IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 12, 2017, accepted August 4, 2017, date of publication September 11, 2017, date of current version November 14, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2749331

Compressing Fisher Vector for Robust Face Recognition

HONGJUN WANG, JIANI HU, AND WEIHONG DENG
Pattern Recognition and Intelligent Systems Laboratory, Beijing University of Posts and Telecommunications, Beijing 100876, China

Corresponding author: Weihong Deng (whdeng@bupt.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Project 61471048, Project 61573068, and Project 61375031, in part by the Beijing Nova Program under Grant Z161100004916088, and in part by the Fundamental Research Funds for the Central Universities under Grant 2014ZD03-01.

ABSTRACT One major topic for robust face recognition could be the efficient encoding of facial descriptors. Among various encoders, Fisher vector (FV) is one of the probabilistic methods that yield promising results. However, its huge representation is fairly forbidding. In this paper, we present approaches to efficiently compress FV and retain its robustness. First, we put forward a new Compact FV (CFV) descriptor. The CFV is obtained by zeroing out small posteriors, calculating first-order statistics and reweighting its elements properly. Second, in light of Iterative Quantization (ITQ) scheme, we present a Generalized ITQ (GITQ) method to binarize our CFV. Finally, we apply our CFV and GITQ to encode convolutional activations of convolutional neural networks. We evaluate our methods on FERET, LFW, AR, and FRGC 2.0 datasets, and our experiments reveal the advantage of such a framework.

INDEX TERMS Fisher vector, face recognition, dimensional reduction, hashing, convolutional activations.

I. INTRODUCTION

The pursuit of fast automatic and precise recognition of face has motivated researchers in a range of fields, and related works have been applied to public security, human-computer interaction, etc. In two recent papers [1], [2], face recognition is applied to unconstrained settings. The task is rather challenging and is still an open problem due to high variability. Additionally, to identify a person of interest, we need to make use of various sources available (e.g. video surveillances, sketches), which could be time-consuming. There are two face recognition tasks: face identification (which is to identify an unknown person given a gallery set) and face verification (which is to decide whether two images are of the same person). In this work, we address both identification task and verification task.

The Bag-of-Features (BoF) model is one of the most popular and effective image classification framework during the last decade. A standard pipeline of the BoF model consists of: local descriptors (such as SIFT) extraction; codebook generation; local feature encoding and classification [3], [4].

Of all the above steps, codebook generation and feature encoding are the core components. Huang *et al.* [4] group the existing coding strategies into five major categories based on their motivations. They are: voting-based, Fisher vector-based, reconstruction-based (sparse coding, local coordinate coding, local-constraint linear coding), local tangent-based (local tangent coding, super vector coding) and saliency-based (salient coding, group salient coding). Our method is based on Fisher Vector (FV).

FV descriptor estimates probability density distribution and is considered more accurate than other methods. However, Sánchez *et al.* [5] stated that the Fisher representation of ILSVRC 2010 dataset (1.4 million images) with 512K dimensions per image requires almost 3TBs. Handling TBs of data makes experimentation difficult if not impractical. Thus, efficient representation is vital in the context of large-scale databases. To combat the problem of large-scale image search, binarization [6] and product quantization [5] are introduced as efficient and effective approaches to performing lossy compression of FVs.

In this paper, we devise approaches to compressing the storage of FV. Firstly, we put forward a Compact FV (CFV) descriptor that halves the dimension of FV. CFV is obtained by zeroing out small posterior probabilities and then computing the 1st order statistics. Elements in CFV are reweighted in a certain way that important geometrical information is preserved. Secondly, based on Iterative Quantization (ITQ), we present a new hashing method called General ITQ (GITQ) to generate short binary codes for similarity search. GITQ is more flexible than ITQ in that the projection matrix does

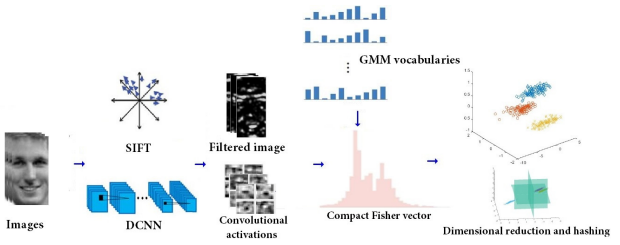

FIGURE 1. Pipeline of our approach.

not need to be square. Finally, we apply CFV and GITQ to convolutional activations of a Convolutional Neural Network (CNN). These activations can be seen as an alternative to local SIFT descriptors. The flowchart of the proposed method is shown in Figure 1.

The novelty of the proposed approach includes: (i) a novel compact CFV descriptor that achieves decent performance at a low dimension; (ii) combination of CFV with further dimension reduction and hashing tools, including a novel Generalized ITQ for efficient hashing; (iii) generalization of CFV to encode CNN convolutional activations. We could see that CFV enhances CNN recognition result.

This paper is built upon our preliminary work reported in [7] and [8]. The main differences are summarized as follows. (i) We introduce a parameter to the weighting of CFV elements to achieve better recognition result. (ii) We extend our previous works by encoding convolutional activations as a 'local' descriptor. This extension could be easily implemented for various pretrained networks. (iii) We re-evaluate our methods while assessing more recent state-of-the-art methods based on code provided by the authors to ensure a fair comparison. Tables and figures are redone for a better demonstration. Also, we experiment on AR and FRGC 2.0 Experiment 4 to prove the effectiveness of our method.

II. RELATED WORKS

A. FISHER VECTOR

Fisher vector (FV) [9] derives from Fisher Kernel (FK), which describes a feature by the gradient vector derived from a probability density function. Let θ denote the parameter of the probability distribution function p. FK characterizes each sample in $X = \{x_t, t = 1, ..., T\}$ with the gradient $\nabla_\theta \log p(x_t|\theta)$, which describes the direction in which parameters should be modified to best fit the data. Thus, the gradient vector provides an intuitive way to describe the geometric relationships between a data point x_t and a probability density function in feature space. FV is a specific version of FK that captures the average higher-order statistics between a local descriptor (such as SIFT) and a center of Gaussian Mixture Model (GMM). Specifically, $\theta = \{w_k, \mu_k, \Sigma_k, i = 1, ..., K\}$, which denote prior, mean vector and covariance matrix of the k-th Gaussian, respectively. Suppose all features are independent, the log-likelihood of the extracted features are $\sum_t \log p(x_t|\theta)$. The likelihood that x_t is generated by the GMM is $p(x_t|\theta) = \sum_{k=1}^{K} w_k p_k(x_t|\theta)$, with $\sum_{k=1}^{K} w_k = 1$ and $p_k(x_t|\theta) = \frac{\exp\{-(x-\mu_k)'\Sigma_k^{-1}(x-\mu_k)/2\}}{(2\pi)^{D/2}|\Sigma_k|^{1/2}}$.

Let $\gamma_t(k) = \frac{w_k p_k(x_t|\theta)}{\sum_{j=1}^{N} \log w_j p_j(x_t|\theta)}$ be the posterior probability of x_t belonging to the k-th Gaussian. The gradients of feature x_t with respect to GMM parameters can be calculated as:

$$\mathscr{G}^X_{w_k} = \frac{1}{\sqrt{w_k}} \sum_t (\gamma_t(k) - w_k) \quad (1)$$

$$\mathscr{G}^X_{\mu_k} = \frac{1}{\sqrt{w_k}} \sum_t \gamma_t(k) \frac{x_t - \mu_k}{\sigma_k} \quad (2)$$

$$\mathscr{G}^X_{\sigma_k} = \frac{1}{\sqrt{2w_k}} \sum_t \gamma_t(k) [\frac{(x_t - \mu_k)^2}{\sigma_k^2} - 1] \quad (3)$$

where $\mathscr{G}^X_{w_k}$ is a scalar of 0th order statistics, and $\mathscr{G}^X_{\mu_k}, \mathscr{G}^X_{\sigma_k}$ are vectors of D dimensions corresponding to 1st and 2nd order statistics respectively. $\mathscr{G}^X_{w_k}$ is often ignored. Thus the final feature is obtained by stacking the gradients in Equation 2 and 3. FV is therefore of dimension $2DK$.

An issue with FV is that the statistics of *all* samples with respect to a Gaussian center are summarized. To enrich the representation, BossaNova [10] is put forward by keeping a histogram of distances between the local descriptors found in the image and those in the codebook.

B. DIMENSION REDUCTION AND HASHING

Representations of images are usually in high-dimensional spaces and suffer from the curse of dimensionality [11] and hubness [12]. Therefore, it is often considered essential to apply proper dimensional reduction tool and hashing techniques.

Principal Component Analysis (PCA) is an unsupervised data compressing method that picks bases by looking for directions in which data varies most. The projected data is $X^{PCA}_{k\times N} = A_{k\times d} X_{d\times N}$, where A is the matrix containing k eigenvectors. Linear Discriminant Analysis (LDA), on the other hand, is a supervised method that seeks to maximize the ratio of *between class scatter* S_b and *within class scatter* S_w. This is equivalent to solving $S_w^{-1} S_b w = \lambda w$ [11]. Regularized Discriminant Analysis (RDA) [13] adds a small parameter ϵ times the identity matrix to prevent unstable matrix inversions. Sometimes, we are faced with Single Sample Per Person (SSPP) face recognition problems, where only *one* training sample for each person is available in the database. If this is the case, we could not use LDA or RDA for dimensional reduction, as intra-person variation cannot be obtained. In this situation, we could use Whitened PCA (WPCA). WPCA is the combination of PCA and the whitening transformation and its projected data is $X^{WPCA}_{k\times N} = \Lambda^{-1/2}_{k\times k} A_{k\times d} X_{d\times N}$, where A is the matrix containing k eigenvectors and $\Lambda = diag\{\lambda_1, \lambda_2, ..., \lambda_k\}$ with λ_i being the i-th leading eigenvalue.

Hashing transforms data to a sequence of bits. Many types of hashing methods exist. Local-sensitive hashing (LSH) [14] is a randomized hashing framework. It relies on hash functions which satisfy locality sensitive property.

Discriminant Binary Coding [15] trains every binary code in a way to ensure discrimination and learnability. Iterative Quantization (ITQ) [16] iteratively minimizes the quantization error of projecting examples from the original feature space to vertices of a binary hypercube. Let W be the PCA dimensionality reduction matrix, and $V = XW$ be the projected data. ITQ optimization starts with a random orthogonal matrix R, and the binary code matrix is updated as $B = sign(VR)$. Then B is fixed and R is updated by solving the classic orthogonal Procustes problem [17]. Previous steps are done iteratively to minimize the quantization loss $Q(B, R) = \|B - VR\|_F^2$. A more detailed description of the methods above and their interconnections can be found in our previous paper [8].

C. DEEP CONVOLUTIONAL ACTIVATIONS

Recently, deep CNNs have significantly improved the state-of-the-art in face recognition field. CNNs enhance traditional features by providing the trade-off between 'breadth' and 'depth'. There are already many deep networks for face recognition. Facebook's DeepFace [18], Google's FaceNet [19] and Baidu's deep network [20] are based on large industrial datasets. There are also some alternatives to the classic 'convolution+FC' framework, such as: [21]–[23]. Following [24], some papers (for example [20], [25]) utilize a set of CNNs to extract complementary facial features from multimodal data and then perform data fusion.

Though facial image transforms are highly non-linear, face images share a relatively similar structure and this is beneficial for transfer learning among databases. Fortunately, Parkhi *et al.* [26] have recently developed VGG Face: a deep network based on more than two million face images. This pretrained network could be utilized to capture features. The reasons why we choose VGG Face are as follows. Firstly, this network is reported to achieve comparable results to the state of the art with less data (than [18] and [19]) and a simpler network architecture (than [21] and [23]). Secondly, the success of utilizing such a classic 'convolution+FC' network architecture could be easily extended to other networks. The details and parameters of VGG Face would be discussed in Section III-C.

III. COMPRESSING FISHER VECTOR

In Section II-A we introduced FV formulation. Our method strives to make FV a compact descriptor while retaining its robustness. In this section, we first put forward CFV that halves the FV dimension. Then we introduce GITQ as an efficient hashing technique. Lastly, we extend our CFV to encode CNN convolutional activations.

A. COMPACT FISHER VECTOR (CFV)

The motivation of CFV is to preserve important geometric information and lower the storage of FV. CFV is built up as shown in Figure 2. Firstly, we prune the original FV by retaining the 1st order statistics. Then, we perform normalization within a block surrounding each Gaussian center. Finally, we devise a reweighting scheme to integrate information

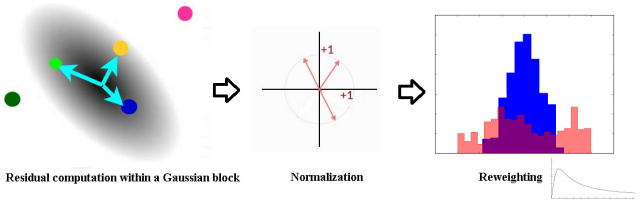

FIGURE 2. Pipeline of CFV.

regarding distances between a feature point and a Gaussian center. The motivation of the reweighting scheme and why it can encode geometric information is provided at the end of this subsection.

We first introduce the concept of the Gaussian block. FV calculates the posterior probability for each local descriptor x_t with respect to each Gaussian center. In practice, this posterior probability is usually small. Thus, for each Gaussian center, we zero out posteriors that gone below a certain threshold c. This would result in a virtual block surrounding the Gaussian center. We believe this approach could abate the negative impact of Gaussian centers that are far away from a local descriptor. We denote this virtual block as a *Gaussian block*.

We form the CFV by computing the sum of residuals in Equation 2 within a Gaussian block. These residuals are deviations of the local features from the Gaussian visual word. However, this total deviation is discounted by the sum of posteriors, rather than the square root of Gaussian prior in Equation 2. The rationale for this alteration is that: these residuals are aggregated within a Gaussian block defined by posterior probability, and the sum of posteriors would depict the selected range more precisely than the prior. The resulting KD-dimensional vector is shown in Equation 4 as:

$$\mathcal{G}_{\mu_k}^X = \frac{1}{\sum_{t,\gamma_t(k)>c} \gamma_t(k)} \sum_{t,\gamma_t(k)>c} \gamma_t(k) \frac{x_t - \mu_k}{\sigma_k} \tag{4}$$

Finally, specific weighting for each element $\mathcal{G}_{\mu_k}^X$ is introduced by taking into account the norm of the residual: $\mathcal{G}_{\mu_k}^X \leftarrow \frac{\|\mathcal{G}_{\mu_k}^X\|}{(\|\mathcal{G}_{\mu_k}^X\|^\alpha + 1)} \mathcal{G}_{\mu_k}^X$. The weighting function is shown in Figure 3a. To ensure the plot accord to this figure, α should be greater than 1. The choice for parameter c and α would be discussed in Section IV-F, and usually we set $c = 0.001$ and $\alpha = 2$. The rationale behind the function is to reduce the impact of the faulty classifier based on angle comparison (e.g. cosine measure). To illustrate, Figure 3b and 3c show two different distributions of local features with respect to a Gaussian center. Green diamond-shaped points refer to local features from one sample while yellow rounded points refer to those of another sample. The arrows signify the sums of residuals computed for the two samples. In both settings, the directions of arrows are distinct, and therefore the classifier would regard the samples as two different patterns. However, features of the two samples in Setting 1 lie close to each other and should be considered as one pattern. The function we proposed has the property of weakening the weights of features that lie

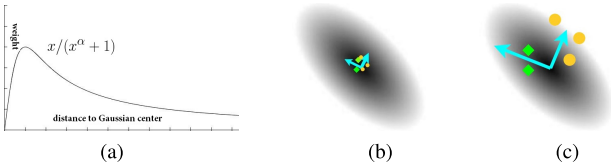

FIGURE 3. Specific weighting and its intuitive explanation. (a) Weighting function. (b) Setting 1. (c) Setting 2.

extremely close to the Gaussian center. The function also penalizes features that are far away in that they tend to be unrelated to the Gaussian center concerned. The method we proposed does not add any extra dimension, yet provides a way to integrate information regarding distances between a feature point and a Gaussian center.

B. GENERALIZED ITQ (GITQ)

We discuss how we further reduce storage of CFV with GITQ hashing algorithm. As we have seen in Section II-B, ITQ projects the data with PCA before hashing. Unfortunately, the length of the desired code are required to be smaller than the dimension of the projected data. We believe this constraint limits its use and may cause lack of stability, as we are more likely to project the data to higher dimensions (at least as high as our desired code length). However, due to noise and the structure of the face, some facial features are discriminative when they are projected to low dimensions (for instance 20 dimensions). Under that specific setting, ITQ code could not exceed 20 bits, which may not be enough for robust classification. Our GITQ could utilize more bits for classification, and in the experiment section, we could see that GITQ outperforms ITQ.

GITQ is a generalized form of ITQ where the projection matrix is not necessarily square. This idea is based on Sparse Projection [27], where the orthogonal constraint of R is relaxed to $R^T R = I$. We believe the Procustes problem is solvable when we use more bits than the original dimension of data.

The processes of GITQ are as follows. Minimization of $Q(B, R) = \|B_{n\times b} - V_{n\times d}R_{d\times b}\|_F^2$ where $b > d$ can be seen as equivalent to padding V with $b - d$ zero columns and then solve classic orthogonal Procrustes problem [28]. This transformation can be interpreted as a rotation of V in to higher dimensional space. Therefore, we compute the SVD of $d \times b$ matrix $V^T B$ as UDV^T, truncate V to $\tilde{V}$ by taking only its first d columns, and then let $R = \tilde{V}U^T$.

If we wish to perform LDA prior to GITQ hashing step, we have to make LDA projection matrix orthogonal to satisfy the constraint of the classic orthogonal Procustes problem. There are reasons for this. Firstly, as [29] pointed out, to generate pairwise uncorrelated bits, we need to satisfy a constraint that hashing hyperplanes should be orthogonal to each other. Secondly, matching performed at orthogonal Procrustes analysis stage is based on Euclidean (ℓ_2) distance. In an oblique coordinate system computations of distances and angles must be modified from Cartesian systems. Though variables of each dimension are uncorrelated after LDA step, discriminants extracted by LDA are not necessarily orthogonal. Therefore, in our experiments, we add an orthogonalization procedure simply by doing QR factorization for the projection matrix of LDA.

C. CFV ON DEEP CONVOLUTIONAL ACTIVATIONS

We now extend our CFV to encode convolutional features of a CNN. Deep convolutional activation features are extracted via VGG Face [30] network. The reason why we choose such a network is already presented in Section II-C. The network comprises 12 convolutional layers and each of them is succeeded by ReLU. Convolutional features commence at 64 channels and achieve 512 channels at the final convolution layer (29th layer). There are also 5 max-pooling layers lie between convolutional layers. Finally, there are 3 FC layers and a softmax function layer. The first two FC layers are of 4,096 dimensions and the last FC layer (FC8) has 2,622 dimensions. The input to the network is a face image of size 224×224 with the average face image subtracted.

Our CFV encoder is based on the activations 29th convolutional layer. For each image, this layer consists of 512 channels of 14×14 activations images, yet dense SIFT features are a collection of 128-dimensional vectors extracted at each pixel of the image. However, if we regard dense SIFT as 128 channels of filtered images (see Figure 1), then convolutional features and dense SIFT features are virtually interchangeable. Therefore, to assess CFV encoding scheme, we densely extract all the 14×14 activations, arranging them as a collection of 512-dimensional feature vector. Then, we append locality features $[x/w - 1/2; y/h - 1/2]$ to these 512-dimensional vectors extracted at location (x, y), where w and h represent the height and width of the response image of the final convolutional layer, respectively. The augmented features reflect their spatial locations.

IV. EXPERIMENTS

The performance of our CFV encoding algorithm has been evaluated on four facial databases: FERET, LFW, AR, and FRGC 2.0 Experiment 4. For FERET and LFW databases, we extract dense SIFT as local features and encode them with CFV. On the other hand, deep convolutional activations are employed for evaluations on AR and FRGC 2.0. There are two parameters (c and α) in our FV formulation. By default, we set $c = 0.0001$ and $\alpha = 2$. A comparison of parameters would be given in Section IV-F.

A. DATASETS

The FERET database [31] is an SSPP face database with 14051 facial images. The set contains a gallery set of 1196 individuals, and four test sets (fafb, fafc, dup1 and dup2) with variations on lighting, facial expression, pose, and age. LFW [32] is a set that contains 13233 training images of 5749 people and is considered as a standard benchmark for face verification. LFW is rather an unconstrained database, and it contains significant variations in pose,

illumination, expression, and occlusion. The evaluation procedure is to divide predefined image pairs into 10 folds and for each fold verify whether the image pair is of the same person. AR database [33] contains over 4,000 face images of 126 persons. In our experiment, we choose a subset consisting of 50 male and 50 female subjects. For each subject, we exclude images with subjects wearing sunglasses. The frontal face with the neutral expression of each person is used to form the gallery and all other images (19 images each person) are used for testing. FRGC 2.0 [34] Experiment 4 measures recognition performances of uncontrolled images. It produces 3 ROC curves corresponding to images within different time spans.

B. FERET

Dense SIFT features are extracted for all images with a stride of 2-pixels. PCA is used to reduce the dimensions of SIFTs from 128 to 64. Then locality features are appended to the descriptors. The resulting filtered images are divided horizontally into 3 blocks, and GMM models are trained independently for each block. Each GMM has 256 Gaussians. At the encoding stage, we zero out posteriors which are below 5×10^{-5}. Our final CFV is of $3 \times 66 \times 256 = 50688$ dimensions for each image. We apply WPCA to reduce the dimensions to 500 and classify them with Linear Ranking Analysis (LRA) [35].

TABLE 1. Recognition accuracies for various methods on FERET.

Method	Dimension	Accuracy
SV(s=2)[3]	101376	**1.0000** 0.9992 0.9252 0.9444
DCP[36]	18432	0.9816 **1.0000** 0.8530 0.8550
MD-DCP[36]	73728	0.9967 **1.0000** **0.9501** 0.9359
HOG+LBP[37]	74724	0.9720 0.9850 0.8530 0.8550
DFD[38]	50176	0.9940 **1.0000** 0.9180 0.9230
CBFD[39]	32000	0.9980 **1.0000** 0.9350 0.9320
FV	101376	**1.0000** 0.9992 0.9335 0.9444
CFV	50688	0.9991 **1.0000** **0.9501** **0.9529**

Some state-of-the-art methods are evaluated on our machine with codes provided by authors. Some details are: DCP [36] features are computed by calculating LBP-like patterns on two radii (3 and 5) and two directions (0 and $\pi/4$). MD-DCP is an improved version of DCP by computing first Gaussian derivative prior to DCP extraction. DCP, MD-DCP, DFD and CBFD aggregate features by histogramming with an 6×6 block division. Table 1 shows recognition accuracies of various methods, and it shows that our CFV achieves decent performance with a low feature dimension. Note that the dimension refers to that of the original feature.

Hashing algorithms for CFV on FERET are also tested. Their performances are shown in Figure 4 with groups of stacked bar plots under a different number of bits. In our experiment, the length of output varies from 100 to 600 bits. For each hashed length, performances of our GITQ are compared with ITQ [40], SP [27] and LSH [41]. Each stacked bar represents one method, and from left to right they signify GITQ, ITQ, SP and LSH respectively. SP is evaluated with a sparsity of 0.9. Finally, each bar represents stacked accuracies of a specific method on four test sets. The figure demonstrates the effectiveness of GITQ over other methods.

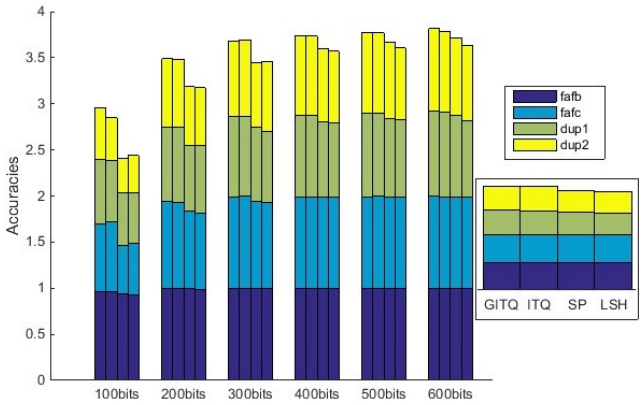

FIGURE 4. Performances of various hashing algorithms on FERET.

C. LFW

We evaluate our algorithms under unrestricted LFW protocol without outside training data. In our experiments, images are firstly cropped to 150×80 pixels. Multiscale Dense SIFT features are extracted with a stride of 2-pixels and the SIFT sliding windows are of 16×16 and 24×24 pixels, respectively. Each SIFT filtered image is horizontally cropped to 3 blocks. Then, in light of [42], we evaluate the power of the SIFT extracted and retain only 60% of the most prominent features extracted. After that, we perform PCA and append locality features, which are similar to our evaluation on FERET dataset. We train GMM models independently for each block and each scale, and each GMM has 128 Gaussians. At the encoding stage, we zero out posteriors which are below 5×10^{-5}. Our final CFV is of 50,688 dimensions for each image. We reduce dimensions first with PCA to 400 and then with LDA to 40 dimensions. Similarities are calculated using cosine measure. Verification accuracies of our CFV along with other local feature learning methods are shown in Table 2. We also compare accuracies of various hashing algorithms by which CFVs are projected to 100 bits. We can see that our CFV performs on par with High-Dimensional LBP (HD-LBP) [43], yet the dimension of CFV is only less than a half of HD-LBP. Moreover, CFV outperforms other features, and hashing with GITQ yields decent verification accuracy.

D. AR

For each image, we firstly extract convolutional activations from the final layer of VGG Face network. Next, locality features are appended to the 512-dimensional response vectors. Then the features are PCAed to 64 dimensions and they are encoded with CFV. To achieve better accuracy, these features are all reduced to 100 dimensions via WPCA.

TABLE 2. Comparison of verification accuracies on LFW.

Method	Dimension	Accuracy
LHS[44]	20480	0.734
LQP[45]	72000	0.862
DFD[38]	50176	0.840
CBFD[39]	32000	0.909
MD-DCP[36]	73728	0.889
HD-LBP[43]	127440	**0.932**
FV	101376	0.927
CFV	50688	**0.932**
CFV+ITQ[40]	100bits	0.881
CFV+SP[27]	100bits	0.879
CFV+GITQ	100bits	**0.899**

TABLE 3. Recognition accuracies for various methods on AR.

Method	Dimension	Accuracy
DCP[36]+LRA	32768	0.9489
MD-DCP[36]+EUCL	131072	0.9221
MD-DCP[36]+COS	131072	0.9663
MD-DCP[36]+LRA	131072	0.9805
DFD[38]+LRA	50176	0.9074
CBFD[39]+EUCL	32000	0.9658
CBFD[39]+COS	32000	0.9658
CBFD[39]+LRA	32000	0.9789
FC8[30]	2622	0.9605
FV (K=32) + EUCL	4096	0.9705
FV (K=32) + COS	4096	0.9705
FV (K=32) + LRA	4096	0.9705
CFV (K=32)+EUCL	2048	0.9700
CFV (K=32)+COS	2048	0.9732
CFV (K=32)+LRA	2048	0.9732
CFV (K=64)+EUCL	4096	**0.9916**
CFV (K=64)+COS	4096	**0.9916**
CFV (K=64)+LRA	4096	**0.9916**

Comparisons of accuracies of many methods combined with different classifiers are shown in Table 3. State-of-the-art methods (DCP, MD-DCP, DFD and CBFD) in Table 3 evaluated on our machine by histogramming with an 8×8 block division. FC8 feature is formed by extracting the final FC layer from VGG Face, which is a common practice for image classification. In the table, K means the *number of Gaussian centers*; EUCL means *nearest neighbor classifier based on Euclidean distance*; COS means *nearest neighbor classifier based on cosine measure* and LRA means *linear ranking analysis classifier*. Compared to state-of-the-art methods, our method achieves the best result with a significantly lower dimension. Under various parameters and classifiers, our CFVs perform on par with FVs with dimensions halved. A major accuracy boost (more than 2%) is obtained when CFVs are the same dimensions as FVs, regardless of the classifier.

Performances of various hashing methods on FV and CFV are shown in Figure 5. The results align with our argument: CFV combined with GITQ is consistently preferable compared with FV and other hashing methods. Some methods, like LSH, are not listed due to their unfavorable performances.

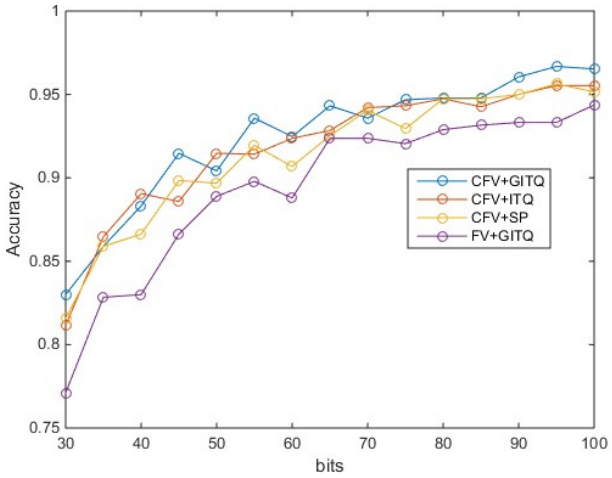

FIGURE 5. Comparison of features and hashing methods on AR.

E. FRGC 2.0

Similar to that of AR, we feed in multi-scale images to VGG Face and extract convolutional activations of 29th layer. Due to limitations of our machine, we cluster GMM (of 128 Gaussians) with only a maximum of 150 convolutional activations (of dimension 64) per image. We resort to PCA and RDA for dimensional reduction of the features, and the regularization parameter is chosen to be 0.001. Verification Rates (VRs) at 0.1% False Acceptation Rate (FAR) for various PCA and RDA choices are shown in Figure 6. In the figure, 'CFV w/o L' and 'FV w/o L' denote that we do not append locality features to convolutional responses. 'FC8' and 'FC7' are features extracted from the final and penultimate FC layers, respectively.

We observe that our CFV outperforms FV on all three test sets with a much lower dimension. Even when locality information is not present, 'CFV w/o L' only suffers from a minor loss and marginally outperforms both 'FV' and 'FV w/o L'. Given the low dimension of CFV compared to FV, we believe our descriptor is robust and functional in various settings. The figure also shows that hashing CFV with GITQ is reasonable compared to the unstable ITQ. FC7 and FC8 do not yield robust performance, which aligns with experiments on AR.

Comparisons with state-of-the-art methods are provided in Table 4. Despite the subsampling of convolutional response, our CFV achieves a satisfactory result: at PCA dimension 1400 and RDA dimension 200 VRs achieve 0.9269, 0.9305 and 0.9335 for ROC I, II and III respectively.

F. CHOICE OF PARAMETERS

There are two parameters in our formulation of CFV: posterior probability threshold c and weighting parameter α. As these two parameters are independent, we fix one parameter to its default value and change the other. We compare the choices of c on FERET (using dense SIFT) and FRGC 2.0 (using deep convolutional features), then we compare choices of α on AR (using deep convolutional features) and LFW

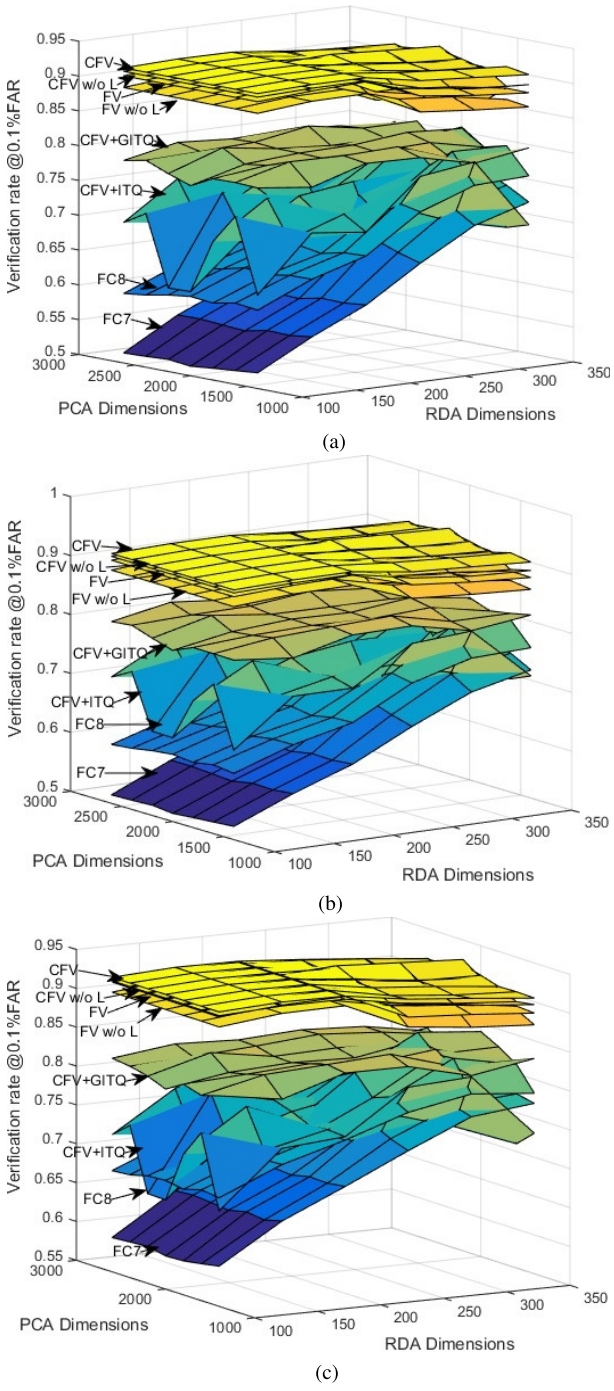

FIGURE 6. Verification rates on FRGC. (a) ROC I. (b) ROC II. (c) ROC III.

TABLE 4. Comparison of VRs at 0.1% FAR on FRGC 2.0.

Method	ROC I	ROC II	ROC III
MLPQH+KKDR[46]	0.8926	0.8980	0.9036
MLPQH+LDA[47]	0.8105	0.8244	0.8380
Fourier+LDA[48]	0.7570	0.7506	0.7433
MD-DCP[36]	**0.9322**	0.9121	0.8893
FV	0.9218	0.9255	0.9280
CFV	0.9269	**0.9305**	**0.9335**

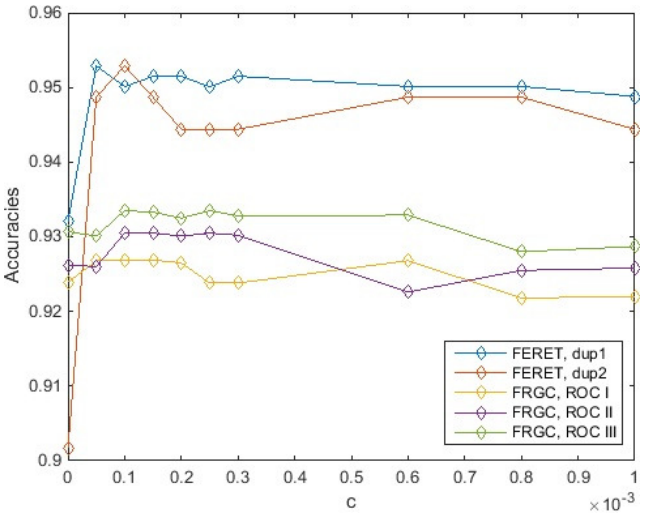

FIGURE 7. Recognition accuracies of CFV on FERET and FRGC under various c.

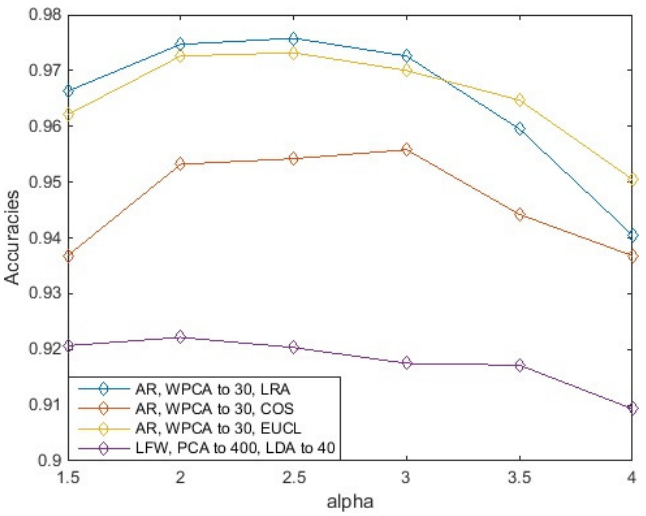

FIGURE 8. Recognition accuracies of CFV on AR and LFW under various α.

(using dense SIFT). We would demonstrate that performances of our CFV is quite stable even though the parameters vary.

First we increase c from 0 to 0.001, where $c = 0$ indicates that all posteriors are taken into account. Other settings are exactly the same as the CFV descriptor in Section IV-B and Section IV-E. For FERET our CFV is reduced to 500 dimensions via WPCA, and for FRGC 2.0 our CFV is reduced to 1400 via PCA and then 200 via RDA. The result is shown in Figure 7. We can see that for both datasets, performance boosts are observed when c ranges from 0.00005 to 0.0002. Therefore we choose $c = 0.0001$ by default.

We compare performances of CFV on AR and LFW with α ranging from 1.5 to 4 (as we have mentioned in Section III-A, α should be greater than 1). The result is shown in Figure 8. We extract AR and LFW with $K = 64$ Gaussian centers. For AR our CFV is reduced to 30 dimensions via WPCA, and for LFW our CFV is reduced to 400 via PCA and then 40 via LDA. All other settings are the same as in Section IV-D and

Section IV-C, respectively. We report accuracies on AR with three classifiers: EUCL (nearest neighbor classifier based on Euclidean distance), COS (nearest neighbor classifier based on cosine measure) and LRA (linear ranking analysis). Verification experiments on LFW are done according to cosine similarity. Though these tasks varies, recognition accuracies are quite stable when α is between 2 and 3, and $\alpha = 2$ is overall a reasonable choice.

V. CONCLUSION

We have addressed the issue of compressing FV and retaining its robustness at the same time. We have proposed a compact version of FV called CFV that preserves the most discriminative information. Moreover, distances between a local descriptor and a codeword are integrated into CFV formulation. Further, to facilitate its application in extremely large-scale image databases, we provide a flexible hashing method called GITQ. The combination of CFV and various dimensional reduction methods and/or GITQ proves to be beneficial for face identification and verification tasks. Finally, CFV is extended to encode deep convolutional activations. The proposed approach shows promising results for face recognition under both controlled (FERET, AR) and uncontrolled (LFW, FRGC 2.0) settings. We believe our CFV could be a successor to FV given its low dimensionality and robustness in face description, and our GITQ could further alleviate high-dimensional issues for large data processing.

ACKNOWLEDGMENTS

The authors were with the Pattern Recognition and Intelligent Systems Laboratory, Beijing University of Posts and Telecommunications, Beijing 100876, China.

REFERENCES

[1] G. Hua *et al.*, "Introduction to the special section on real-world face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 10, pp. 1921–1924, Oct. 2011.
[2] L. Best-Rowden, H. Han, C. Otto, B. F. Klare, and A. K. Jain, "Unconstrained face recognition: Identifying a person of interest from a media collection," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 12, pp. 2144–2157, Dec. 2014.
[3] K. Chatfield, V. S. Lempitsky, A. Vedaldi, and A. Zisserman, "The devil is in the details: An evaluation of recent feature encoding methods," in *Proc. Brit. Mach. Vis. Conf.*, 2011, p. 8.
[4] Y. Huang, Z. Wu, L. Wang, and T. Tan, "Feature coding in image classification: A comprehensive study," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 3, pp. 493–506, Mar. 2014.
[5] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek, "Image classification with the Fisher vector: Theory and practice," *Int. J. Comput. Vis.*, vol. 105, no. 3, pp. 222–245, 2013.
[6] F. Perronnin, Y. Liu, J. Sánchez, and H. Poirier, "Large-scale image retrieval with compressed Fisher vectors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 3384–3391.
[7] H. Wang and W. Deng, "Face recognition via compact Fisher vector," in *Proc. Chin. Conf. Biometric Recognit.*, 2015, pp. 68–77.
[8] H. Wang, J. Hu, and W. Deng, "Binary matching for high-dimensional image descriptors," in *Proc. 3rd IAPR Asian Conf. Pattern Recognit. (ACPR)*, Nov. 2015, pp. 401–405.
[9] F. Perronnin and C. Dance, "Fisher kernels on visual vocabularies for image categorization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2007, pp. 1–8.
[10] S. Avila, N. Thome, M. Cord, E. Valle, and A. de A. Araújo, "Pooling in image representation: The visual codeword point of view," *Comput. Vis. Image Understand.*, vol. 117, no. 5, pp. 453–465, 2013.
[11] C. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.
[12] M. Radovanović, A. Nanopoulos, and M. Ivanović, "Hubs in space: Popular nearest neighbors in high-dimensional data," *J. Mach. Learn. Res.*, vol. 11, pp. 2487–2531, Sep. 2010.
[13] J. H. Friedman, "Regularized discriminant analysis," *J. Amer. Stat. Assoc.*, vol. 84, no. 405, pp. 165–175, 1989.
[14] A. Andoni and P. Indyk, "Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions," in *Proc. 47th Annu. IEEE Symp. Found. Comput. Sci. (FOCS)*, Oct. 2006, pp. 459–468.
[15] M. Rastegari, A. Farhadi, and D. Forsyth, "Attribute discovery via predictable discriminative binary codes," in *Computer Vision ECCV*. New York, NY, USA: Springer, 2012, pp. 876–889.
[16] Y. Gong and S. Lazebnik, "Iterative quantization: A procrustean approach to learning binary codes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 817–824.
[17] P. H. Schönemann, "A generalized solution of the orthogonal Procrustes problem," *Psychometrika*, vol. 31, no. 1, pp. 1–10, 1966.
[18] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the gap to human-level performance in face verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 1701–1708.
[19] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 815–823.
[20] J. Liu, Y. Deng, T. Bai, Z. Wei, and C. Huang. (Jun. 2015). "Targeting ultimate accuracy: Face recognition via deep embedding." [Online]. Available: https://arxiv.org/abs/1506.07310
[21] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," in *Proc. 27th Int. Conf. Adv. Neural Inf. Process. Syst.*, 2014, pp. 1988–1996.
[22] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation from predicting 10,000 classes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1891–1898.
[23] Y. Sun, X. Wang, and X. Tang, "Deeply learned face representations are sparse, selective, and robust," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 2892–2900.
[24] C. Ding and D. Tao, "Robust face recognition via multimodal deep face representation," *IEEE Trans. Multimedia*, vol. 17, no. 11, pp. 2049–2058, Nov. 2015.
[25] C. Ding and D. Tao, "Trunk-branch ensemble convolutional neural networks for video-based face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 99, no. 5, pp. 225–236, May 2017.
[26] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *Proc. Brit. Mach. Vis. Conf.*, 2015, vol. 1. no. 3, p. 6.
[27] Y. Xia, K. He, P. Kohli, and J. Sun, "Sparse projections for high-dimensional binary codes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3332–3339.
[28] J. C. Gower, *Procrustes Problems*. New York, NY, USA: Oxford Univ. Press, 2004.
[29] J. Wang, S. Kumar, and S.-F. Chang, "Semi-supervised hashing for large-scale search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 12, pp. 2393–2406, Dec. 2012.
[30] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *Proc. Brit. Mach. Vis. Conf.*, 2015, p. 6.
[31] P. J. Phillips, H. Wechsler, J. Huang, and P. J. Rauss, "The FERET database and evaluation procedure for face-recognition algorithms," *Image Vis. Comput.*, vol. 16, no. 5, pp. 295–306, 1998.
[32] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database forstudying face recognition in unconstrained environments," in *Proc. Workshop Faces Real-Life Imag., Detection, Alignment, Recognit.*, 2008, pp. 1–15.
[33] A. M. Martinez, "The AR face database," CVC Birmingham, Tech. Rep. 24, 1998, vol. 24.
[34] P. J. Phillips *et al.*, "Overview of the face recognition grand challenge," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1. Jun. 2005, pp. 947–954.
[35] W. Deng, J. Hu, and J. Guo, "Linear ranking analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 3638–3645.
[36] C. Ding, J. Choi, D. Tao, and L. S. Davis, "Multi-directional multi-level dual-cross patterns for robust face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 3, pp. 518–531, Mar. 2016.

[37] W. R. Schwartz, H. Guo, J. Choi, and L. S. Davis, "Face identification using large feature sets," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 2245–2255, Apr. 2012.
[38] Z. Lei, M. Pietikäinen, and S. Z. Li, "Learning discriminant face descriptor," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 2, pp. 289–302, Feb. 2014.
[39] J. Lu, V. E. Liong, X. Zhou, and J. Zhou, "Learning compact binary face descriptor for face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 10, pp. 2041–2056, Oct. 2015.
[40] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin, "Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 12, pp. 2916–2929, Dec. 2013.
[41] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," in *Proc. 20th Annu. Symp. Comput. Geometry*, 2004, pp. 253–262.
[42] I. Everts, J. C. van Gemert, T. Mensink, and T. Gevers, "Robustifying descriptor instability using Fisher vectors," *IEEE Trans. Image Process.*, vol. 23, no. 12, pp. 5698–5706, Dec. 2014.
[43] D. Chen, X. Cao, F. Wen, and J. Sun, "Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 3025–3032.
[44] G. Sharma, S. ul Hussain, and F. Jurie, "Local higher-order statistics (LHS) for texture categorization and facial analysis," in *Computer Vision ECCV*. New York, NY, USA: Springer, 2012, pp. 1–12.
[45] S. U. Hussain, T. Napoléon, and F. Jurie, "Face recognition using local quantized patterns," in *Proc. Brit. Machive Vis. Conf.*, 2012, p. 11.
[46] C. H. Chan, M. A. Tahir, J. Kittler, and M. Pietikäinen, "Multiscale local phase quantization for robust component-based face recognition using kernel fusion of multiple descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 5, pp. 1164–1177, May 2013.
[47] C.-H. Chan, J. Kittler, and M. A. Tahir, "Kernel fusion of multiple histogram descriptors for robust face recognition," in *Proc. Joint IAPR Int. Workshops Stat. Techn. Pattern Recognit. (SPR) Struct. Syntactic Pattern Recognit. (SSPR)*, 2010, pp. 718–727.
[48] W. Hwang, G. Park, J. Lee, and S.-C. Kee, "Multiple face model of hybrid Fourier feature for large face image set," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2. Jun. 2006, pp. 1574–1581.

HONGJUN WANG received the master's degree from the China University of Mining and Technology. He is currently pursuing the Ph.D. degree with the School of Information and Communication Engineering, Beijing University of Post and Telecommunications. His research interests include machine learning, computer vision, and information extraction.

JIANI HU received the Ph.D. degree from the Beijing University of Posts and Telecommunications, China. He is currently an Associate Professor with the Pattern Recognition and Intelligent Systems Laboratory, Beijing University of Posts and Telecommunications. Her current research interests include computer vision, pattern recognition, and machine learning.

WEIHONG DENG received the Ph.D. degree from the Beijing University of Posts and Telecommunications, China. He is currently an Associate Professor with the Pattern Recognition and Intelligent Systems Laboratory, Beijing University of Posts and Telecommunications. His current research interests include computer vision, pattern recognition, and machine learning.