FISHER VECTOR ENCODED DEEP CONVOLUTIONAL FEATURES FOR UNCONSTRAINED FACE VERIFICATION

Jun-Cheng Chen[1*], *Jingxiao Zheng*[1*], *Vishal M. Patel*[2], *and Rama Chellappa*[1]

1. University of Maryland, College Park
2. Rutgers, The State University of New Jersey

pullpull@cs.umd.edu, jxzheng@umiacs.umd.edu, vishal.m.patel@rutgers.edu, rama@umiacs.umd.edu

ABSTRACT

We present a method to combine the Fisher vector representation and the Deep Convolutional Neural Network (DCNN) features to generate a rerpesentation, called the Fisher vector encoded DCNN (FV-DCNN) features, for unconstrained face verification. One of the key features of our method is that spatial and appearance information are simultaneously processed when learning the Gaussian mixture model to encode the DCNN features. Evaluations on two challenging verification datasets show that the proposed FV-DCNN method is able to capture the salient local features and also performs well when compared to many state-of-the-art face verification methods.

1. INTRODUCTION

Learning invariant and discriminative features from images and videos is one of the central goals of research in many computer vision tasks such as object recognition and face recognition. Many approaches have been proposed in the literature that extract over-complete and high-dimensional features from images to handle large data variations and noise. For instance, the high-dimensional multi-scale Local Binary Pattern (LBP) [1] representation extracted from local patches around facial landmarks is reasonably effective for face recognition. Face representation based on Fisher vector (FV) has also shown to be effective for face recognition problems [2], [3], [4].

However, in recent years, deep convolutional neural networks (DCNN) have demonstrated impressive performances on several computer vision problems such as object recognition [5][6], object detection [7], and face verification [8]. It has been shown that a DCNN model can not only characterize large data variations but also learn a compact and discriminative feature representation when the size of the training data is sufficiently large.

Motivated by the success of FV and DCNN models for various computer vision problems, we propose to combine them for face verification. We adopt a network architecture similar to the one proposed in [9] which has demonstrated impressive performance for face recognition. The DCNN model is trained using the CASIA-WebFace dataset which consists of 10,575 subjects. This model builds a very deep architecture for convolutional neural network by stacking small filters (*i.e.* 3×3) together as VGGNet [10] and is trained with 10,575 subjects as the DeepID Net [11]. Finally, average pooling is applied right after the last convolutional layer to generate a global descriptor and reduce the number of parameters for the fully connected layers. When average pooling is used, the spatial information is often lost. Hence, to exploit the local appearance information, we propose to apply FV to encode the local convolutional features from each image into a high-dimensional vector instead of applying the average pooling operation.

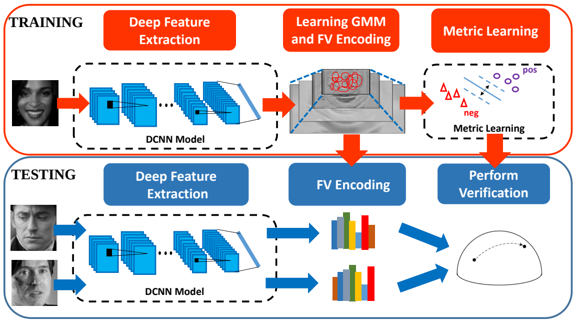


Fig. 1. An overview of the proposed FV-DCNN representation for unconstrained face verification.

Our work is somewhat similar to the previous work proposed by Cimpoi *et al.* [12] for texture recognition. They also propose to combine the FV and DCNN features. However, their approach does not take into account any spatial information. Compared to textures, human faces have well-defined structure (*e.g.* faces are approximately symmetric.). Thus, the main distinction between the two works is that each local DCNN feature is augmented with spatial information in the image when we apply FV to encode DCNN features from an image. An example is illustrated in Section 4 to show how the learned GMM looks like with and without incorporating the spatial information. Furthermore, as was shown in [12], since lower layers contain less discriminative information, we focus on the conv52 features to incorporate the spatial information with FV.

2. RELATED WORK

In this section, we briefly review several recent works on face verification.

Robust feature learning is a key component in a face verification system. It can be roughly classified into two categories: hand-crafted features and features learned directly from data. In the first category, Ahonen *et al.* [13] showed that the Local Binary Pattern (LBP) is effective for face recognition. Gabor wavelets [14] have also been widely used to encode multi-scale and multi-orientation information for face images. Chen *et al.* [1] demonstrated good results for face verification using the high-dimensional multi-scale LBP features extracted from patches around facial landmarks. In the

The first two authors equally contribute to this work. This research is based upon work supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA R&D Contract No. 2014-14071600012. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

second category, Simonyan *et al.* [2] and Parkhi *et al.* [3] used the FV encoding to generate over-complete and high-dimensional feature representation for still and video-based face recognition. Some of the other feature encoding methods include Bag-of-Visual-Words (BoVW) model [15], VLAD [16] and Super Vector Coding [17].

The high-dimensionality of feature vectors makes these methods difficult to train and scale to large datasets. However, recent advances in deep learning methods have shown that compact and discriminative representation can be learned using DCNN from very large datasets. Taigman *et al.* [18] learned a DCNN model on the frontalized faces generated with a general 3D shape model from a large-scale face dataset. Sun *et al.* [19][11] achieved results surpassing human performance for face verification on the LFW dataset using an ensemble of 25 simple DCNNs with fewer layers trained on weakly aligned face images from a much smaller dataset than the former. Schroff *et al.* [8] adapted the state-of-the-art deep architecture in object recognition to face recognition and trained on a large-scale unaligned private face dataset. Parkhi *et al.* [20] trained a very deep convolutional network based on VGGNet for face verification and demonstrated impressive results. These works essentially demonstrate the effectiveness of the DCNN model for feature learning and detection/recognition/verification problems.

3. PROPOSED METHOD

An overview of our FV-DCNN method for face verification is shown in Figure 1. Each training image is first passed through a pre-trained DCNN model to extract the convolutional features. Then, we learn the Gaussian mixture model over them and perform FV encoding over these local convolutional features which have already encoded the rich face feature information. Finally, we learn the metric. In the testing phase, we extract the DCNN features and use the learned GMM to perform FV feature encoding. We then apply the learned metric to compute the similarity scores. In what follows, we describe the details of each of these components.

Face Preprocessing: Each face image is detected and aligned using the open-source library dlib [21] [22]. Each face is aligned into the canonical coordinate using the similarity transform and seven landmark points (*i.e.* two left eye corners, two right eye corners, nose tip, and two mouth corners). After alignment, the face image resolution is 100 × 100 pixels.

Fisher Vector Encoding: The FV is one of bag-of-visual-word approaches which encodes a large set of local features into a high-dimensional vector according to the parametric generative model fitted for the features. The FV representation is computed by encoding the local features with the derivatives of the log-likelihood of the learned model with respect to the model parameters. Similar to [23], we use a GMM in our work. The first-and second-order statistics of the features with respect to each component for the FV representation are computed as follows:

$$\mathbf{\Phi}_{ik}^{(1)} = \frac{1}{N\sqrt{w_k}} \sum_{p=1}^{N} \alpha_k(\mathbf{v}_p) \left(\frac{\mathbf{v}_{ip} - \boldsymbol{\mu}_{ik}}{\boldsymbol{\sigma}_{ik}} \right), \tag{1}$$

$$\mathbf{\Phi}_{ik}^{(2)} = \frac{1}{N\sqrt{2w_k}} \sum_{p=1}^{N} \alpha_k(\mathbf{v}_p) \left(\frac{(\mathbf{v}_{ip} - \boldsymbol{\mu}_{ik})^2}{\boldsymbol{\sigma}_{ik}^2} - 1 \right), \tag{2}$$

$$\alpha_k(\mathbf{v}_p) = \frac{w_k \exp[-\frac{1}{2}(\mathbf{v}_p - \boldsymbol{\mu}_k)^T \mathbf{\Sigma}_k^{-1} (\mathbf{v}_p - \boldsymbol{\mu}_k)]}{\sum_i^K w_i \exp[-\frac{1}{2}(\mathbf{v}_p - \boldsymbol{\mu}_i)^T \mathbf{\Sigma}_i^{-1} (\mathbf{v}_p - \boldsymbol{\mu}_i)]}, \tag{3}$$

where w_k, $\boldsymbol{\mu}_k$, $\mathbf{\Sigma}_k = diag(\boldsymbol{\sigma}_{1k}, ..., \boldsymbol{\sigma}_{dk})$ are the weights, means, and diagonal covariances of the kth mixture component of the GMM. Here, $\mathbf{v}_p \in \mathbb{R}^{d \times 1}$ is the pth feature vector and N is the number of feature vectors. The parameters are learned from the training data using the EM algorithm. $\alpha_k(\mathbf{v}_p)$ is the posterior of $\mathbf{v}_p$ belonging to the kth mixture component. The FV representation, $\mathbf{\Phi}(\mathbf{I})$, of an image $\mathbf{I}$ is obtained by concatenating all the $\mathbf{\Phi}_k^{(1)}$s and $\mathbf{\Phi}_k^{(2)}$s into a high-dimensional vector $\mathbf{\Phi}(\mathbf{I}) = [(\mathbf{\Phi}_1^{(1)})^T, (\mathbf{\Phi}_1^{(2)})^T, ..., (\mathbf{\Phi}_K^{(1)})^T, (\mathbf{\Phi}_K^{(2)})^T]^T$, whose dimensionality is $D = 2Kd$ where K is the number of mixture components, and d is the dimensionality of the local feature vector where we use $d = 322$ in this work.

Metric Learning: For the face verification task, the standard protocol defines the same and different pairs for comparison which can be also used to train discriminative similarity function to improve the verification performance. In this work, our focus is on robust feature representation, and thus we choose to use joint Bayesian approaches which are widely used for face verification [24]. The joint Bayesian approach models both intra-class, $P(\mathbf{x}_i, \mathbf{x}_j | H_I) \sim N(0, \mathbf{\Sigma}_I)$, and inter-class, $P(\mathbf{x}_i, \mathbf{x}_j | H_E) \sim N(0, \mathbf{\Sigma}_E)$, joint feature distribution of ith and jth images as Gaussians. In addition, each face feature vector is modeled as $\mathbf{x} = \boldsymbol{\mu} + \boldsymbol{\epsilon}$, where $\boldsymbol{\mu}$ stands for the identity and $\boldsymbol{\epsilon}$ for pose, illumination, and other variations. Both $\boldsymbol{\mu}$ and $\boldsymbol{\epsilon}$ are also assumed to be independent zero-mean Gaussian distributions, $N(0, \mathbf{S}_\mu)$ and $N(0, \mathbf{S}_\epsilon)$, respectively.

The closed-form log likelihood ratio of intra- and inter-classes, $r(\mathbf{x}_i, \mathbf{x}_j)$, can be written as

$$r(\mathbf{x}_i, \mathbf{x}_j) = \log \frac{P(\mathbf{x}_i, \mathbf{x}_j | H_I)}{P(\mathbf{x}_i, \mathbf{x}_j | H_E)} = \mathbf{x}_i^T \mathbf{M} \mathbf{x}_i + \mathbf{x}_j^T \mathbf{M} \mathbf{x}_j - 2\mathbf{x}_i^T \mathbf{R} \mathbf{x}_j, \tag{4}$$

where $\mathbf{M}$ and $\mathbf{R}$ are both negative semi-definite matrices [24]. If we let $\mathbf{B} = \mathbf{R} - \mathbf{M}$, then (4) can be rewritten as $(\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{M} (\mathbf{x}_i - \mathbf{x}_j) - 2\mathbf{x}_i^T \mathbf{B} \mathbf{x}_j$. With this, one can directly optimize the distance in a large-margin framework as follows:

$$\underset{\mathbf{M},\mathbf{B},b}{\operatorname{argmin}} \sum_{i,j} max[1 - y_{ij}(b - (\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{M} (\mathbf{x}_i - \mathbf{x}_j) + 2\mathbf{x}_i^T \mathbf{B} \mathbf{x}_j), 0], \tag{5}$$

where b is the bias, $\mathbf{M} = \mathbf{W}^T \mathbf{W}$, and $\mathbf{B} = \mathbf{V}^T \mathbf{V}$ (*i.e.*, $\mathbf{W}$, $\mathbf{V} \in \mathbb{R}^{\hat{d} \times D}$ and $\hat{d} \ll D$. We use $\hat{d} = 100$ in this paper). The model parameters can be updated using the stochastic gradient descent algorithm. More details can be found in [4].

Deep Face Feature Representation: The DCNN model used in this paper is similar to the one proposed in [9] which includes 10 convolutional layers, 5 pooling layers and 1 fully connected layer. The model is trained using the CASIA-WebFace dataset which contains 10,575 subjects. The differences between the proposed method and the one proposed in [9] are that we train the model only with softmax identification loss and without modeling the pair-wise verification cost. Each activation function, the rectified linear unit (ReLU), is replaced with the parametric ReLU (PReLU) [25] which allows negative responses and usually improves the network performance. The dimensionality of the input layer is 100 × 100 × 1 for gray-scale images. Most recent works in face verification like [26] and [27] use similar architecture as [9] and take the pool5 features followed by L_2-normalization to perform recognition. The training details of DCNN model can be found at [28].

Since pool5 takes the average pooling to aggregate the DCNN features of the last convolutional layer, conv52, its features contain the global information of the appearance. However, average pooling also causes the loss of spatial and local appearance information. In the next Section, we combine both the FV and DCNN features to utilize spatial information for face verification. Furthermore, we only focus on the conv52 features to incorporate the spatial information with FV as the lower layer contains less discriminative information [12].

FV Encoded DCNN Features (FV-DCNN): For human faces, the appearances around facial local regions are different. Thus, instead of using the global pool5 features for the verification task as used in [9] [27] (*i.e.* the pool5 feature $\in \mathbb{R}^{320\times 1}$ is the average of 7×7 features of the conv52 layer $\in \mathbb{R}^{7\times 7\times 320}$), the spatial information should be taken into consideration as well. In order to incorporate the spatial information into the model, we augment two additional dimensions (normalized x and y coordinates with respect to image width and height, $[\frac{x}{w} - \frac{1}{2}, \frac{y}{h} - \frac{1}{2}]^T$, say spatial features) to the original conv52 features and apply FV encoding on these augmented features. Then the learnt GMM will not only cluster the features with similar appearance, but also consider their spatial relationships. Hence, the FV will mostly be encoded by the features in the neighborhoods of the corresponding Gaussians.

To balance the strength of appearance and spatial features, we take the square root and perform L_2 normalization on appearance features before augmenting spatial features. Moreover, we introduce an encoding scheme called "spatial encoding". Instead of using the original posterior (3), by spatial encoding, we enforce the feature to be encoded by its neighborhood, which is defined as

$$\tilde{\alpha}_k(\mathbf{v}_p) = \frac{w_k \exp[\frac{1}{2}(\tilde{\mathbf{v}}_p - \tilde{\boldsymbol{\mu}}_k)^T \tilde{\boldsymbol{\Sigma}}^{-1}(\tilde{\mathbf{v}}_p - \tilde{\boldsymbol{\mu}}_k)]}{\sum_i^K w_i \exp[\frac{1}{2}(\tilde{\mathbf{v}}_p - \tilde{\boldsymbol{\mu}}_i)^T \tilde{\boldsymbol{\Sigma}}^{-1}(\tilde{\mathbf{v}}_p - \tilde{\boldsymbol{\mu}}_i)]}, \quad (6)$$

where $\tilde{\boldsymbol{\mu}}_k$ and $\tilde{\boldsymbol{\Sigma}}_k$ are the mean and covariance of the two-dimensional spatial features for the kth Gaussian. The new posterior only considers the spatial distance between Gaussians and dense features, instead of the distance calculated among all dimensions. Spatial encoding improves the performance with well aligned images and reliable spatial information.

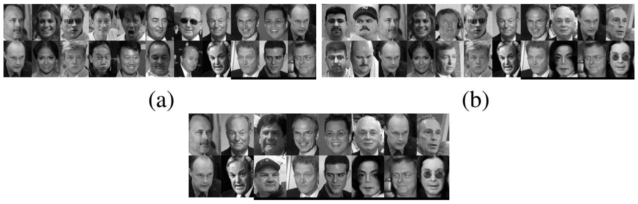

(a) (b) (c)

Fig. 2. Errors made by different methods on the split 10 of the LFW dataset. (a) FV-DCNN errors . (b) pool5 errors. (c) FV-DCNN + pool5 errors. Errors are significantly reduced when FV-DCNN and pool5 features are fused for verification.

While conducting our experiments, what we observed is that the error patterns are different between pool5 and FV-DCNN features. Figure 2 shows the errors made in the split 10 of the LFW dataset by FV-DCNN shown in (a) and pool5 features shown in (b). It is interesting to see how much the error is reduced when the FV-DCNN and pool5 features scores are fused. This can be seen by comparing the errors shown in (c) with (a) and (b) where (c) is attained by summing the similarity scores of pool5 and FV-DCNN with a linear weight.

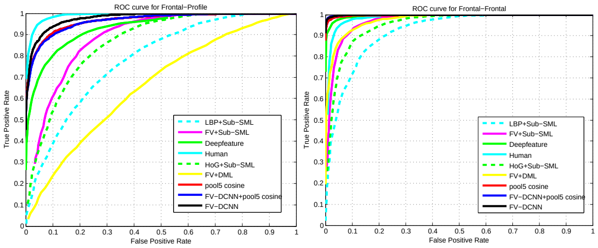
4. EXPERIMENTAL RESULTS

We evaluate the proposed FV-DCNN method on two challenging face verification datasets: Celebrities in Frontal-Profile (CFP) Dataset [29] and Labeled Face in the Wild (LFW) dataset [30]. The algorithms are evaluated using various measures, including the receiver operating characteristic curves (ROC), equal error rate (EER), area under curve (AUC), and accuracy based on the test protocols defined for each dataset.

For the LFW dataset, we learn sixty four Gaussians but use spatial encoding. A whitening PCA is applied as the initialization of joint Bayesian metric learning. For the CFP dataset, we learn 64 Gaussians and use traditional encoding. This is because the alignment for profile faces is not reliable. (*i.e.*, since only half of facial landmarks are available for the profile faces of the dataset, we use two eye corners of either left or right eye, nose tip, one mouth corner of either left or right side with similarity transform to the canonical coordinate defined for the frontal face as mentioned in Section 3.) Similarly, the whitening PCA is applied for initializing the joint Bayesian metric learning. Given the scores from the FV metric learning and the cosine distance of pool5 features, score level fusion is done by summing the similarity scores of pool5 and FV-DCNN with a linear weight.

Celebrities in Frontal-Profile Dataset (CFP)[29]: First, to investigate how pose variations influence the performance of the proposed FV-DCNN method, we conduct experiments on the recently introduced CFP face verification dataset. This dataset focuses on the unconstrained frontal to profile face verification protocol where most profile faces are in extreme poses. Sample face pairs are shown in Figure 3. The dataset contains 500 subjects, and each subject contains 10 frontal and 4 profile images. Similar to the LFW dataset, the CFP dataset consists of 20 splits in total, 10 for frontal-to-frontal and the other 10 for frontal-to-profile face verification tasks. Each split has 350 same and 350 different pairs, respectively. For this dataset, the human performance for the frontal-to-profile verification is 94.57% accuracy and frontal-to-frontal is 96.24% accuracy. The dataset has been evaluated in [29] using previous state-of-the-art algorithms, including Fisher vector based on SIFT features, Sub-SML [32], and a deep learning approach which uses a similar architecture and ReLU as the activation function without applying data augmentation.

Fig. 3. Sample image pairs from the Celebrities in Frontal-to-Profile dataset [29] where our method is able to successfully verify the pairs whereas both FV and DCNN-based methods fail.



(a) (b)

Fig. 4. The ROC curves corresponding to (a) Frontal-Profile matching and (b) Frontal-Frontal matching on the CFP dataset.

The evaluation results and the ROC curves are shown in Table 2 and Figure 4, respectively. From the figure, even though there exists the large performance drop in the frontal-to-profile setting, the proposed FV-DCNN approach still perform comparable to the human performance and better than pool5 features and other approaches, including the DCNN algorithm using ReLU. Since FV-DCNN encodes the spatial and appearance information contained in conv52

Method	#Net	Training Set	Metric	Mean Accuracy ± Std
DeepFace [18]	1	4.4 million images of 4,030 subjects, private	cosine	95.92% ± 0.29%
DeepFace	7	4.4 million images of 4,030 subjects, private	unrestricted, SVM	97.35% ± 0.25%
DeepID2 [11]	1	202,595 images of 10,117 subjects, private	unrestricted, Joint-Bayes	95.43%
DeepID2	25	202,595 images of 10,117 subjects, private	unrestricted, Joint-Bayes	99.15% ± 0.15%
DeepID3 [31]	50	202,595 images of 10,117 subjects, private	unrestricted, Joint-Bayes	99.53% ± 0.10%
FaceNet [8]	1	260 million images of 8 million subjects, private	L2	99.63% ± 0.09%
Yi et al. [9]	1	494,414 images of 10,575 subjects, public	cosine	96.13% ± 0.30%
Yi et al.	1	494,414 images of 10,575 subjects, public	unrestricted, Joint-Bayes	97.73% ± 0.31%
Wang et al. [27]	1	494,414 images of 10,575 subjects, public	cosine	96.95% ± 1.02%
Wang et al.	7	494,414 images of 10,575 subjects, public	cosine	97.52% ± 0.76%
Wang et al.	1	494,414 images of 10,575 subjects, public	unrestricted, Joint-Bayes	97.45% ± 0.99%
Wang et al.	7	494,414 images of 10,575 subjects, public	unrestricted, Joint-Bayes	98.23% ± 0.68%
Ding et al. [26]	8	471,592 images of 9,000 subjects, public	unrestricted, Joint-Bayes	99.02% ± 0.19%
Human, funneled [27]	N/A	N/A	N/A	99.20%
pool5 cosine	1	494,414 images of 10,575 subjects, public	cosine	97.82% ± 0.59%
FV-DCNN	1	494,414 images of 10,575 subjects, public	unrestricted, Joint-Bayes	97.72% ± 0.61%
FV-DCNN + pool5 cosine	1	494,414 images of 10,575 subjects, public	unrestricted, Joint-Bayes	98.13% ± 0.40%

Table 1. Performance comparison of different methods on the LFW dataset dataset.

	Frontal-Profile		
Algorithm	Accuracy	EER	AUC
HoG+Sub-SML	$77.31 \pm 1.61\%$	$22.20 \pm 1.18\%$	$85.97 \pm 1.03\%$
LBP+Sub-SML	$70.02 \pm 2.14\%$	$29.60 \pm 2.11\%$	$77.98 \pm 1.86\%$
FV+Sub-SML	$80.63 \pm 2.12\%$	$19.28 \pm 1.60\%$	$88.53 \pm 1.58\%$
FV+DML	$58.47 \pm 3.51\%$	$38.54 \pm 1.59\%$	$65.74 \pm 2.02\%$
Deep features	$84.91 \pm 1.82\%$	$14.97 \pm 1.98\%$	$93.00 \pm 1.55\%$
Human	$\mathbf{94.57 \pm 1.10\%}$	$\mathbf{5.02 \pm 1.07\%}$	$\mathbf{98.92 \pm 0.46\%}$
pool5	$90.41 \pm 1.16\%$	$9.63 \pm 1.21\%$	$96.53 \pm 0.99\%$
FV-DCNN+pool5	$89.83 \pm 1.88\%$	$10.40 \pm 1.85\%$	$96.37 \pm 0.97\%$
FV-DCNN	$\mathbf{91.97 \pm 1.70\%}$	$\mathbf{8.00 \pm 1.68\%}$	$\mathbf{97.70 \pm 0.82\%}$

	Frontal-Frontal		
Algorithm	Accuracy	EER	AUC
HoG+Sub-SML	$88.34 \pm 1.33\%$	$11.45 \pm 1.35\%$	$94.83 \pm 0.80\%$
LBP+Sub-SML	$83.54 \pm 2.40\%$	$16.00 \pm 1.74\%$	$91.70 \pm 1.55\%$
FV+Sub-SML	$91.30 \pm 0.85\%$	$8.85 \pm 0.74\%$	$96.87 \pm 0.39\%$
FV+DML	$91.18 \pm 1.34\%$	$8.62 \pm 1.19\%$	$97.25 \pm 0.60\%$
Deep features	$96.40 \pm 0.69\%$	$3.48 \pm 0.67\%$	$99.43 \pm 0.31\%$
Human	$\mathbf{96.24 \pm 0.67\%}$	$\mathbf{5.34 \pm 1.79\%}$	$\mathbf{98.19 \pm 1.13\%}$
pool5	$97.79 \pm 0.38\%$	$2.20 \pm 0.36\%$	$99.73 \pm 0.18\%$
FV-DCNN+pool5	$\mathbf{98.67 \pm 0.36\%}$	$\mathbf{1.40 \pm 0.37\%}$	$\mathbf{99.90 \pm 0.09\%}$
FV-DCNN	$98.41 \pm 0.45\%$	$1.54 \pm 0.43\%$	$99.89 \pm 0.06\%$

Table 2. Performance comparison of different methods on the CFP dataset where pool5 means ”pool5 + cosine distance”.

features into the high-dimensional feature vector, it is robust to large pose variations than other approaches. Also notice that by fusing FV-DCNN and pool5, we improve the performance for frontal-to-frontal setting. But frontal-to-profile setting is not as good as single FV-DCNN. This is because under extreme poses, global features are not robust and will degrade the overall performance.

Labeled Face in the Wild Dataset (LFW)[30]: We also evaluate our approach on the LFW dataset using the standard protocol for the face verification task which defines 3,000 positive pairs and 3,000 negative pairs in total. The pairs are further split into 10 disjoint subsets for cross validation, and each subset consists of 300 same and 300 different pairs. It contains 7,701 images of 4,281 subjects. We show the mean accuracy of the proposed FV-DCNN representation with the other state-of-the-art deep learning-based methods on the “funneled” LFW images: DeepFace [18], DeepID2 [11], DeepID3 [31], FaceNet [8], Yi *et al.* [9], Wang *et al.* [27], and human performance. The results are summarized in Table 1. As can be seen from this table, the proposed FV-CNN method performs comparable to many other deep learning-based methods. In addition, it also shows that the error reduces when we fuse the similarity scores of both FV-DCNN representation (local descriptor) and normal pool5 representation (global descriptor). Note that some of the deep learning-based methods compared in Table 1 use millions of data samples for training the model that typically has tens of millions of parameters or fuse multiple DCNN models together. In contrast, we use only the CASIA dataset which has less than 500K images to train a single DCNN model with about five million parameters.

Visualization of the Learnt GMMs: Figure 5 shows an image in the LFW dataset along with the last two dimensions (which are the spatial coordinates) of the Gaussians learnt by our method. Gaussians are learnt from the original 320-dimensional conv52 features plus two dimensional spatial features without dimension reduction, from the images in the LFW split 1. We only choose Gaussians whose corresponding energy in the learnt projection matrices are among the top eight or bottom eight, which corresponds to the discriminative power of these Gaussians. Figures 5(a) and 5(b) are Gaussians learnt after we apply square root and L_2 normalization on the conv52 features. Figures 5(c) and 5(d) are Gaussians learnt without any normalization. From Figures 5(a) and 5(b), the top eight Gaussians are located near eyes, nose and mouth after normalization. The bottom eight Gaussians are out of the face region in general. But in Figures 5(c) and (d), without pre-normalization, the top eight Gaussians are everywhere in the image with large variations in spatial location. Also, the bottom eight Gaussians are all located in the center of the face, which should not be the case. Comparison of these four figures shows that the spatial information is not encoded into Gaussians if we do not apply normalization before learning the Gaussians.

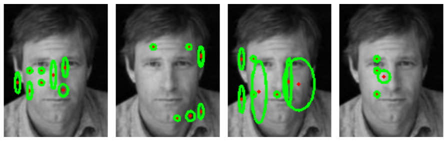

(a) (b) (c) (d)

Fig. 5. (a) Top eight Gaussians using square root and L_2 normalization. (b) Bottom eight Gaussians using square root and L_2 normalization. (c) Top eight Gaussians without normalization. (d) Bottom eight Gaussians without normalization.

5. CONCLUSIONS

In this paper, we proposed a FV-DCNN model for unconstrained face verification which combines FV with DCNN features. We demonstrated the effectiveness of the proposed method on the standard LFW and the challenging CFP datasets with large pose variations, respectively. It was shown that the FV-DCNN method can capture the local variations and the original DCNN pool5 features characterize the global variations. Performance gains were obtained by simply fusing their similarity scores without training another new DCNN model.

In future, we plan to incorporate hard positive and hard negative mining to perform metric learning, including joint Bayesian metric or triplet loss.

6. REFERENCES

[1] D. Chen, X. D. Cao, F. Wen, and J. Sun, "Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[2] K. Simonyan, O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Fisher vector faces in the wild," in *British Machine Vision Conference*, 2013, vol. 1, p. 7.

[3] O. M. Parkhi, K. Simonyan, A. Vedaldi, and A. Zisserman, "A compact and discriminative face track descriptor," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[4] Jun-Cheng Chen, Swami Sankaranarayanan, Vishal M. Patel, and Rama Chellappa, "Unconstrained face verification using fisher vectors computed from frontalized faces," in *IEEE International Conference on Biometrics: Theory, Applications and Systems*, 2015.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

[6] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *arXiv preprint arXiv:1409.4842*, 2014.

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 580–587.

[8] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," *arXiv preprint arXiv:1503.03832*, 2015.

[9] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Learning face representation from scratch," *arXiv preprint arXiv:1411.7923*, 2014.

[10] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[11] Y. Sun, X. Wang, and X. Tang, "Deeply learned face representations are sparse, selective, and robust," *arXiv preprint arXiv:1412.1265*, 2014.

[12] M. Cimpoi, S. Maji, and A. Vedaldi, "Deep filter banks for texture recognition and segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[13] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 2037–2041, 2006.

[14] B. C. Zhang, S. G. Shan, X. L. Chen, and W. Gao, "Histogram of Gabor phase patterns (hgpp): a novel object representation approach for face recognition," *IEEE Transactions on Image Processing*, vol. 16, no. 1, pp. 57–68, 2007.

[15] Gabriella Csurka, Christopher R. Dance, Lixin Fan, Jutta Willamowski, and Cdric Bray, "Visual categorization with bags of keypoints," in *Workshop on Statistical Learning in Computer Vision, ECCV*, 2004, pp. 1–22.

[16] H. Jegou, M. Douze, C. Schmid, and P. Perez, "Aggregating local descriptors into a compact image representation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 3304–3311.

[17] Xi Zhou, Kai Yu, Tong Zhang, and Thomas S. Huang, "Image classification using super-vector coding of local image descriptors," in *European Conference on Computer Vision*, 2010, pp. 141–154.

[18] Y. Taigman, M. Yang, M. A. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1701–1708.

[19] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," in *Advances in Neural Information Processing Systems*, 2014, pp. 1988–1996.

[20] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," *British Machine Vision Conference*, 2015.

[21] P. Viola and M. J. Jones, "Robust real-time face detection," *IJCV*, vol. 57, no. 2, pp. 137–154, 2004.

[22] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1867–1874.

[23] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the fisher kernel for large-scale image classification," in *In European Conference on Computer Vision*, pp. 143–156. 2010.

[24] D. Chen, X. D. Cao, L. W. Wang, F. Wen, and J. Sun, "Bayesian face revisited: A joint formulation," in *European Conference on Computer Vision*, pp. 566–579. 2012.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," *arXiv preprint arXiv:1502.01852*, 2015.

[26] C. Ding and D. Tao, "Robust face recognition via multimodal deep face representation," *arXiv preprint arXiv:1509.00244*, 2015.

[27] D. Wang, C. Otto, and A. K. Jain, "Face search at scale: 80 million gallery," *arXiv preprint arXiv:1507.07242*, 2015.

[28] J.-C. Chen, V. M Patel, and R. Chellappa, "Unconstrained face verification using deep cnn features," *arXiv preprint arXiv:1508.01722*, 2015.

[29] S. Sengupta, J-C. Chen, C. Castillo, V. M. Patel, R. Chellappa, and D. Jacobs, "Frontal to profile face verification in the wild," in *IEEE Winter Conference on Applications of Computer Vision*, 2016.

[30] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," in *Workshop on Faces in Real-Life Images: Detection, Alignment, and Recognition*, 2008.

[31] Yi Sun, Ding Liang, Xiaogang Wang, and Xiaoou Tang, "Deepid3: Face recognition with very deep neural networks," *CoRR*, vol. abs/1502.00873, 2015.

[32] Q. Cao, Y. Ying, and P. Li, "Similarity metric learning for face recognition," in *IEEE International Conference on Computer Vision (ICCV)*, 2013, pp. 2408–2415.