after convolution layers and the choice of activation function is inspired by [11]. Also, we apply many training tricks, such as dropout from [12].

3. Algorithms

The goal of feature-level data augmentation is to generate virtual samples by adding random noise to pre-extracted features, so as to enrich the data amount for training the classifier, especially for subjects with too few samples. In this section, we describe this procedure in a flow of algorithms.

3.1. Face alignment

Although the underlying objective of data augmentation is to generate unseen views of subjects, we first perform face alignment to all original training samples. This is because in the face recognition problem, data augmentation and even multi-sample training are actually a supplementary to the face alignment process. Consider that if we have a perfect face alignment method, then data augmentation is not necessary anymore, since any incoming face can be correctly frontalized, thus a model can easily distinguish subjects from an absolutely frontal faces. Therefore, the objective of data augmentation in this work becomes to generate views of subjects which are hard negatives of face alignment methods.

Given a face image, we adopt the face alignment method proposed in [13] to locate the facial landmarks. Denoting the distribution of the landmarks as a vector, which is a concatenation of the coordinates, the alignment task can be expressed as:

$$\min_{W^t,\Phi^t} \sum_{i=1}^{N} \left\| \pi_l \circ \Delta\widehat{S_i^t} - W_l^t \Phi_l^t (I_i, S_i^{t-1}) \right\|_2^2 \tag{1}$$

where i iterates over all the training samples, specifying using the lth facial landmark, π_l is an operation extracting the coordinate of the facial landmark, and $\Delta\widehat{S_i^t}$ denotes the increment from the current estimated shape of sample I to the ground-truth shape. ${\Phi_l}^t$ extracts local binary features from the sample image I_i according to the current estimated shape S_i^{t-1}, and ${W_l}^t$ is part of a global regression matrix. The training process can be divided into 5 cascade stages each with two phases. One is using standard regression random forests to learn the ${\Phi_l}^t$s, and the other is to use global linear regression to learn the ${W_l}^t$s.

In each stage, a random forest consists of a set of decision trees. Each tree divide samples according to one currently estimated facial landmarks location. In each node, two pairs of coordinate offsets together with a randomly generated threshold are selected to best decrease the shape increment variance of all the samples fallen into this node. Difference of two pixels located by the summation of the landmarks location and the pair of offsets are compared to the threshold to further divide the samples to two sub-nodes. Each tree generates for a sample a short binary vector, which contains only one 1 at the index of leaf which the sample falls into. And all these binary vectors are concatenated to be the final binary vector for the sample in this stage.

With all the binary vectors as well as the target shape increments, a global matrix can be learnt by linear regression to map these vectors to increments. Then the estimated shapes are updated by adding the

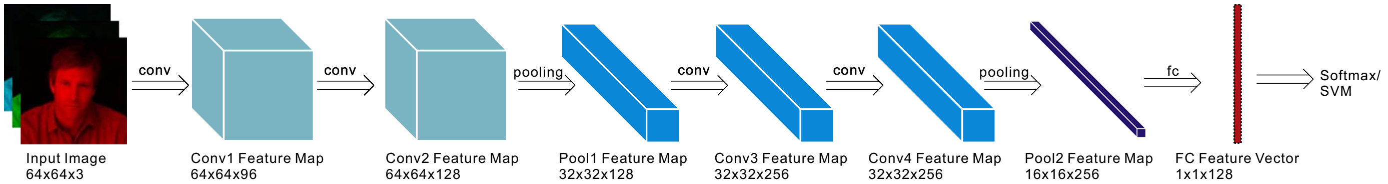


Fig. 1. Structure of out shallow CNN used for feature extraction. Feature map sizes remain the same after each convolution layer and drop only after pooling layer. Two sets of two convolution layers and a max-pooling layer precedes a full connection layer, which outputs a 128-dimensional feature vector.

product of the corresponding binary vectors and the matrix to them. The updated shapes are used for the next stages regression. The shapes are expected to incrementally approach the ground-truth shapes.

3.2. Feature extraction

With located facial landmarks, as described in [6], we can generate LBP features in multi-scale square patches centered at the landmarks, then concatenated into a high dimensional vector, which is then compressed by methods like Principle Components Analysis (PCA) into a relatively low dimension feature.

To examine the efficiency of out data augmentation method on different features, we also design a shallow CNN to extract feature from detected faces. Our CNNs structure is shown in Fig. 1. It contains 2 sets of layers, each of which contains two adjacent convolution layers and one max-pooling layer. The activation functions are set as ReLU. The size of feature map remains the same after each convolution layer and changes only after pooling layers. After the second pooling layer comes a full connection layer, which directly maps the pooled feature maps into a 128-dimensional feature vector.

At training phase, there is also a softmax layer after the feature layer for loss computing. We use a database of face images collected from the Internet, with 5,000 subjects and 190,000 images, to train this CNN. At test phase, the softmax layer is removed, and the feature output can be utilized by SVMs.

The shallow CNN is designed based on the observation that, in many deep models, the first two layers act as general feature extraction regardless of classification targets, which is similar to the human-manipulated features such as LBP. For a fair comparison between CNN-based models and LBP-based models, we only adopt two layers for 2D feature extraction. Note that the shallow CNN can be replaced with modern deep models, taking output features (or feature maps) of deep stages for augmentation.

3.3. Data augmentation on feature level

As described in [10], a face can be represented by the sum of two independent Gaussian variables and a constant:

$$x = \mu + \epsilon + c, \tag{2}$$

where x is the observed face, μ is the identity, ϵ is the face variation like illumination, poses and expressions, within the same subject, and c is a constant base offset, which can be viewed as an average of all the faces. The μ and ϵ are viewed as two Gaussian hidden variables. The original purpose of this representation is to estimate the covariance matrices S_μ and S_ϵ for computing the log likelihood ratio. The training process adopts an EM-like algorithm. In the E-step, µs and ϵs are estimated in each class respectively according to the currently estimated covariance matrices, using the following formula:

$$E(h|x) = \Sigma_h P^T \Sigma_x^{-1} x, \tag{3}$$

where

$$P = \begin{bmatrix} I & I & 0 & \dots & I \\ I & 0 & I & \dots & I \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ I & I & 0 & \dots & I \end{bmatrix}, \tag{4}$$

x=[x_1,..., x_m] are the m observed samples in the class, h=[μ; ϵ_1;...; ϵ_m] represent the hidden variables, Σ_h=$diag$(S_μ, S_ϵ,..., S_ϵ) is the covariance matrix of x if we assume that $x \sim N(0,\Sigma_x)$. Then in the M-step, the covariance matrices are updated according to all the new μs and ϵs.

In this work, we instead make use of the μs and ϵs. Assuming that face variations caused by the same illumination, poses or expression changes have similar effects on the feature vectors of faces sharing similar characteristics, we can extract these changes to create virtual samples.

First, we estimate the μs and ϵs from classes with multiple samples. It is better if we have an extra dataset, so that we can have abundant of variation materials. Then for a class we want to augment, we apply the trained S_μ and S_ϵ on its samples to get the class identity vector μ and variation vector(s) ϵ(s) using the same method as the E-step at the training phase. Then we search in all the pre-stored samples for the best reference by finding a pair of identity vector and variation vector in the same class closest to the pair consisting of the mu and an ϵ in this class. The similarity function of two vectors can be defined as:

$$F(x_i, x_0) = \cos < x_i, x_0 > , \tag{5}$$

which is a simple cosine distance function. A similar pair of vectors may satisfy the assumption mentioned above, so the other variation vectors in the pre-stored class can be added to the μ to create reasonable virtual samples.

Note that for multi-sample classes, we randomly choose an ϵ to form a pair of reference vector together with its μ. This process is shown in Fig. 2.

3.4. Dual locality sensitivity hashing

It is time costing to linearly compare the query vectors to each sample. Here we develop the Locality Sensitivity Hashing (LSH) algorithm to boost this procedure. LSH is an algorithm highly efficiently solving the Approximate Nearest Neighbor (ANN) problem. It divides the vectors into several bins by one or several hash tables. Vectors falling into the same bin are expected to have high possibility to be close to each other. So for each query vector, the hash functions are applied on it to get the bin number, then all the samples in the bin are taken out to conduct linear comparing to the query vector to find the most similar vector.

We choose the hash function designed for the simple cosine distance:

$$H(V) = sgn(V \cdot R), \tag{6}$$

where R is a random projecting vector and sgn is the signal function. The operation can be viewed as dividing the feature space by a hyperplane specified by R.

In this work, we use only one hash table with AND-construction, that is, we generate several k hash functions (k=$log_2(N)$ in our work, N is the number of pre-stored samples), and each hash function outputs a binary code. A concatenation of these binary codes is regarded as the bin label.

However, the LSH is designed for one-to-one vector comparison, and here we need to find a similar pair of vectors. Because we actually does not need highly precise similarity, we adopt the following solution:

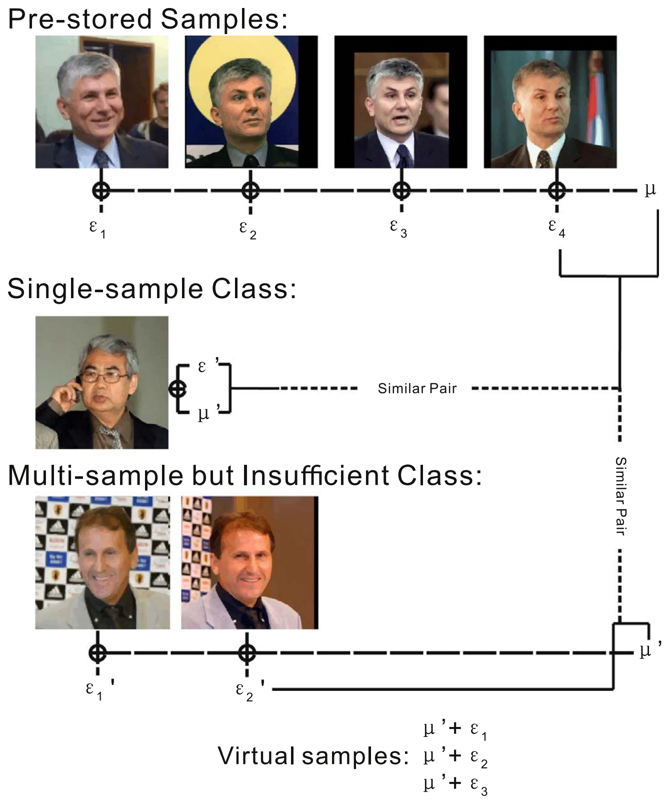


Fig. 2. Sketch map of the data augmentation procedure. For multi-sample classes, the selected material vector pair only need to be similar to a pair of vectors consisting of the identity vector μ and a random variation vector ϵ. The virtual sample generating method is same for the two kinds of classes.

a) Build LSH tables on the pre-stored samples for identity vectors and variation vectors respectively.
b) For a query pair of identity vector μ and variation vector ϵ, get the identity vectors in the same bin as ϵ, count the similarity values s_1's between them and ϵ, and figure out which classes have appeared.
c) Get the variation vectors in the same bin as ϵ and choose only those whose class has appeared in step b). Count the similarity values s_2's between them and ϵ, and pick a pair of identity vector and variation vector in a same class with the maximum $s_1 + s_2$.
d) Add the other variation vectors in the chosen class to μ respectively to create virtual samples.

This procedure can run in a relatively high speed because there are much fewer identity vectors than variation vectors, and the condition in step c) further limit the times of linear comparison.

Note that in step c), it is possible that we cannot find any variation vectors satisfying the condition. That might be due to a too large value selected for k. The k should be well decided according to the size of the pre-stored material set, ensuring both high speed and high possibility to satisfy the condition in step c). In case of extreme conditions, if the step c) fails, this procedure should run again by dropping a hashing function, equivalent to combining a neighbor bin to that bin. Also, careful selection of the projecting vectors can also help to prevent this problem.

4. Experiments

4.1. Basic tests

We first test the basic performance of the two algorithms described above, under the unrestricted protocol of LFW, which evaluate verification accuracy. The ROC is shown in Fig. 3. It is not fascinating compared to the reported highly accurate algorithms, but considering its limited size, which indicates high speed in application, this accuracy is acceptable. At test phase, our model can process images at an average speed of 20 ms per image on CPU (Intel i7, single thread with SIMD optimization). Compared to the high dimensional LBP feature, which runs at about 240 ms per image, our model is much faster.

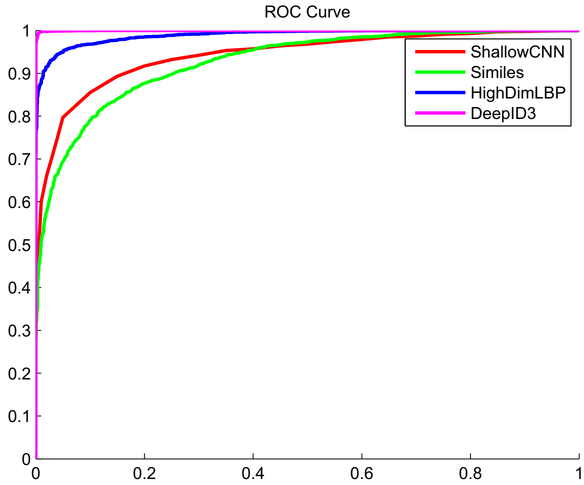


Fig. 3. ROC curve of our shallow CNN model on LFW. The AUC is 94.21%. We compare it with results of algorithms from [6,14,15], also on LFW. Due to the limited size of our model, the performance of the shallow CNN model is apparently lower than the deep models.

4.2. Classification with augmented data

To test the accuracy of these methods for identification task, we randomly pick subjects holding more than one samples from the LFW dataset to form the test set, so that for each classification query, there is at least one sample of the same subject in the example set. For the High-Dim LBP algorithm, we train a binary classification SVM, which takes two feature vectors as input, and determines whether they indicate a same identity. For the Shallow CNN, the 128-dim feature vectors extracted from the two test samples are directly compared by cosine distance. For data augmentation, we generate variation materials from the same dataset we use for training. The whole process takes 1 h 24 min. This time might varies according to hardware equality, hash bin number and random generation of projecting vectors.

For comparison, we first test the classification accuracy of the two models trained without augmentation data. As expected, the accuracy is only 72.353% and 74.572% for the Shallow CNN algorithm and High-dim LBP respectively.

To verify to which size of samples to augment best improves the accuracy of the two models, we conduct experiments of classification accuracy with different settings. For example, if we take 109 as the target size, for a subject with less than 109 samples, the sample size will be augmented to 109 samples; while for a subject with more than 109 samples, exactly 109 samples will be randomly picked from the sample set, in order to avoid unbalanced classification. Results of a series of this experiments with different resulting sample number are shown in Table 1. It can be found that a too high target size and too low target size can both harm the improvement of this augmentation method. A too low target size can reduce the original rich information of subjects with sufficient samples, while a too high target size might introduce too much virtual information that significantly lowers down the portion of real information, thus the possibility of introducing outliers increases largely. Note that this is only valid under our experiment settings. For other settings such as with different training sets or test sets, the best target augmentation size might vary.

However, our method does not work quite well if the original

Table 1
Classification accuracy augmenting sample number in each class to different level. Accuracy rises as augmentation number drops to reasonable extent, and falls back a little when augmentation is not sufficient.

Sample number	Shallow CNN classification accuracy	HighDim LBP Classification accuracy
500	71.235%	73.897%
236	72.532%	74.462%
144	73.139%	75.743%
121	77.560%	78.309%
109	84.202%	85.979%
77	90.809%	92.334%
49	93.876%	95.106%
29	93.932%	94.998%
14	84.336%	85.072%
7	82.239%	84.019%

training set has only one sample for each subject. We try using only one sample from each class for training, and test the SVM on the rest samples. The accuracy becomes 60.209% and 60.783% respectively. This is because Joint Bayesian analysis is more accurate on classes with more samples, and also under single sample condition, we are not able to utilize information from multi-sample classes, losing superiority over other sample augmentation algorithms.

5. Conclusion

In this work, we proposed a novel method to augment datasets to solve the problem that traditional classifiers like SVMs and Softmax cannot work well on training sets where some subjects have few or one sample. We used joint Bayesian Analysis to analyze the feature vectors of classes that need to be augmented, then applied an efficient algorithm to find reasonable variation vectors from a pre-stored material set. We have conducted experiments showing this kind of feature augmentation does work for the unbalanced training set condition, based on shallow CNN feature extraction and high dimensional LBP feature extraction.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (No.61472023) and the project of the State Key Laboratory of Software Development Environment (No.SKLSDE-2016ZX-24).

References

[1] B. Moghaddam, T. Jebara, A. Pentland, Bayesian face recognition, Pattern Recognit. 33 (99) (2000) 1771C1782.

[2] G.B. Huang, M. Mattar, T. Berg, E. Learned-Miller, Labeled faces in the wild: a database for studying face recognition in unconstrained environments, in: Proceedings of the Workshop on Faces in 'Real-Life' Images: Detection, Alignment, and Recognition.

[3] E. Zhang, Y. Li, F. Zhang, A single training sample face recognition algorithm based on sample extension, in: Proceedings of the 2013 Sixth International Conference on Advanced Computational Intelligence (ICACI), IEEE, 2013, pp. 324–327.

[4] B. Wang, F. Zhou, W. Li, Z. Li, Q. Liao, Combining specific learning and generic learning for single-sample face recognition, in: Proceedings of the 2012 5th International Congress on Image and Signal Processing (CISP), IEEE, 2012, pp. 1219–1223.

[5] C. Zhan, W. Li, P. Ogunbona, Face recognition from single sample based on human face perception, in: Proceedings of the 24th International Conference, Image and Vision Computing New Zealand, IVCNZ'09, IEEE, 2009, pp. 56–61.

[6] 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 Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2013, pp. 3025–3032.

[7] J. Lu, Y.-P. Tan, G. Wang, Discriminative multimanifold analysis for face recognition from a single training sample per person, IEEE Trans. Pattern Anal. Mach. Intell. 35 (1) (2013) 39–51.

[8] Q.-x. Gao, L. Zhang, D. Zhang, Face recognition using flda with single training image per person, Appl. Math. Comput. 205 (2) (2008) 726–734.

[9] Y. Su, S. Shan, X. Chen, W. Gao, Adaptive generic learning for face recognition from a single sample per person, in: Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2010, pp. 2699–2706.

[10] D. Chen, X. Cao, L. Wang, F. Wen, J. Sun, Bayesian face revisited: A joint formulation, in: Proceedings of the Computer Vision–ECCV 2012, Springer, 2012, pp. 566–579.

[11] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, arXiv:1409.4842.

[12] G.E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, R.R. Salakhutdinov, Improving neural networks by preventing co-adaptation of feature detectors, arXiv:1207.0580.

[13] S. Ren, X. Cao, Y. Wei, J. Sun, Face alignment at 3000 fps via regressing local binary features, in: Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2014, pp. 1685–1692.

[14] N. Kumar, A.C. Berg, P.N. Belhumeur, S.K. Nayar, Attribute and simile classifiers for face verification, in: Proceedings of the 2009 IEEE 12th International Conference on, Computer Vision, IEEE, 2009, pp. 365–372.

[15] Y. Sun, D. Liang, X. Wang, X. Tang, Deepid3: Face recognition with very deep neural networks, arXiv:1502.00873.

Biao Leng received the B.Sc. degree from the School of Computer Science and Technology, National University of Defense Technology, Changsha, China, in 2004, and the Ph.D. degree from the Department of Computer Science and Technology, Tsinghua University, Beijing, China, in 2009. He is an associate professor at the School of Computer Science and Engineering, Beihang University, Beijing, China. His current research interests include 3D model retrieval, image processing, pattern recognition, data mining and intelligent transportation systems.

Kai Yu is currently pursuing master's degree in the School of Computer Science and Engineering in Beihang University. His research is mainly on computer vision.

Jingyan Qin is the Full Professor at University of Science & Technology Beijing. She is the PhD Supervisor in Big Data Information Visualization and Interaction Design at Computer Science School of USTB, and she is the Director of HCI and Design for Sustainability Research Center at Industrial Design Department, USTB. Dr. QIN is selected as the Fellow of New Century Talents Plan of Ministry of Education of China in 2013. Dr. QIN's research and education focuses on data mining, interaction design, digital entertainment design and new media art.