Pattern Recognition 48 (2015) 3160–3168

Contents lists available at ScienceDirect

Pattern Recognition

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

Learning predictable binary codes for face indexing

Ran He [a,*], Yinghao Cai [b], Tieniu Tan [a], Larry Davis [c]

[a] The Center for Research on Intelligent Perception and Computing and the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, No. 95 ZhongGuanCun East St, HaiDian District, Beijing 100190, China
[b] Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
[c] The Institute for Advanced Computer Studies and the Department of Computer Science, University of Maryland, College Park, MD 20742, United States

ARTICLE INFO

Article history:
Received 26 September 2014
Received in revised form
29 January 2015
Accepted 17 March 2015
Available online 1 April 2015

Keywords:
Binary codes
Hashing
Face index
Large scale
Feature learning

ABSTRACT

High dimensional dense features have been shown to be useful for face recognition, but result in high query time when searching a large-scale face database. Hence binary codes are often used to obtain fast query speeds as well as reduce storage requirements. However, binary codes for face features can become unstable and unpredictable due to face variations induced by pose, expression and illumination. This paper proposes a predictable hash code algorithm to map face samples in the original feature space to Hamming space. First, we discuss the 'predictability' of hash codes for face indexing. Second, we formulate the predictable hash coding problem as a non-convex combinatorial optimization problem, in which the distance between codes for samples from the same class is minimized while the distance between codes for samples from different classes is maximized. An Expectation Maximization method is introduced to iteratively find a sparse and predictable linear mapping. Lastly, a deep feature representation is learned to further enhance the predictability of binary codes. Experimental results on three commonly used face databases demonstrate the superiority of our predictable hash coding algorithm on large-scale problems.

1. Introduction

A face recognition system aims to identify or verify one person by comparing an input facial image with registered images in a database. Computational requirements are aggravated by the high dimensionality of discriminative features [1] as well as the large number of registered people [2]. Many strategies, such as social network context [3], two-stage strategies [4], and cascade structures [5], have been applied to speed up training or searching in a large-scale face database.

Recently, hashing methods have drawn attention in large-scale image retrieval and face recognition, where the terminology 'hashing' refers to learning compact binary codes with Hamming distance computation. For image retrieval, similarity-sensitive hashing or locality-sensitive hashing algorithms [6–11], support vector machine [12,13], decision trees [14] and deep learning [15,16] have been studied to map high-dimensional data into a similarity-preserving low-dimensional Hamming space. Jegou et al. [17] used Hamming embedding to replace vector quantization in bag-of-feature construction. Wang et al. [18,19] introduced sequential projection learning and semi-supervised learning for hashing with compact codes. Biswas et al. [20] developed an efficient and robust algorithm to map shape features to a hash table. To discover or preserve the neighborhood structure in the data for compact codes, Liu et al. [21] and Kong et al. [22] presented a graph-based hashing method and a Manhattan Hashing method respectively. Based on similar and dissimilar data pairs, Gong and Lazebnik [23] developed an iterative procrustean approach to learning binary codes, and Liu et al. [24] further proposed a kernel-based supervised hashing model. In LDAHash, Strecha et al. [25] performed linear discriminant analysis (LDA) or difference of covariances on the descriptors before binarization. And for multi-view or cross-view retrieval, deep multi-view hashing [26], predictable dual-view hashing [27], co-regularized hashing [28], and collective matrix factorization hashing [29] were developed. For a brief review of binary hash codes for large-scale image search, refer to [16,27].

For face recognition, Ngo et al. [30] discretized the PCA coefficients of a face image to binary codes by using a bit-extraction method. In BioHashing methods [31,32], randomized dimension reduction or optimal linear transformations are generated to calculate the dot product of test features. Zeng et al. [33] addressed the hashing problem of high dimensional SIFT vectors based on the p-stable distribution locality sensitive hashing scheme. Shi et al. [34] built a connection between hashing kernels and compressed sensing, and applied hashing to speed up sparse representation based face recognition. Then Yan et al. [35] made use of a group of hashing function to learn similarity binary codes. Sattar et al. [36] proposed to use the 2-D discrete cosine transform and K-means clustering to learn hash codes. In addition, Wu et al.

[*] Corresponding author. Tel.: +86 10 82544641; fax: +86 10 82544485.
E-mail address: rhe@nlpr.ia.ac.cn (R. He).

http://dx.doi.org/10.1016/j.patcog.2015.03.016
0031-3203/© 2015 Elsevier Ltd. All rights reserved.

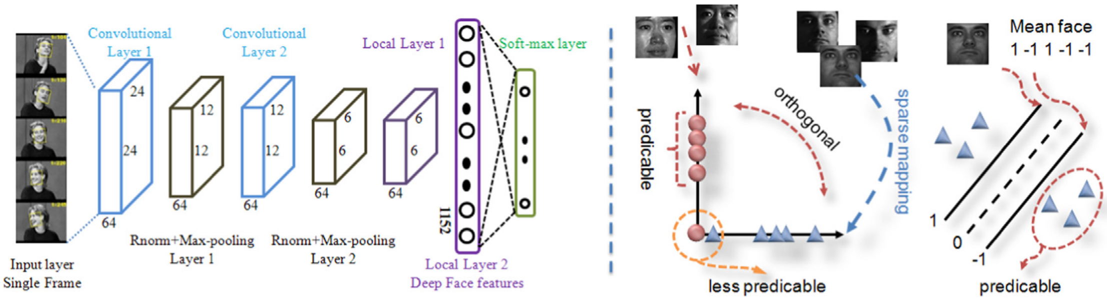

Fig. 1. The proposed scheme to learn predictable binary codes. 'Predictability' indicates that hash codes are predictable across facial variations. Our basic idea (right figure) is that the facial codes from one person are similar to the code of the mean face of this person with a predictable margin; meanwhile the codes of mean faces from different persons are significantly different and tend to be orthogonal (or low correlation [9,27]) to one another. Convolutional neural networks (left figure) are adopted to learn deep face features to improve the predictability of binary codes.

[2] and Chen et al. [37] resorted to hash codes for inverted indexing to speed up scalable face image retrieval.

Learning binary hash codes has been a key step to facilitate face recognition or image retrieval. And all the above hashing related methods indeed speed up retrieval or searching time. However, for face recognition, the hash codes learned from facial features tend to be unstable and unpredictable due to face variations induced by pose, expression and illumination. When one applies hashing methods to encode high-dimensional facial features, the learned codes should be predictable[1] across facial variations. This is due to the fact that two facial images and their corresponding binary codes from one person will generally not be the same. In addition, to the best of our knowledge, although several hashing methods have been used for face recognition, the fundamental question of what type of binary hash codes is good for face recognition has not been addressed.

This paper discusses the application of predictable hash codes to face indexing, and proposes a predictable hash code (PHC) learning scheme to embed high-dimensional dense facial features into Hamming space. First, based on the concept of linear discriminant analysis (or Fisherfaces in face recognition), we discuss the 'predictability' of hash codes for face indexing. Second, as illustrated in Fig. 1, we require that the distance between the codes from the same person (within-class distance) is minimized while the distance between the codes (between-class distance) from different classes is maximized. To achieve this goal, we relax the notion of within-class distance and between-class distance to the similarity of codes for images of the same person to the code of the mean face of that person and orthogonality (or low correlation [9,27]) of the codes of mean faces of different people, respectively. This allows us to formulate the predictable hash coding problem as a non-convex combinatorial optimization problem, which can be solved with an Expectation Maximization (EM) method to iteratively find a sparse and predictable linear mapping. Lastly, to further enhance the predictability of binary codes in real-world scenarios, convolutional neural networks (CNN) are adopted to learn a deep face representation. Experimental results on three commonly used face recognition databases demonstrate the superiority of our predictable hash coding algorithms on large-scale face indexing problems (the number of comparisons is larger than 900 million). Particularly on the YouTube Celebrities dataset, our proposed algorithms only use a 128-bits representation to achieve state-of-the-art results.

The rest of this paper is organized as follows. We discuss the 'predictability' of hash codes and present our predictable hash coding (PHC) algorithm in Section 2. Section 3 provides a series of experiments to validate our PHC algorithm, prior to summary in Section 4.

Table 1
Important notations used in this paper.

Notations	Descriptions
d	The number of features
n	The number of total training samples
C	The number of classes
k	The length of hash codes
X	The data matrix $X=[x_1,\dots,x_n]\in R^{d\times n}$
M	The mean face matrix $M\in R^{d\times c}$
H_w	Maximum within-class Hamming distance
H_b	Minimum between-class Hamming distance
H_m	Margin of codes H_b-H_w

2. Predictable hash codes

Unsupervised and supervised dimensionality reduction methods have been widely used in face recognition. But many methods do not scale to large datasets because their complexity is quadratic (or worse) in the number of data points [16]. Hence hashing has been used to improve query speeds and reduce storage costs. However, previous hashing based methods [30–32,25] often treat dimensionality reduction and hashing as two independent steps, which makes the learned binary codes less discriminative. In this section, we apply the concept of dimensionality reduction into Hamming space and study predictable hash codes for face indexing.

2.1. Problem formulation

Table 1 summarizes the notation needed to present the method. The generic learning problem of dimensionality reduction for face recognition is formulated as follows. Consider a dataset X from C classes, which consists of n samples x_i $(1\le i\le n)$ in a high-dimensional Euclidean space R^d. Each class has n_c samples with that set denoted as X^c. That is $X=[X^1,\dots,X^C]=[x_1,\dots,x_n]$. Let matrix M contain mean faces m_c. That is $M=[m_1,\dots,m_C]$. A dimensionality reduction method aims to learn a linear or non-linear mapping (or projection) matrix $W\in R^{d\times k}$ to project samples into a low-dimensional Euclidean space R^k.

One of the most widely used dimensionality reduction methods in face recognition is linear discriminant analysis (LDA). It maximizes a loss function that encourages a large separation between the projected class means while also encouraging a small variance within each class. Inspired by LDA, we define a within-class distance and a between-class distance for binary codes in Hamming space. H_w denotes the maximum within-class Hamming distance between any two codes for samples from the same class. H_b denotes the minimum between-class Hamming distance between any two codes from different classes. Given H_w and H_b, the margin of binary

[1] Predicability indicates that maximum within-class Hamming distance H_w is smaller than minimum between-class Hamming distance H_b.

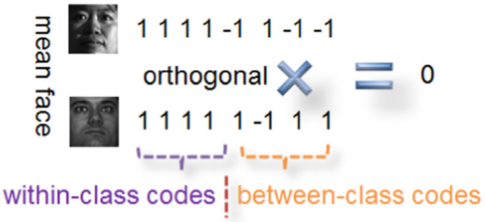

Fig. 2. Minimum and maximum Hamming distances of faces. If two binary codes from two mean faces are orthogonal (or have low correlation [9,27]), only half of their codes is different. In ideal conditions, one can use the half of one code to model within-class variation and the remaining half to separate different classes.

codes is defined to be $H_m = H_b - H_w$. Then, we propose the following definition of 'predictability' of hash codes for face recognition.

Definition 1. A hashing algorithm A is predictable for face variation if it satisfies $0 \le H_w^A < H_b^A \le k$ and has a margin H_m^A as large as possible.

Definition 1 makes facial codes from different people have a large separation in Hamming space. $H_w^A \ge H_b^A$ indicates that there may be a misclassification incurred by hashing algorithm A. As shown in Fig. 2, if the binary codes of the samples of one person are identical, H_w^A achieves the minimum value 0; and if the binary codes of any two samples from two different persons are entirely different, H_b^A achieves the maximum value k (the length of a binary code).

In face recognition, facial images from one person often have large variations induced by pose, expression and illumination so that their binary codes will vary. So penalizing within class codes based on their Hamming distances might be too strong a constraint for code learning. Instead, we enforce the constraint during code learning that the binary codes of all images of a given person are mostly similar to the code of the mean face of that person. That is, H_w^A is minimized resulting in a small variance of the binary codes within each class. This yields the following optimization criteria:

$$\min_{W_m}\sum_i\sum_j\|\mathrm{sgn}(W_m x_j^i)-\mathrm{sgn}(W_m m_i)\|_2^2, \tag{1}$$

where sgn(.) is the signum function. The mean face m_c in M is not a real face since it just corresponds to the center of one class. So we use different projection matrices for X and M. Then (1) becomes

$$\min_{W_v,W_m}\sum_i\sum_j\|\mathrm{sgn}(W_v x_j^i)-\mathrm{sgn}(W_m m_i)\|_2^2. \tag{2}$$

Note that mean faces are commonly used in face recognition. Eigen faces (PCA) and Fisher faces (LDA) involve in the computation of mean faces. The motivation of (2) to use mean faces is based on LDA. In [24], anchor points are used to reduce the computational costs of the calculation of graph Laplacian eigenvectors. Anchor points measure similarities of all database points. The cluster centers of K-means are often utilized as anchor points. In contrast to anchor points, the purpose of the mean face of each class in (2) is to reduce the within-class variation of the binary codes of one class.

As shown in Figs. 1 and 2, we also expect that there is a large separation between the codes of different classes. Due to the large within-class variation, it is not possible to ensure that each bit of the binary codes from different persons is different. And it is impossible to make the codes of mean faces of different people to be entirely different due to the large number of classes. Hence, we relax the large margin constraint between different people and only require that the codes of mean faces are orthogonal (or have low correlation [9,27]). Adding this constraint to (2) results in the following form:

$$\begin{aligned}&\min_{W_v,W_m}\ \|W_v^T X - B_m\|_2^2 + \|W_m^T M - B_v\|_2^2 + \|B_m B_m^T - I\|_2^2\\ &\quad \text{s.t.}\ \ B_v = \mathrm{sgn}(W_v^T X),\quad B_m = \mathrm{sgn}(W_m^T M)\end{aligned} \tag{3}$$

Obviously, the $\|B_m B_m^T - I\|_2^2$ loss term[2] cannot make the hashing algorithm achieve the largest separation, as shown in Fig. 2. Let B_m^j be the jth column of B_m. If we directly enforce that the distances between the binary codes B_m^j of all mean faces to be large, we will have the following problem:

$$\max_{B_m}\sum_{j\ne k}\|B_m^j - B_m^k\|_0 \tag{4}$$

where $\|.\|_0$ is the counting norm (i.e., the number of nonzero entries in a vector or matrix). However, solving $\|.\|_0$ is NP-hard so that (4) is difficult to minimize. Hence, we make use of the orthogonality constraint to obtain an approximate solution. We will show that this orthogonality constraint is suitable for separating different people. That is, we can use some bits in a binary code to capture within-class variations and employ the remaining bits to ensure that different classes have a safe margin.

In addition, during binarization, we need to binarize a learned real value to a bit (-1 or 1). If this real value is around zero, a small face variation will change the sign of this binary value, which makes the learned bit unstable and unpredictable. As shown in the right figure in Fig. 1, we expect that learned real values have a margin to -1 or 1 so that a binary value is not changed under small face variations during binarization. Finally, a sparse projection is learned to project high-dimensional dense features to Hamming space so that the computational cost of the projection is reduced as in [1]. Hence we have the following minimization problem:

$$\begin{aligned}&\min_{W_v,W_m}\ \sum_{l\in\{v,m\}}(C_l\xi_l + \|W_l\|_1) + \|B_m B_m^T - I\|_2^2\\ &\quad \text{s.t.}\ \ B_v = \mathrm{sgn}(W_v^T X_v),\ B_m = \mathrm{sgn}(W_m^T M)\\ &\qquad B_m^{ij}(w_{vi}^T X^j) \ge 1 - \xi_v^{ij},\quad B_v^{ij}(w_{mi}^T M^j) \ge 1 - \xi_m^{ij}\\ &\qquad \xi_v \ge 0 \ \text{ and } \ \xi_m \ge 0\end{aligned} \tag{5}$$

For any person, the third and fourth constraints in (5) make each bit of its samples predictable with respect to that of that person's mean face.

Algorithm 1. Predictable Hash Coding (PHC).

```
Input: data matrix $X \in R^{d\times n}$ and code length $k$.
Output: $W_v \in R^{d\times k}$ and $B_v \in R^{n\times k}$
Compute mean face $m_c$ for each class.
$W_v \leftarrow PCA(X,k)$; $W_m \leftarrow PCA(M,k)$
$B_v \leftarrow \mathrm{sgn}(W_v X)$; $B_m \leftarrow \mathrm{sgn}(W_m M)$
repeat
    $W_v \leftarrow$ Weights of $k$ linear SVMs
    $B_v \leftarrow \mathrm{sgn}(W_v X)$
    $W_m \leftarrow$ Weights of $k$ linear SVMs
    $B_m \leftarrow \mathrm{sgn}(W_m M)$
Update $B_m$ according to (6) and $B_m \leftarrow \mathrm{sgn}(B_m)$
until Converges
$B_v \leftarrow \mathrm{sgn}(W_v X)$
```

2.2. Solution

The minimization problem in (5) is a non-convex combinatorial optimization problem and hence it is difficult to minimize.

[2] When $k < C$, $\|B_m B_m^T - I\|_2^2$ is used in (3); and when $k \ge C$, $\|B_m^T B_m - I\|_2^2$ is used in (3). In addition, other separation constraints, such as the constraints in [11], can be also used in place of the orthogonality constraint.

Fortunately, (5) can be efficiently solved by an Expectation Maximization (EM) iterative algorithm as in [27]. An overview of our iterative algorithm is as follows.

First, we fix variables W_m, B_m and ξ_m, and solve for variables W_v and ξ_v, which is a multiple linear SVM problem (one for each bit) [38]. To learn the ith SVM,[3] the columns of X and the elements of the ith row of B_m are used as training data and labels respectively. Then, we use the output weights of these SVMs, W_v, to compute a new $B_v = \text{sgn}(W_v^T X)$. Second, by fixing W_v, B_v and ξ_v, we can use the same approach to update W_m and B_m. Third, we update B_m to minimize the correlation between bits via minimizing $\|B_m B_m^T - I\|_2^2$. Because this problem is not trivial to solve, we use spectral relaxation [9,27] by creating a Gram matrix $S = B_m^T B_m \in R^{C\times C}$ [4] and a diagonal matrix $D(i,i) = \sum_j S(i,j)$, resulting in the following relaxed problem:

$$\min_{B_m} \quad \text{tr}(B_m(D-S)B_m^T)$$

$$\text{s.t.} \quad B_m B_m^T = I \tag{6}$$

The solutions of (6) are the k eigenvectors of $D-S$ w.r.t. minimal eigenvalues, which we binarize by taking the sign of the elements. Note that when the number of bits k is larger than the number of classes C, we simply use Gram–Schmidt process to obtain an orthogonal solution of B_m. All three steps are repeated until convergence of the objective function. As in [27], principal component analysis (PCA) and canonical correlation analysis (CCA) can be used to initialize W_v and W_m. The details of the proposed algorithm are summarized in Algorithm 1.

2.3. Learning a deep feature representation

Feature representation is a key ingredient for face recognition systems [39]. A discriminative feature representation will significantly improve the predictability of binary codes. Recent advance in Feature Learning [40] shows that deeply learned features with convolutional neural networks (CNN) [41] can greatly improve the performance of many tasks in computer vision [42,43]. Benefiting from CNN's deep architecture and supervised learning approach, CNN's can deal with large amounts of data and generate a compact and effective feature representation. It has far fewer parameters compared to its standard feed-forward neural network counterparts, so that it is much easier to train in practice.

To improve the predictability of hashing methods, we learn deep features for face images with a CNN. We employ the CNN network proposed by Alex[5] to accommodate our deep architecture with the intrinsic characteristics of video face images. This CNN first feeds gray scale images to two convolutional layers, each followed by a normalization layer and a max-pooling layer. Then, two locally connected layers are connected to the output of the second max-pooling layer, and finally a C-way soft-max regression layer (C is the number of classes) that produces a distribution over class labels. The details of the network are illustrated in Fig. 1. The input of this network is the cropped gray scale face images; no preprocess is implemented. The last C-way soft-max regression layer provides supervised information for learning of face representations. The outputs of the last locally connected layers are employed as face representations.

3. Experiments

In this section, we present some quantitative results on three common face datasets [44–46] to highlight the benefits of our proposed predictable hash code algorithm. Note that the aim of hash coding is to save computational cost and storage space rather than to improve recognition accuracy (or to solve challenging face recognition problems). Hence the first two datasets [44,45] are used to systematically evaluate different hashing algorithm for face indexing problems. The last dataset [46] is used to perform large-scale experiments, in which the average number of comparisons is more than 900 million.

We implement two models of Eq. (5), i.e., L2-norm and L1-norm on W_l. When the L2-norm is used, a dense mapping W_l is learned; and when the L1-norm is used, a sparse mapping W_l is learned. We denote these two versions as PHC-L2 and PHC-L1. We compare our methods with popular hashing methods, including Locality Sensitive Hashing (LSH) [47],[6] Spectral Hashing (SH) [9],[7] Iterative Quantization (ITQ) [23],[8] Linear Discriminative Analysis Hash (LDAH) [25], Binary Reconstruction Embedding (BRE) [6],[9] Kernel-Based Supervised Hashing (KSH) [24],[10] and Fast Supervised Hashing (FastH) [14].[11] For ITQ, its supervised version (CCA-ITQ) and unsupervised version (PCA-ITQ) are included. PCA is used as a preprocessing step for CCA-ITQ. Since LDA can only find a $C-1$ dimensional subspace, we only report the results of LDAH when the number of bits is fewer than $C-1$. Since the biohashing method in [32] and the random projection hashing method in [2] are similar to LSH and obtain similar results to LSH, we only report the results of LSH.

3.1. Results on the FRGC dataset

We collected facial images from a subset of the most challenging FRGC version 2 face database [45]. There are 8014 images of 466 subjects in the query set for FRGC experiment 4. These uncontrolled images contain variations in illumination, expression, time, and blurring. We take the first 20 facial images of each subject if the number of facial images is more than 20. Accordingly, we obtain 3720 facial images of 186 subjects. Each facial image is cropped to size 32 × 32 as shown in Fig. 4. The down-sampled facial images are directly used as facial features as in [48,49].

We consider two scenarios and use the first 10 images of each subject from the first 100 subjects as the training set. (1) In the first closed-set scenario,[12] we directly take the training set as the gallery set and take the remaining 10 images of each subject from the first 100 subjects as the probe set. Hence the subjects in the training and testing sets (including probe and gallery sets) are the same. (2) In the second open-set scenario, we take the first 10 facial images of each person in the last 86 subjects as the gallery set and the remaining 10 images as the probe set. Hence, the subjects in the training and testing sets are different.

Fig. 3(a) and (b) shows experimental results on these two scenarios. Since there are 100 classes in the training set, the minimum number of bits needed to separate different classes is 7. We observe that FastH achieves the highest recognition rate on the close-set scenario (Fig. 3(a)), and PHC-L1 and PHC-L2 methods achieve the highest recognition rates on the open-set scenario

[3] The ℓ_1 regularized linear SVM is implemented by lIBLINEAR: http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[4] When there are C classes, the total number of different mean faces in B_m is also C.

[5] https://code.google.com/p/cuda-convnet/.

[6] http://www.mit.edu/~andoni/LSH/.

[7] http://www.cs.huji.ac.il/~yweiss/SpectralHashing/.

[8] http://www.unc.edu/~yunchao/itq.htm.

[9] http://www.cse.ohio-state.edu/~kulis/pubsbytype.htm.

[10] http://www.ee.columbia.edu/~wliu/.

[11] https://bitbucket.org/chhshen/fasthash/.

[12] The closed-est scenario often occurs in face verification or retrieval on a family album or social network.

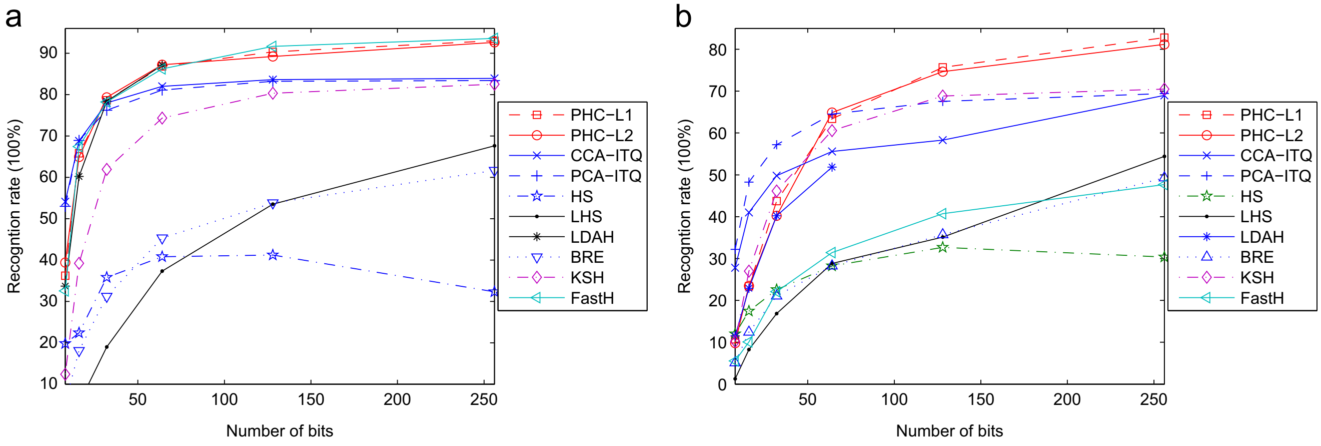

Fig. 3. Recognition accuracy as a function of the number of bits on the FRGC dataset. (a) Recognition rates of different methods on the training set. (b) Recognition rates of different methods on an open-set dataset.

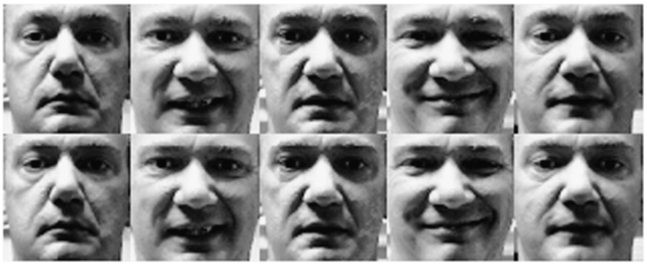

Fig. 4. Cropped facial images of one subject in the FRGC database.

(Fig. 3(b)). In addition, when different subjects are used in the testing set, all compared methods need more bits to achieve their highest recognition rates. Particularly, the recognition rates of FastH decrease significantly when FastH is applied to the open-set scenario. This may be while the boosted decision trees of FastH can accurately capture the non-linear structure of a training set, when new subjects (or classes) are given, the learned structure does not capture the structure of new subjects. When the number of bits is 256, PHC-L1 and PHC-L2 obtain the highest recognition rate in Fig. 3(b). It seems that PHC-L1 and PHC-L2 methods need more bits to achieve higher recognition rates than PCA-ITQ. This may be because when the number of bits is small, the two PHC methods over-learn on the training set. We also observe that CCA+ITQ does not perform well in this data set. This may be due to the large variations of the face images in the FRGC database so that CCA+ITQ over-learns on the training set.

3.2. Results on the AR dataset

The AR database [44] is composed of over 4000 facial images including different facial variations – facial expressions (neutral, smile, anger, and scream), illumination variations (left light on, right light on, and all side lights on), and occlusion by sunglasses or scarf. These images are from 126 subjects (70 men and 56 women). For each subject, 26 facial images are taken in two separate sessions. In this section, we selected a subset of the data set consisting of 100 subjects as in [48]. Eight frontal images of each subject are used, as shown in Fig. 5. The gray-scale images were down-sampled to 28×23.

We make use of the first $50 \times 8 = 400$ images from the first 50 subjects as the training set. Then we use the 400 images from the remaining 50 subjects as the testing set. Hence the subjects in the training and testing set are entirely different. In the test set, the first four images of each subject in the first session are used as the probe set and the last four images in the second session are used as the gallery set.

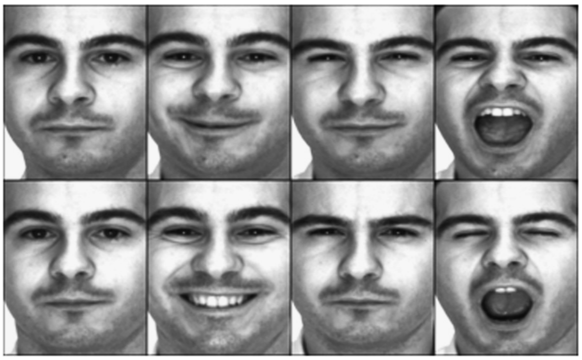

Fig. 5. Cropped facial images of the first subject in the AR database. The images in the first and second row are from the first and second session respectively.

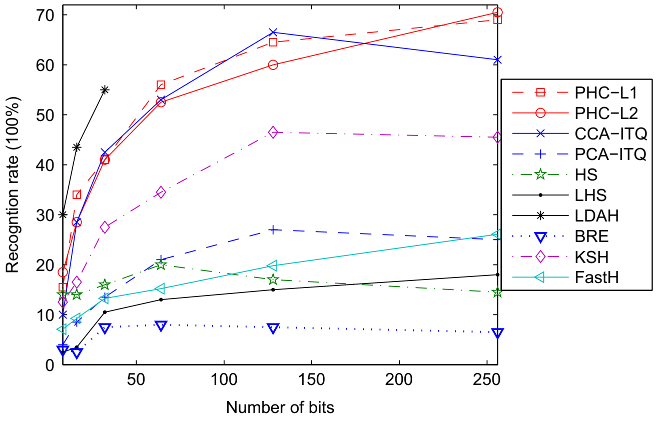

Fig. 6. Recognition accuracy as a function of the number of bits on the AR dataset.

Fig. 6 shows the recognition accuracy as a function of the number of bits on the AR dataset. We observe that four supervised hashing methods, including LDAH, CCA-ITQ, PHC-L1 and PHC-L2, perform better than other unsupervised ones. This is because the frontal images from two sessions in the AR database have small illumination and pose variations so that supervised hashing methods can learn discriminative hyperplanes to map binary codes. However, when the number of bits increases, the recognition rate of CCA-ITQ increases slowly. We observe that this phenomenon is consistent with the results observed in image

Fig. 7. Cropped facial images of three different subjects in the YouTube Celebrities dataset.

retrieval [50]. That is, CCA often works well in low-dimensional space. And LDAH only works when the number of bits is smaller than the number of classes. In addition, the recognition rates of all methods increase as the number of bits increases. The highest recognition rate is only around 70%. It seems that a larger number of bits is needed to achieve the highest recognition rate when hashing methods are applied to an open-set problem.

3.3. *Results on the YouTube celebrities dataset*

The YouTube Celebrities face tracking and recognition dataset [46] contains 1910 video clips of 47 human subjects from the YouTube website, which is often used to evaluate the performance of different video based face recognition methods. Roughly 41 clips were segmented from three unique videos for each person. Each facial image is cropped to size 30 × 30 as shown in Fig. 7. This dataset is challenging because it contains a lot of noise, pose and facial variations (see Fig. 7).

Following the standard setup, the testing dataset is composed of six test clips, two from each unique video, per person. The number of face images in the testing set is 44,172. The remaining 239,997 were used as the input to the CNN to learn a 1152-D feature representation. Part of the remaining dataset is used, and each time, only one frame of video (one single image) is fed into the CNN. Image flip is implemented to augment the data. To train hashing algorithms, we randomly selected three training clips, one from each unique video. All experiments are averaged over 10 runs. The average number of the training set for hashing algorithm is larger than 20,000. Hamming distance is computed on each pair of face samples in the training and testing set. As a result, the average number of comparisons is more than 900 million. The nearest neighbor classifier was adopted to classify each image in the testing set. Since each test clip contains many face images, we made use of the class label of the majority class in a clip as the final class label of the clip.

Fig. 8 plots the average recognition rates of different hashing algorithms over 10 runs. Since the computational costs of BRE are too large, we only report its results when the number of bits is smaller than 16. The ‘baseline’ indicates that we directly make use of CNN features. The recognition rate of the baseline is 68.79%. We observe that PHC-L2, PCA-ITQ, SH and LSH methods can further improve recognition rates based on the learned CNN features. As expected, our PHC-L2 algorithm performs better than its competitors and achieves the highest recognition rates at all numbers of bits.

To further demonstrate the effectiveness and predictability of the proposed hashing method, we also compare ours against the state-of-the-art methods on the YouTube Celebrities dataset, including discriminative canonical correlations (DCC) [51], manifold discriminant analysis (MDA) [52], sparse approximated nearest point (SANP) [53], sparse representation for video (SRV) and its kernelized version KSRV [54], covariance discriminative learning (Cov+PLS) [55], jointly learning dictionary and subspace structure (JLDSS) [56], image sets alignment (ImgSets) [57], regularized nearest points (RNP) [58], and mean sequence sparse representation-based classification (MSSRC) [59]. The recognition rates for other competing methods are cited directly from their papers. Fig. 9 plots the recognition rates of our PHC method and its competitors.

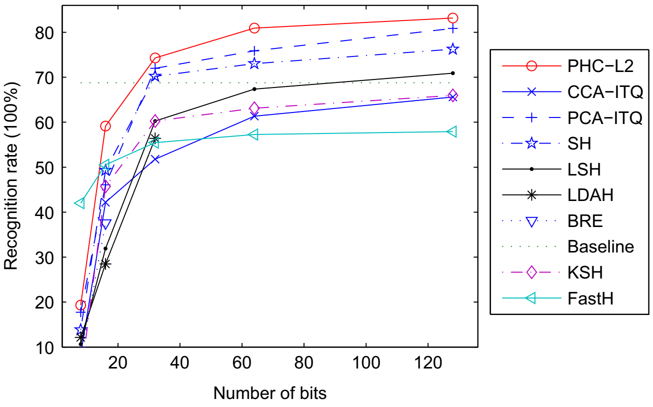

Fig. 8. Recognition accuracy as a function of the number of bits on the YouTube Celebrities dataset.

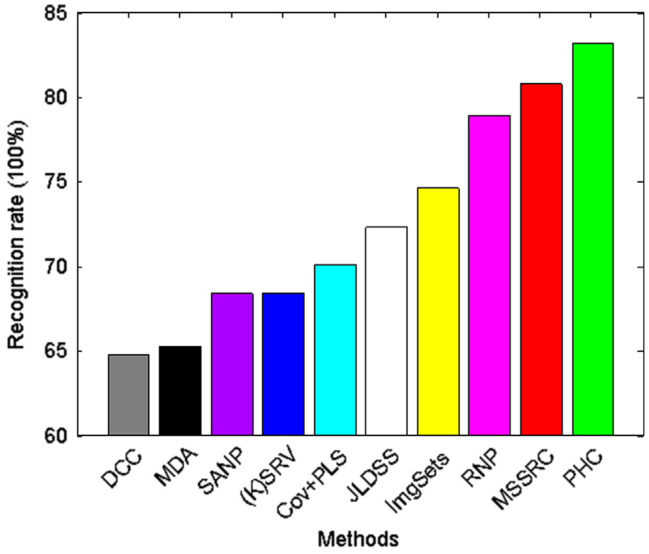

Fig. 9. Recognition rates of the competing methods. We cited the recognition rates from the literature.

From Fig. 9, we observe that the methods can be ordered in ascending recognition rates as DCC, MDA, SANP, (K)SRV, Cov+PLS, JLDSS, ImageSets, RNP, MSSRC and PHC. The recognition rates of MSSRC and PHC are 80.75% and 83.15% respectively. Since two ImageSets methods [57,58] efficiently make use of image set techniques to deal with the face samples in video, their classifier is more suitable for video based face recognition than the other methods so that they obtain the third best result. Although the simple nearest neighbor classifier with voting is used as the classifier in PHC to report recognition rates, PHC achieves the highest recognition rate. This improvement of recognition rates against its competitors mainly derives from predictable hash codes and the deep feature representation. Note that since our PHC

Table 2
Predictability (average accuracy ± standard deviation) of PHC with different feature representations.

Number of bits	16	32	64	128
Pixels	21.13 ± 3.01	46.31 ± 1.43	54.82 ± 2.68	58.37 ± 4.36
CNN	59.13 ± 2.10	74.29 ± 3.20	80.94 ± 1.65	83.15 ± 0.68

produces binary codes, any other strong classifiers can be further used to improve recognition rates. Experimental results on the YouTube Celebrities dataset demonstrate that the proposed predictable scheme (deep representation plus hash coding) is effective for face indexing and results in state-of-the-art accuracy.

Table 2 shows average recognition rates and standard deviation of PHC with two feature representations. A higher recognition rate with a lower standard deviation indicates that the learned binary codes are more predictable. We observe that when face images (pixels) are directly used as features, the highest recognition rate of PHC is only 58.37%, which is much lower than the rate of CNN features (68.79%). We regard these results as reasonable on the YouTube dataset. Since the YouTube dataset is a challenging dataset, face recognition methods often resort to robust trackers [59] or feature representations (e.g., LBP, HOG and Gabor wavelets) to improve recognition rates. We also observe that as the number of bits increases, the average recognition rate of PHC+CNN increases while the standard deviation of PHC+CNN decreases. This indicates that our PHC+CNN scheme becomes more stable and predictable when the number of bits increases, which is consistent with our orthogonality assumption (i.e., some redundant bits may be potentially useful for unpredictable variations).

3.4. Discussion

Code length vs. recognition rate: One merit of the proposed hash coding methods is that they can further improve recognition rates as the number of bits increases. In general, the more bits are used, the better the approximation is [2]. In addition, compared with some hashing methods [11], our methods do not need additional steps for learning longer codes. However, compared with CCA+ITQ and LDAH, one limitation of our methods seems to need a larger number of bits to achieve a higher recognition rate. We consider that this phenomenon is consistent with our assumption in Section 2.1. That is, some redundant bits may be potentially useful for face variations because face variations are often unpredictable in the real world. A basic issue of hashing methods is how many bits are sufficient to achieve a high face recognition rate. And another issue is to study other efficient constraints on B_m that enforce a larger margin.

L2-norm vs. L1-norm: In the previous two experiments, it seems that a sparse mapping (L1-norm on W_l in Eq. (5)) works slightly better than a dense mapping (L2-norm on W_l in Eq. (5)). In face recognition, facial features are often high-dimensional and dense so that a sparse mapping (or projection) is often helpful to save computational cost and storage space [1]. Hence our proposed L1-norm based method is applicable to real world face recognition systems. A future issue is to further validate our proposed methods to encode high-dimensional local descriptors (such as Gabor and SIFT) on large scale database (such as LFW [60]).

Closed-set vs. open-set: Face recognition is an open set problem. It is impossible to collect facial images for each person to perform learning. Experimental results show that when new persons are used, the accuracy of hashing methods drops. Particularly, FastH achieves the highest recognition rate on the close-set scenario (Fig. 3(a)) whereas its recognition rates drop significantly on the open-set scenario (Fig. 3(b)). For a closed-set problem, hashing methods only need a small number of bits to achieve a high recognition rate; but for an open set problem, they often need a larger number of bits to achieve a stable recognition rate. This indicates that a larger number of bits can model more intra-class facial variations. Hence it is recommended to use more bits to encode face features for face recognition. And it would also be of interest to make hash coding methods work with other face recognition techniques, such as pose correction and illumination removal, to reduce intra-class variations.

Feature learning vs. predictability: Learning discriminative feature representations plays an important role in face recognition. A good feature representation can significantly improve face recognition accuracy as well as the predictability of binary codes. As illustrated in Section 3.3, when low-resolution face images are used as the feature representation of PHC, the best recognition rate of PHC is only 58.37%. In contrast, when the CNN learned features are used, PHC's recognition rates are improved at all numbers of bits. However, it is difficult for a feature representation to entirely separate all people because facial variations are often large in real-world scenarios. The recognition rate of the learned CNN features is only 68.79% on the YouTube dataset. Hence, when mapping feature representations to binary codes, we still need a predictable hashing algorithm to make the learned binary codes discriminative in Hamming space. When our PHC is adopted, the highest recognition rate of 83.15% is achieved at 128 bits. The experimental results on the real-world YouTube dataset support the use of the scheme of feature learning and hashing to improve the predictability of binary codes in practice.

4. Conclusion and future work

We introduced the problem of predictable hash coding for face indexing, and developed a predictable hash coding algorithm to map face samples in the original feature space into Hamming space. The code problem is formulated as a non-convex combinatorial optimization problem, in which the distance between intra-class codes is minimized while the distance between extra-class codes is maximized. An Expectation Maximization method was developed to iteratively find a sparse and predictable linear mapping. A deep feature representation was also learned to further improve the 'Predictability' of hash codes. Experimental results on three face recognition datasets show that our proposed predictable hash coding algorithm can outperform other hash coding methods on large-scale face indexing problems, and obtains state-of-the-art results on the YouTube Celebrities dataset.

In video based face recognition, there are often a large number of face images in a single video. Although hashing method can learn binary codes to reduce computation costs, the costs will still tend to be large if all face images in videos are compared. In the future, one potential direction is to select or learn representative face samples to represent a face video.

Conflict of interest

None declared.

Acknowledgment

The authors would like to greatly thank the associate editor and the reviewers for their valuable comments and suggestions, and Dr. Shu Zhang for implementing the codes of deep learning. This work is funded by National Basic Research Program of China

(Grant no. 2012CB316300) and National Natural Science Foundation of China (Grant nos. 61473289 and 61135002).

References

[1] D. Chen, X. Cao, F. Wen, J. Sun, Blessing of dimensionality: high-dimensional feature and its efficient compression for face verification, in: CVPR, 2013.
[2] Z. Wu, Q. Ke, J. Sun, H.-Y. Shum, Scalable face image retrieval with identity-based quantization and multireference reranking, IEEE TPAMI 33 (10) (2011) 1991–2001.
[3] Z. Stone, T. Zickler, T. Darrell, Toward large-scale face recognition using social network context, Proc. IEEE 98 (8) (2010) 1408–1415.
[4] R. He, W.-S. Zheng, B.-G. Hu, X.-W. Kong, Two-stage nonnegative sparse representation for large-scale face recognition, IEEE TNNLS 24 (1) (2013) 35–46.
[5] D. Yi, Z. Lei, Y. Hu, S. Z. Li, Fast Matching by 2 Lines of Code for Large Scale Face Recognition Systems, Technical Report, CASIA, arXiv:1302.7180, 2013.
[6] B. Kulis, T. Darrell, Learning to hash with binary reconstructive embeddings, in: NIPS, 2009, pp. 1042–1050.
[7] B. Kulis, K. Grauman, Kernelized locality-sensitive hashing for scalable image search, in: ICCV, 2009.
[8] M. Raginsky, S. Lazebnik, Locality-sensitive binary codes from shift-invariant kernels, in: NIPS, 2009.
[9] Y. Weiss, A. Torralba, R. Fergus, Spectral hashing, in: NIPS, 2009, pp. 1753–1760.
[10] M.M. Bronstein, A.M. Bronstein, F. Michel, N. Paragios, Data fusion through cross-modality metric learning using similarity-sensitive hashing, in: CVPR, 2010.
[11] K. He, F. Wen, J. Sun, K-means hashing: an affinity-preserving quantization method for learning binary compact codes, in: CVPR, 2013.
[12] Y. Mu, G. Hua, W. Fan, S.-F. Chang, Hash-svm: Scalable kernel machines for large-scale visual classification, in: CVPR, 2014.
[13] G. Lin, C. Shen, J. Wu, Optimizing ranking measures for compact binary code learning, in: ECCV, 2014.
[14] G. Lin, C. Shen, Q. Shi, A. van den Hengel, D. Suter, Fast supervised hashing with decision trees for high-dimensional data, in: CVPR, 2014.
[15] G. Hinton, R. Salakhutdinov, Discovering binary codes for documents by learning deep generative models, Cognit. Sci. 411 (2010) 1–18.
[16] K. Grauman, R. Fergus, Learning binary hash codes for large-scale image search, Mach. Learn. Comput. Vis. 411 (2013) 49–87.
[17] H. Jegou, M. Douze, C. Schmid, Product quantization for nearest neighbor search, IEEE TPAMI 33 (1) (2011) 117–128.
[18] J. Wang, S. Kumar, S.F. Chang, Sequential projection learning for hashing with compact codes, in: ICML, 2010, pp. 1753–1760.
[19] J. Wang, S. Kumar, S.F. Chang, Semi-supervised hashing for scalable image retrieval, in: CVPR, 2010.
[20] S. Biswas, G. Aggarwal, R. Chellappa, An efficient and robust algorithm for shape indexing and retrieval, IEEE Trans. Multimed. 12 (5) (2010) 372–385.
[21] W. Liu, J. Wang, S. Kumar, S.-F. Chang, Hashing with graphs, in: ICML, 2011.
[22] W. Kong, W.-J. Li, M. Guo, Manhattan hashing for large-scale image retrieval, in: SIGIR, 2012.
[23] Y. Gong, S. Lazebnik, Iterative quantization: a procrustean approach to learning binary codes, in: CVPR, 2011.
[24] W. Liu, J. Wang, R. Ji, Y. Jiang, S. Chang, Supervised hashing with kernels, in: CVPR, 2012, pp. 2074–2081.
[25] C. Strecha, A.M. Bronstein, M.M. Bronstein, P. Fua, LDAHash: improved matching with smaller descriptors, IEEE TPAMI 34 (1) (2012) 1–14.
[26] Y. Kang, S. Kim, S. Choi, Deep learning to hash with multiple representations, in: ICDM, 2012, pp. 930–935.
[27] M. Rastegari, J. Choi, S. Fakhraei, H. Daume, L.S. Davis, Predictable dual-view hashing, in: ICML, 2013.
[28] Y. Zhen, D. Yeung, Co-regularized hashing for multimodal data, in: CVPR, 2012.
[29] G. Ding, Y. Guo, J. Zhou, Collective matrix factorization hashing for multimodal data, in: CVPR, 2014.
[30] D.C.L. Ngo, A.B.J. Teoh, A. Goh, Eigenspace-based face hashing, in: Biometric Authentication, 2004, pp. 195–199.
[31] A.T.B. Jin, D.N.C. Ling, A. Goh, Biohashing: two factor authentication featuring fingerprint data and tokenised random number, Pattern Recognit. 37 (2004) 2245–2255.
[32] A. Kong, K.-H. Cheung, D. Zhang, M. Kamel, J. You, An analysis of biohashing and its variants, Pattern Recognit. 39 (7) (2006) 1359–1368.
[33] Z. Zeng, T. Fang, S. Shah, I.A. Kakadiaris, Local feature hashing for face recognition, in: BTAS, 2009, pp. 119–126.
[34] Q. Shi, H. Li, C. Shen, Rapid face recognition using hashing, in: CVPR, 2010, pp. 2753–2760.
[35] J. Yan, Z. Lei, D. Yi, S. Z.Li, Towards incremental and large scale face recognition, in: IJCB, 2011, pp. 1–6.
[36] D. Sattar, S. Mohsin, K. Ayub, M. Raza, S. Mohsin, Enhanced and fast face recognition by hashing algorithm, J. Appl. Res. Technol. 10(4) (2012) 607–617.
[37] B.-C. Chen, Y.-Y. Chen, Y.-H. Kuo, W. H. Hsu, Scalable face image retrieval using attribute-enhanced sparse codewords, IEEE Trans. Multimed. 15(5) (2013) 1163–1173.
[38] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, C.-J. Lin, Liblinear: a library for large linear classification, JMLR 9 (2008) 1871–1874.
[39] Z. Lei, R. Chu, R. He, S. Liao, S. Z. Li, Face recognition by discriminant analysis with Gabor tensor representation, in: IAPR/IEEE International Conference on Biometrics, 2007.
[40] Y. Bengio, A. Courville, P. Vincent, Representation learning: a review and new perspectives, TPAMI 35 (8) (2013) 1798–1828.
[41] Y.L. Cun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, L. D. Jackel, Hand-written digit recognition with a back-propagation network, in: NIPS, 1990.
[42] Y. Taigman, M. Yang, M. A. Ranzato, L. Wolf, Deepface: Closing the gap to human-level performance in face verification, in: CVPR, 2014.
[43] Y. Sun, X. Wang, X. Tang, Deep learning face representation from predicting 1000 classes, in: CVPR, 2014.
[44] A.M. Martinez, R. Benavente, The AR Face Database, Computer Vision Center (CVC), Technical Report.
[45] P.J. Phillips, P.J. Flynn, T. Scruggs, K. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, W. Worek, Overview of the face recognition grand challenge, in: CVPR, 2005.
[46] M. Kim, S. Kumar, V. Pavlovic, Rowley, Face tracking and recognition with visual constraints in real-world videos, in: CVPR, 2008.
[47] A. Gionis, P. Indyk, R. Motwani, Similarity search in high dimensions via hashing, in: VLDB, 1999.
[48] J. Wright, A.Y. Yang, A. Ganesh, S.S. Sastry, Y. Ma, Robust face recognition via sparse representation, IEEE TPAMI 31 (2) (2009) 210–227.
[49] R. He, W.-S. Zheng, B.-G. Hu, Maximum correntropy criterion for robust face recognition, IEEE TPAMI 33 (8) (2011) 1561–1576.
[50] K. Wang, R. He, W. Wang, L. Wang, T. Tan, Learning coupled feature spaces for cross-modal matching, in: ICCV, 2013.
[51] T. Kim, O. Arandjelovic, R. Cipolla, Discriminative learning and recognition of image set classes using canonical correlations, IEEE TPAMI 29 (2007) 1005–1018.
[52] R. Wang, S.G. Shan, X.L. Chen, W. Gao, Manifold-manifold distance with application to face recognition based on image set, in: CVPR, 2008.
[53] Y. Hu, A. Mian, R. Owens, Sparse approximated nearest points for image classification, in: ICCV, 2011.
[54] Y.C. Chen, V. Patel, S. Shekhar, R. Chellappa, P. Phillips, Sparse approximated nearest points for image classification, in: IEEE Conference on Automatic Face and Gesture Recognition, 2013.
[55] R. Wang, H. Guo, L. Davis, Q. Dai, Covariance discriminative learning: a natural and efficient approach to image set classification, in: CVPR, 2012.
[56] G. Zhang, R. He, L. Davis, Jointly learning dictionary and subspace structure for video-based face recognition, in: ACCV, 2014.
[57] Z. Cui, H. Zhang, S. Lao, X. Chen, Image sets alignment for video-based face recognition, in: CVPR, 2012.
[58] M. Yang, P. Zhu, L. V. Gool, L. Zhang, Face recognition based on regularized nearest points between image sets, in: IEEE Conference on Automatic Face and Gesture Recognition, 2013.
[59] E.G. Ortiz, A. Wright, M. Shah, Face recognition in movie trailers via mean sequence sparse representation-based classification, in: CVPR, 2013.
[60] G.B. Huang, M. Ramesh, T. Berg, E. Learned-Miller, Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments, Technical Report, University of Massachusetts, 2007.

Ran He received the BE and MS degrees in computer science from Dalian University of Technology, and the PhD degree in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Sciences, in 2001, 2004, and 2009, respectively. Since September 2010, he has been with the National Laboratory of Pattern Recognition, where he is currently an associate professor. He currently serves as an associate editor of Neurocomputing (Elsevier) and serves on the program committees of several conferences. His research interests include information theoretic learning, pattern recognition, and computer vision.

Yinghao Cai is a postdoctoral research associate working in University of Southern California. Before joining USC, she worked in the University of Oulu, Finland, since she received her PhD in Computer Science from National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences in 2009. Dr Cai's primary research interests include computer vision, image/video processing and machine learning.

Tieniu Tan received the BSc degree in electronic engineering from Xi'an Jiaotong University, China, in 1984, and the MSc and PhD degrees in electronic engineering from Imperial College London, United Kingdom, in 1986 and 1989, respectively. He is currently a professor in the Center for Research on Intelligent Perception and Computing (CRIPAC), National Laboratory of Pattern Recognition (NLPR) at the Institute of Automation, Chinese Academy of Sciences (CASIA). His current research interests include biometrics, image and video understanding, information hiding, and information forensics. He is a fellow of the IEEE and the IAPR (International Association of Pattern Recognition).

Larry S. Davis received the BA degree from Colgate University in 1970 and the MS and PhD degrees in computer science from the University of Maryland in 1974 and 1976, respectively. From 1977 to 1981, he was an assistant professor in the Department of Computer Science, University of Texas, Austin. He returned to the University of Maryland as an associate professor in 1981. From 1985 to 1994, he was the director of the University of Maryland Institute for Advanced Computer Studies. He is currently a professor at the Institute and in the Computer Science Department, as well as the chair of the Computer Science Department. He is known for his research in computer vision and high-performance computing. He has published more than 100 papers in journals and has supervised more than 20 PhD students. He is an associate editor of the International Journal of Computer Vision and an area editor for Computer Models for Image Processing: Image Understanding. He has served as the program or general chair for most of the field's major conferences and workshops, including the Fifth International Conference on Computer Vision, the 2004 Computer Vision and Pattern Recognition Conference, the 11th International Conference on Computer Vision. He became a fellow of the IEEE in 1997.