Unconstrained Face Verification Using Fisher Vectors Computed From Frontalized Faces

Jun-Cheng Chen, Swami Sankaranarayanan, Vishal M. Patel and Rama Chellappa
Center for Automation Research
University of Maryland, College Park, MD 20742
{pullpull, swamiviv, pvishalm, rama}@umiacs.umd.edu

Abstract

We present an algorithm for unconstrained face verification using Fisher vectors computed from frontalized off-frontal gallery and probe faces. In the training phase, we use the Labeled Faces in the Wild (LFW) dataset to learn the Fisher vector encoding and the joint Bayesian metric. Given an image containing the query face, we perform face detection and landmark localization followed by frontalization to normalize the effect of pose. We further extract dense SIFT features which are then encoded using the Fisher vector learnt during the training phase. The similarity scores are then computed using the learnt joint Bayesian metric. CMC curves and FAR/TAR numbers calculated for a subset of the IARPA JANUS challenge dataset are presented.

1. Introduction

Face recognition/verification has been one of the core problems in computer vision and has been actively researched for over two decades [32]. Many algorithms have been shown to work well on images that are collected in controlled settings. However, the performance of these algorithms often degrades significantly on images that have large variations such as pose, illumination, expression, aging, cosmetics, and occlusion.

To deal with this problem, many methods have focused on finding invariant and discriminative representations from face images and videos. For instance, it was shown in [6] that the high-dimensional multi-scale Local Binary Pattern (LBP) features extracted from local patches centered at each facial landmarks can find discriminative representation for face recognition. Recognition methods based on Fisher vector (FV) representation have also been considered. In particular, FV representation has shown to work well for face recognition problems [23], [21]. In these methods, the FV is applied to the videos by pooling the features extracted from each frame or averaging the encoded FVs over the frames. Even though the FV descriptors are compact for videos and produce discriminative features for verification, they often fail when faces contain large pose variations.

To mitigate this pose problem in FV encoding, we present a method which essentially performs FV encoding on frontalized images. The overview of our method is shown in Figure 1. The common preprocessing steps in training and testing phases include face/landmark detection on the input image, face frontalization to compensate for pose variations, and local feature extraction. In the training phase, these local features are pooled together to learn a Gaussian Mixture Model (GMM) whose means and covariances are used in the FV encoding procedure. In order to get a more efficient representation of the encoded features, a metric is learnt using the joint Bayesian metric learning procedure. In the testing phase, given a face verification pair, for each image of the pair, the preprocessing steps are performed and the local features are encoded using the GMM learnt in the training stage. The similarity scores are then computed using the learnt metric. All these are done independently for each image of the verification pair.

The rest of the paper is organized as follows. We briefly review some related works in Section 2. Details of the different components of the proposed approach which include frontalization, FV representation and joint Bayesian metric learning are given in Section 3. In Section 4, we present the protocol for the JANUS CS0 dataset and present results and comparisons with commercial face matchers. We conclude the paper in Section 5 with a brief summary and discussion.

2. Related Work

In this section, we briefly review several recent works on face verification and related problems. In particular, we survey recent feature learning, metric learning and pose normalization methods.

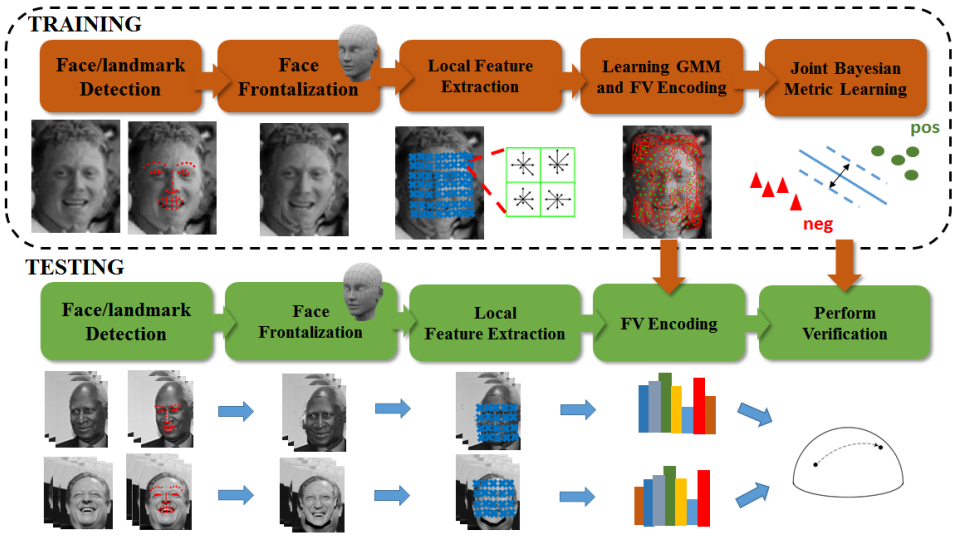


Figure 1: An overview of our Fisher vector representation algorithm on frontalized faces for face verification.

2.1. Feature Learning

Learning invariant and discriminative representation is the first step towards a successful face verification system. Ahonen *et al.* [1] showed that LBP is effective for face recognition. Several variants of LBP have been proposed: Local Ternary Patterns (LTP) [25] and three-patch LBP (TP-LBP) [29]. Like LBP, Gabor wavelets [31][30] have been widely used to encode multi-scale and multi-orientation information for given face images. On the other hand, Coates *et al.* [11] showed that over-complete representation is critical for achieving high recognition rates regardless of the encoding methods. In [4], it was shown that densely sampling overlapped image patches helps to improve the recognition performance. For still-face recognition, Chen *et al.* [7] demonstrated excellent results using the high-dimensional multi-scale LBP features extracted from patches centered at dense facial landmarks. These works showed that over-complete and high-dimensional features are effective for face recognition. Li *et al.* [20] proposed a probabilistic elastic model which learned a GMM using dense local spatial-appearance features, selected sparse representative features for each Gaussian, and finally concatenated those features into a high-dimensional vector. This method is effective for face verification through matching the correspondence between facial parts of a pair of images (i.e. each Gaussian represents a facial part). On the contrary, Simonyan *et al.* [23], Parkhi *et al.* [21], and Chen *et al.* [8] showed that FV, a feature encoding method widely used for object and image classification, can be successfully applied to face recognition. Their experiments showed that FV effectively encode over-complete and dense features into generating a robust representation. In addition, Chen *et al.* [9][10] proposed a video-based dictionary framework. Each video dictionary can be learned independently for each video and can effectively model the face variations for a video with joint-sparsity constraints.

2.2. Metric Learning

The similarity measure is the other key component in a face verification system. Due to the large volume of metric learning approaches in the literature, we briefly review several works on learning a discriminative metric for verification problems. Guillaumin *et al.* [15] proposed to learn two robust distance measures: Logistic Discriminant-based Metric Learning (LDML) and Marginalized kNN (MkNN). The LDML method learns a distance through performing logistic discriminant analysis on a set of labeled image pairs and the MkNN method marginalizes a k-nearest-neighbor classifier to both images of the given test pair using a set of labeled training images. Taigman *et al.* [24] learned the Mahalanobis distance for face verification using the Information Theoretic Metric Learning (ITML) method proposed in [12]. Wolf *et al.* [28] proposed the one-shot similarity (OSS) kernel based on a set of pre-selected reference images mutually exclusive to the pair of images being compared and training a discriminative classifier between the test image and the new reference set. Kumar *et al.* [19] proposed two classifiers for face verification: attribute classifier and simile classifiers. Attribute classifiers are a set of binary classifiers used to detect the presence of certain visual concepts where visual concepts are defined in advance. Simile classifiers were trained to measure the similarities of facial parts of a person to specific reference people. Chen *et al.* [6] proposed a joint Bayesian approach which models the joint distribution of a pair of face images instead of modeling the difference vector of them and uses the ratio of

between-class and within-class probabilities as the similarity measure.

2.3. Pose Normalization

Traditionally, pose normalization for face recognition has used similarity transform-based warping using the 2D coordinates of several fiducial points computed on the input face, the most relevant being the FV face recognition work of Simonyan *et al.* [23]. 3D face models have been used to generate a frontal face warp since the work of Blanz *et al.*. [3] The drawbacks of such methods are the time needed to fit the model, availability of a 3D database during test time and non-robustness to unconstrained settings. A more detailed survey of recent pose normalization methods can be found in [13]. In this work, we use the frontalization framework of Hassner *et al.* [16] which provides a fast and efficient solution to the problem of generating a frontal face warp using a single reference 3D model. To make the method more robust to the failures of the landmark detection process, we ignore the landmark points of the face contour in the frontalizing process.

3. Proposed Approach

In this section, we present the details of our method based on FV encoding of frontalized faces (FVFF) for face verification.

3.1. Face Frontalization

The first step of our algorithm is to make non-frontal faces frontal. In this section, we describe the method used to obtain a frontalized warp of the given face image. To generate a frontal warp we use the given 2D-3D correspondences of a reference image from the USF-HUMAN ID database for which we can precompute the camera projection matrix. Specifically, the frontalization procedure is outlined below:

- A random image from the USF-HUMAN ID database is taken as the reference and rendered in the frontal view. Let $\mathbf{C}_{ref}$ be the corresponding camera matrix. Facial landmark detection on the frontalized image is performed and using the camera matrix the corresponding 3D points are computed. Let $\mathbf{p}_{ref}$ be the 2D coordinates of the landmark points, then the 3D coordinates $\mathbf{P}_{ref}$ are fixed using the following equation:

$$\mathbf{p}_{ref} = \mathbf{C}_{ref} \cdot \mathbf{P}_{ref}.$$

- Given an input image in a non-frontal pose, let $\mathcal{L}_{inp}$ be the detected landmark points. Then, the reference 3D coordinates $\mathbb{P}_{ref}$ are used as the 3D surface coordinates from which the given input view is generated, which then allows the calculation of the rotation matrix $\mathbb{R}_{inp}$ and the translation vector $\mathbf{t}_{inp}$ that correspond to the pose in the given input.

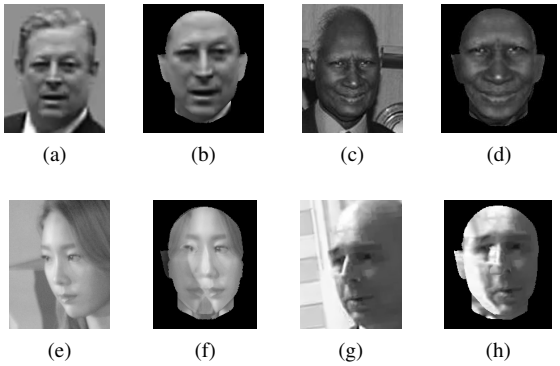


Figure 2: Input images and frontalized outputs:(a)-(d): Good examples. (e)-(h): Examples with artifacts.

- Using the estimated rotation and translation parameters, each pixel in the input image is warped onto a location in the frontal view, the color at each pixel computed using bicubic interpolation.

More details of this method can be found in [16]. We use the source code provided by the authors to generate the frontal warp. The frontalization process can be performed using the output of any face detector. For this work, we have used [2] which provides 66 landmark points for a face image. To be robust to extreme pose variations, we ignore the landmark points corresponding to the face contour for the frontalization procedure. Since, the amount of landmark points is very sparse, we encounter some artifacts in the frontalized image corresponding to the non-frontal poses, as shown in Figure 2.

3.2. Fisher Vector Representation

The Fisher vector is one of bag-of-visual-word encoding methods which aggregates a large set of local features into a high-dimensional vector. In general, the FV is extracted by fitting a parametric generative model for the features and encoding them using the derivatives of the log-likelihood of the learned model with respect to the model parameters. As in [22], a Gaussian mixture model (GMM) with diagonal covariances is used in this work. In addition, the first-and second-order statistics of the features with respect to each component are computed as follows:

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

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

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

where w_k, $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k = diag(\boldsymbol{\sigma}_{1k}, ..., \boldsymbol{\sigma}_{dk})$ 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 can be learned from the training data using the EM algorithm. $\alpha_k(\mathbf{v}_p)$ is the weight of $\mathbf{v}_p$ belonging to the kth mixture component. The final FV, $\boldsymbol{\Phi}(\mathbf{I})$, of an image $\mathbf{I}$ is obtained by concatenating all the $\boldsymbol{\Phi}_k^{(1)}$ and $\boldsymbol{\Phi}_k^{(2)}$s into a high-dimensional vector $\boldsymbol{\Phi}(\mathbf{I}) = [\boldsymbol{\Phi}_1^{(1)}, \boldsymbol{\Phi}_1^{(2)}, ..., \boldsymbol{\Phi}_K^{(1)}, \boldsymbol{\Phi}_K^{(2)}]$, whose dimensionality is $2Kd$ where K is the number of mixture components and d is the dimensionality of the extracted features.

In this work, we use the dense SIFT features as local features. To incorporate spatial information, each SIFT feature is augmented with the normalized x and y coordinates [20][23] as $[\mathbf{a}_{xy}, \frac{x}{w} - \frac{1}{2}, \frac{y}{h} - \frac{1}{2}]^T$, where $\mathbf{a}_{xy}$ is the SIFT descriptor at (x, y), and w and h are the width and height of the image, respectively. To satisfy the diagonal covariance assumption, all the SIFT features are de-correlated with PCA first. In this paper, we use $K = 512$ number of components and $d = 66$ feature dimensionality after augmentation. In addition, each FV is also processed with signed square-rooting and L_2 normalization as suggested in [22]. To extract the FV from an image set and videos, one can either (1) pool all the SIFT features extracted from each face image/frame together into a large feature matrix and then perform FV encoding on the pooled matrix, or (2) perform FV encoding to the features of individual face image/frame and then average all the FVs into one. All the experiments in this paper are performed using (2).

3.3. Joint Bayesian Metric Learning

Once feature representations for two face images/videos have been extracted, we compute their similarity. One of the simplest similarity measures is the Euclidean distance. However, because of the high-dimensionality of the FV and complex distribution of face images/videos, directly applying it usually results in unsatisfactory performance. Recently, the joint Bayesian method to face metric learning has been shown to achieve good performance for face verification [6][5]. Instead of modeling the difference vector between the two faces, this approach directly models the joint distribution of feature vectors of both ith and jth images, $\{\mathbf{x}_i, \mathbf{x}_j\}$, as a Gaussian. Let $P(\mathbf{x}_i, \mathbf{x}_j|H_I) \sim N(0, \boldsymbol{\Sigma}_I)$ when $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to the same class, and $P(\mathbf{x}_i, \mathbf{x}_j|H_E) \sim N(0, \boldsymbol{\Sigma}_E)$ when they are from different classes. In addition, each face vector can be 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 assumed to be independent zero-mean Gaussian distributions, $N(0, \mathbf{S}_\mu)$ and $N(0, \mathbf{S}_\epsilon)$, respectively. Then, the covariances for intra-class, $\boldsymbol{\Sigma}_I$, and for inter-class, $\boldsymbol{\Sigma}_E$, can be derived as follows

$$\boldsymbol{\Sigma}_I = \begin{bmatrix} \mathbf{S}_\mu + \mathbf{S}_\epsilon & \mathbf{S}_\mu \\ \mathbf{S}_\mu & \mathbf{S}_\mu + \mathbf{S}_\epsilon \end{bmatrix}, \boldsymbol{\Sigma}_E = \begin{bmatrix} \mathbf{S}_\mu + \mathbf{S}_\epsilon & 0 \\ 0 & \mathbf{S}_\mu + \mathbf{S}_\epsilon \end{bmatrix}. \tag{4}$$

The log likelihood ratio of intra- and inter-classes, $r(\mathbf{x}_i, \mathbf{x}_j)$, which has a closed-form solution can be computed as follows:

$$r(\mathbf{x}_i, \mathbf{x}_j) = \log \frac{P(\mathbf{x}_i, \mathbf{x}_j|H_I)}{P(\mathbf{x}_i, \mathbf{x}_j|H_E)} = \mathbf{x}_i^T\mathbf{M}\mathbf{x}_i + \mathbf{x}_j^T\mathbf{M}\mathbf{x}_j - 2\mathbf{x}_i^T\mathbf{R}\mathbf{x}_j, \tag{5}$$

where

$$\mathbf{M} = (\mathbf{S}_\mu + \mathbf{S}_\epsilon)^{-1} - (\mathbf{F} + \mathbf{R}) \tag{6}$$

$$\begin{bmatrix} \mathbf{F} + \mathbf{R} & \mathbf{R} \\ \mathbf{R} & \mathbf{F} + \mathbf{R} \end{bmatrix} = \boldsymbol{\Sigma}_I^{-1}. \tag{7}$$

More details of this method can be found in [6]. Both $\mathbf{S}_\mu$ and $\mathbf{S}_\epsilon$ can be estimated using the EM algorithm which optimizes the similarity measure indirectly. Instead of using the EM algorithm, we optimize the closed-form distance in a large-margin framework. Equation (5) can be derived into the form as $(\mathbf{x}_i - \mathbf{x}_j)^T\mathbf{M}(\mathbf{x}_i - \mathbf{x}_j) - 2\mathbf{x}_i^T(\mathbf{R} - \mathbf{M})\mathbf{x}_j$. Directly learning of $\mathbf{M} \in \mathbb{R}^{D\times D}$ and $\mathbf{R} \in \mathbb{R}^{D\times D}$ are intractable because of the high dimensionality of FVs, where $D = 2Kd$. Let $\mathbf{M} = \mathbf{H}^T\mathbf{H}$ and $\mathbf{B} = (\mathbf{R} - \mathbf{M}) = \mathbf{V}^T\mathbf{V}$, where $\mathbf{H} \in \mathbb{R}^{r\times D}$ and $\mathbf{V} \in \mathbb{R}^{r\times D}$. With this definitions, we choose $r = 128 \ll D$ in our work. Finally, we solve the following optimization problem

$$\begin{aligned} \underset{\mathbf{H},\mathbf{V},b}{\operatorname{argmin}} \sum_{i,j} \quad & max[1 - y_{ij}(b - (\mathbf{x}_i - \mathbf{x}_j)^T\mathbf{H}^T\mathbf{H}(\mathbf{x}_i - \mathbf{x}_j) \\ & + 2\mathbf{x}_i^T\mathbf{V}^T\mathbf{V}\mathbf{x}_j), 0], \end{aligned} \tag{8}$$

where $b \in \mathbb{R}$ is the threshold, and y_{ij} is the label of a pair: $y_{ij} = 1$ if person i and j are the same and $y_{ij} = -1$, otherwise. For simplification, we denote $(\mathbf{x}_i - \mathbf{x}_j)^T\mathbf{H}^T\mathbf{H}(\mathbf{x}_i - \mathbf{x}_j) - 2\mathbf{x}_i^T\mathbf{V}^T\mathbf{V}\mathbf{x}_j$ as $d_{\mathbf{H},\mathbf{V}}(\mathbf{x}_i, \mathbf{x}_j)$. In addition, $\mathbf{H}$ and $\mathbf{V}$ are updated using stochastic gradient descent as follows and are equally trained on positive and negative pairs in turn:

$$\begin{aligned} \mathbf{H}_{t+1} &= \begin{cases} \mathbf{H}_t, & \text{if } y_{ij}(b_t - d_{\mathbf{H},\mathbf{V}}(\mathbf{x}_i, \mathbf{x}_j)) > 1 \\ \mathbf{H}_t - \gamma y_{ij}\mathbf{H}_t\boldsymbol{\Psi}_{ij}, & \text{otherwise,} \end{cases} \\ \mathbf{V}_{t+1} &= \begin{cases} \mathbf{V}_t, & \text{if } y_{ij}(b_t - d_{\mathbf{H},\mathbf{V}}(\mathbf{x}_i, \mathbf{x}_j)) > 1 \\ \mathbf{V}_t + \gamma y_{ij}\mathbf{V}_t\boldsymbol{\Gamma}_{ij}, & \text{otherwise,} \end{cases} \\ b_{t+1} &= \begin{cases} b_t, & \text{if } y_{ij}(b_t - d_{\mathbf{H},\mathbf{V}}(\mathbf{x}_i, \mathbf{x}_j)) > 1 \\ b_t + \gamma_b y_{ij}, & \text{otherwise,} \end{cases} \end{aligned} \tag{9}$$

where $\boldsymbol{\Psi}_{ij} = (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T$, $\boldsymbol{\Gamma}_{ij} = \mathbf{x}_i\mathbf{x}_j^T + \mathbf{x}_j\mathbf{x}_i^T$, γ is the learning rate for $\mathbf{H}$ and $\mathbf{V}$, and γ_b for the bias b. We perform the whitening PCA to the extracted features and initialize both $\mathbf{H}$ and $\mathbf{V}$ with r largest eigenvectors. Note that $\mathbf{H}$ and $\mathbf{V}$ are updated only when the constraints are violated. The training and testing algorithms are summarized in Algorith 1 and Algorithm 2, respectively.

Algorithm 1 TRAINING

Input: (a) Training images and labels with positive and negative pairs from LFW dataset [17] and (b) maxIter iterations.

Output: (a) Model parameters of Gaussians, $\boldsymbol{\mu}_i$, $\boldsymbol{\Sigma}_i$, and w_i for $i = 1...K$, and (b) projection matrices learned from metric learning, $\mathbf{H}$ and $\mathbf{V}$.

1: Perform face and landmark detection for each training images.
2: Apply the face frontalization step discussed in Section 3.1.
3: Extract multi-scale dense root-SIFT features from the whole frontalized face and augment them with normalized x and y coordinates.
4: Learn a K-component GMM ($\boldsymbol{\mu}_i$, $\boldsymbol{\Sigma}_i$, and w_i) for the dense features using EM algorithm
5: Perform FV encoding to the feature vectors.
6: Apply stochastic gradient descent using the training positive and negative face pairs in turn to optimize (8) until the maxIter iteration is reached to learn $\mathbf{H}$ and $\mathbf{V}$.

Algorithm 2 TESTING

Input: (a) Model parameters of Gaussians, $\boldsymbol{\mu}_i$, $\boldsymbol{\Sigma}_i$, and w_i for $i = 1...K$, (b) target and query images/video frames, $\{\mathbf{T}\}_{i=1}^{N_t}$ and $\{\mathbf{Q}\}_{i=1}^{N_q}$, (c) projection matrices $\mathbf{H}$ and $\mathbf{V}$ which are used to measure face similarity between a pair of images/video frames.

Output: Similarity matrix, $\mathbf{S}$.

1: Perform face and landmark detection for each target and query images/video frames.
2: Perform the same frontalization techniques to all the cropped faces of target and query images/video frames.
3: Extract multi-scale dense root-SIFT features using the whole face of testing face images/video frames and augment them with normalized x and y coordinates.
4: Perform FV encoding to feature vectors of frames of a video using the learned $\boldsymbol{\mu}_i$, $\boldsymbol{\Sigma}_i$, and w_i for $i = 1...K$. and average all of them as the final descriptor.
5: Apply the learned joint Bayesian metric to each testing pair of faces to get the face similarity matrix, $\mathbf{S}$.

4. Experimental Results

We evaluated the performance of the proposed method on a subset of the challenging IARPA Janus Benchmark A (IJB-A) [18]. The receiver operating characteristic (ROC) curves and the cumulative match characteristic (CMC) scores are used to evaluate the performance of different algorithms. The ROC curve measures the performance in the verification scenarios, and the CMC score measures accuracy in a closed set identification scenarios.

4.1. IARPA Janus Benchmark Challenge Set 0

The IARPA Janus Benchmark Challenge Set 0 (IJB-CS0) is the first released version of the IJB-A dataset [18] and the IJB-CS0 is a subset of IJB-A. The IJB-CS0 dataset has 150 subjects in total with 2103 images and 858 videos split into 7438 frames. Resolutions of the images differ. Sample images and video frames from this dataset are shown in Fig. 3. It contains a variety of challenging conditions on pose, illumination, resolution, and image quality. The dataset is divided into training and testing sets. The training set contains 100 subjects and the testing set contains the remaining 50 subjects with no overlapping subjects between the two sets. Ten random splits of training and testing are provided by the benchmark. For the test set, the image and video frames of each subject are randomly split into gallery and probe with no overlap between them. Moreover, each subject's imagery is further separated into protocol A and B using their genuine match scores produced by a commercial matcher such that protocol B is much harder than protocol A.

Unlike the Label Face in the Wild (LFW) [17] and Youtube Face (YTF) [27] datasets which only use a sparse set of negative pairs to evaluate the verification performance, IJB-CS0 protocol divides the images/video frames into gallery and probe sets so that it uses all the available positive and negative pairs for the evaluation. In addition, each gallery and probe set consist of multiple templates. That is, one template corresponds to a subset of images/video frames of one subject in gallery and different templates may correspond to the same subject in probe. Each template contains a combination of images or frames sampled from multiple image sets or videos of a subject. For example, the size of the similarity matrix for split1 A is 50×441 where 50 for the gallery and 441 for the probe (the same subject reappears in several templates). Moreover, some templates contain only one profile face in low quality. Thus, traditional video-based face verification algorithms can not be directly applied to this dataset. Finally, the dataset contains faces with full pose and illumination variations. In contrast, both the LFW and the YTF datasets only include faces detected by the Viola Jones face detector [26]. These factors essentially make the IJB-CS0 a challenging face dataset.

We compare the results of the proposed method with the FV method (i.e. without pose normalization) and three other commercial off-the-shelf matchers, COTS1, COTS2, and GOTS. The COTS and GOTS baselines provided by IJB-CS0 are the top performers from the most recent NIST FRVT study [14]. The performance of our frontalization algorithm relies on the quality of detected landmarks. The extreme pose is still a challenging problem to the landmark detector. Therefore, we use the pose information estimated from the landmark detector and select face images/video frames whose yaw angle are less than or equal to $\pm$ 25 degrees for each gallery and probe set. If there are no images/frames satisfying the constraint, we choose the most frontal one. For a fair comparison, the same frame selection is also used for the FV method. Moreover, all the training images of subjects in LFW who also appear in IJB-CS0

are removed for both FV and FVFF. Figures 4 and 5 show the ROC curves and the CMC curves, respectively for the verification results using the previously described protocol. From the ROC and CMC curves, we see that the proposed method performs better than the FV method. This can be attributed to the fact that our frontalization method improves the encoding of faces compared with the similarity transform which is used in the traditional FV encoding.

4.2. Discussion

For the FV method, before feature extraction, we apply self-quotient image [25] to normalize the illumination which usually improves the performance. Nevertheless, the same illumination normalization is not effective on the frontalized faces due to the artifacts introduced by frontalization. These artifacts are especially significant at regions surrounding the boundary of the face contour. This is the main reason why we do not apply it to the proposed method. However, when we fuse the similarity matrices from the FV and the proposed method, improved results are seen. This essentially shows that the FV and the proposed method complement each other. The performance after the fusion is comparable to the COTS in protocol A and much better in the harder protocol B. To give the readers a clear quantitative results, we also summarize all the TARs of different approaches in Table 1 when FAR = 0.1,0.01,0.001 and 0.0001. Similarly, the CMC scores are illustrated in Table 2. These numbers also show how the proposed method improves the performance of the FV in both identification and verification settings.

Interestingly, the IJB-CS0 dataset comes with several attributes, (1) forehead visibility, (2) eye visibility, (3) nose/mouth visibility, (4) indoor/outdoor, (5) gender, (6) skin tone, and (7) age. To investigate the effectiveness of different attributes on face verification, we use the gender attribute to combine it with the fusion results from the FV and the proposed method. Gender is a strong attribute as the gender information of a pair of faces essentially indicate whether the faces correspond to a same subject or not. Therefore, if the genders are different, we directly assign -Inf to the corresponding entry of the similarity matrix. From the experimental results, we find that the gender attribute mainly helps to improve the right half of the ROC curves close to FAR=0.1 for both protocols A and B and improves a little for the left half part. The ratio of male to female over 10 splits is around 1:1 for gallery template sets and 3:2 for probe template sets.

4.3. Runtime

On average, the frontalization per image using the given face bounding box from the metadata takes around 5 seconds (including the landmark detection time). In addition, the SIFT feature extraction and the FV encoding with 512 components takes around 0.5 second and 2 second per image, respectively. To verify a pair of templates takes 0.2 second using the average FV feature . The entire experiments are performed in a cluster with 64 cores of AMD Opteron 6274 processors (2.2 Ghz per core) with 128GB memory. Frontalization and feature extraction parts can be fully parallelized and distributed over all available cores. However, we use an iterative algorithms for training the model which can only run in sequential mode utilizing single core. It is the most time consuming part that takes around 4.5 hours (EM for GMM) and 9 hours for the SGD metric learning.

Figure 3: Sample images and frames from the IJB-CS0 dataset. A variety of challenging variations on pose, illumination, resolution, occlusion, and image quality are present in these images.

5. Conclusion

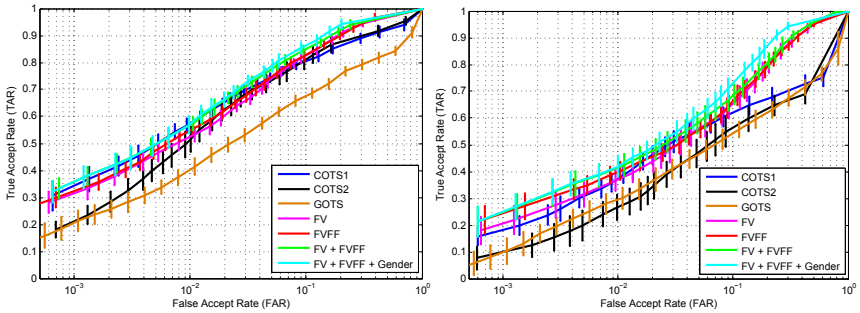
In this paper, we presented a method based on FV representation on frontalized faces for face verification. Our method essentially takes advantage of the discriminative power of features extracted from frontalized images. Furthermore, the joint Bayesian metric learning is applied to learn the projection matrices to reduce the feature dimensionality for efficiency and improving discriminative performance. Preliminary experiments on the challenging Janus dataset demonstrate the effectiveness of our proposed approach.

6. Acknowledgments

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.

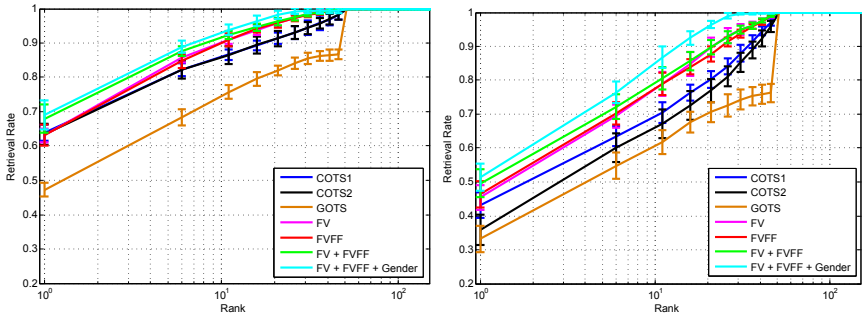
References

[1] T. Ahonen, A. Hadid, and M. Pietikainen. Face description with local binary patterns: Application to face recognition.



(a) (b)

Figure 4: (a) The average ROC curves with the standard deviation for the CS0-A dataset and (b) the CS0-B dataset over 10 splits. Our proposed FVFF method performs better than the FV method.



(a) (b)

Figure 5: (a)The average CMC curves with the standard deviation for the CS0-A dataset and (b) the CS0-B dataset over 10 splits. Our proposed FVFF performs better than the FV method.

FAR@CS0-A	COTS1	COTS2	GOTS	FV	FVFF	FV + FVFF	FV + FVFF + Gender
1e-4	0.166±0.044	0.057±0.04	0.039±0.038	0.18±0.055	0.162±0.042	**0.194±0.041**	0.189±0.041
1e-3	0.346±0.048	0.209±0.034	0.205±0.042	0.32±0.042	0.323±0.03	0.357±0.038	**0.358±0.044**
1e-2	0.575±0.026	0.512±0.047	0.403±0.032	0.527±0.028	0.55±0.019	0.568±0.027	**0.573±0.031**
1e-1	0.805±0.018	0.814±0.032	0.676±0.02	0.827±0.020	0.829±0.017	0.852±0.019	**0.864±0.019**
FAR@CS0-B	COTS1	COTS2	GOTS	FV	FVFF	FV + FVFF	FV + FVFF + Gender
1e-4	0.084±0.029	0.034±0.025	0.011±0.017	0.093±0.044	0.133±0.045	0.130±0.033	**0.130±0.033**
1e-3	0.183±0.036	0.096±0.044	0.099±0.047	0.207±0.052	0.245±0.047	0.251±0.049	**0.251±0.047**
1e-2	0.373±0.032	0.269±0.046	0.295±0.039	0.371±0.038	0.398±0.033	0.412±0.045	**0.42±0.039**
1e-1	0.621±0.039	0.567±0.052	0.545±0.032	0.656±0.042	0.665±0.035	0.684±0.038	**0.728±0.355**

Table 1: The TARs of all the approaches at FAR=0.1, 0.01, 0.001, and 0.0001 for the ROC curves for the both protocols A and B of the IJB-CS0 dataset.

CMC@CS0-A	COTS1	COTS2	GOTS	FV	FVFF	FV + FVFF	FV + FVFF + Gender
Rank-1	0.639±0.024	0.635±0.03	0.473±0.02	0.631±0.02	0.632±0.03	0.68±0.041	**0.691±0.043**
Rank-5	0.802±0.019	0.81±0.029	0.668±0.024	0.841±0.011	0.832±0.017	0.861±0.017	**0.874±0.017**
CMC@CS0-B	COTS1	COTS2	GOTS	FV	FVFF	FV + FVFF	FV + FVFF + Gender
Rank-1	0.432±0.038	0.358±0.045	0.332±0.038	0.455±0.036	0.464±0.038	0.496±0.042	**0.514±0.04**
Rank-5	0.617±0.034	0.58±0.048	0.529±0.037	0.668±0.042	0.675±0.032	0.697±0.037	**0.732±0.035**

Table 2: The Rank-1 and Rank-5 retrieval accuracies for the both protocols A and B of the IJB-CS0 dataset.

IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(12):2037–2041, 2006.

[2] A. Asthana, S. Zafeiriou, S. Y. Cheng, and M. Pantic. Robust discriminative response map fitting with constrained local models. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3444–3451, 2013.

[3] V. Blanz and T. Vetter. Face recognition based on fitting a 3d morphable model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(9):1063–1074, 2003.

[4] Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce. Learning mid-level features for recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2559–2566, 2010.

[5] X. D. Cao, D. Wipf, F. Wen, G. Q. Duan, and J. Sun. A practical transfer learning algorithm for face verification. In *IEEE International Conference on Computer Vision*, pages 3208–3215. IEEE, 2013.

[6] 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*, pages 566–579. Springer, 2012.

[7] 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.

[8] J.-C. Chen, V. M. Patel, and R. Chellappa. Landmark-based Fisher vector representation for video-based face verification. In *IEEE Conference on Image Processing*, 2015.

[9] Y.-C. Chen, V. M. Patel, P. J. Phillips, and R. Chellappa. Dictionary-based face recognition from video. In *European Conference on Computer Vision*, pages 766–779. 2012.

[10] Y.-C. Chen, V. M. Patel, S. Shekhar, R. Chellappa, and P. J. Phillips. Video-based face recognition via joint sparse representation. In *IEEE conference on Automatic Face and Gesture Recognition*, 2013.

[11] A. Coates, A. Y. Ng, and H. L. Lee. An analysis of single-layer networks in unsupervised feature learning. In *International Conference on Artificial Intelligence and Statistics*, pages 215–223, 2011.

[12] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon. Information-theoretic metric learning. In *International Conference on Machine learning*, pages 209–216, 2007.

[13] C. Ding and D. Tao. A comprehensive survey on pose-invariant face recognition. *arXiv preprint arXiv:1502.04383*, 2015.

[14] P. Grother and M. Ngan. Face recognition vendor test(frvt): Performance of face identification algorithms. *NIST Interagency Report 8009*, 2014.

[15] M. Guillaumin, J. Verbeek, and C. Schmid. Is that you? metric learning approaches for face identification. In *IEEE International Conference on Computer Vision*, pages 498–505, 2009.

[16] T. Hassner, S. Harel, E. Paz, and R. Enbar. Effective face frontalization in unconstrained images. *arXiv preprint arXiv:1411.7964*, 2014.

[17] 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 *Workshop on Faces in Real-Life Images: Detection, Alignment, and Recognition*, 2008.

[18] B. F. Klare, B. Klein, E. Taborsky, A. Blanton, J. Cheney, K. Allen, P. Grother, A. Mah, M. Burge, and A. K. Jain. Pushing the frontiers of unconstrained face detection and recognition: IARPA Janus Benchmark A. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[19] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar. Attribute and simile classifiers for face verification. In *IEEE International Conference on Computer Vision*, pages 365–372, 2009.

[20] H. X. Li, G. Hua, Z. Lin, J. Brandt, and J. C. Yang. Probabilistic elastic matching for pose variant face verification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3499–3506, 2013.

[21] 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.

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

[23] K. Simonyan, O. M. Parkhi, A. Vedaldi, and A. Zisserman. Fisher vector faces in the wild. In *British Machine Vision Conference*, volume 1, page 7, 2013.

[24] Y. Taigman, L. Wolf, and T. Hassner. Multiple one-shots for utilizing class label information. In *British Machine Vision Conference*, pages 1–12, 2009.

[25] X. Y. Tan and B. Triggs. Enhanced local texture feature sets for face recognition under difficult lighting conditions. *IEEE Transactions on Image Processing*, 19(6):1635–1650, 2010.

[26] P. Viola and M. J. Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004.

[27] L. Wolf, T. Hassner, and I. Maoz. Face recognition in unconstrained videos with matched background similarity. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[28] L. Wolf, T. Hassner, and Y. Taigman. Similarity scores based on background samples. In *Asian Conference on Computer Vision*, pages 88–97. 2010.

[29] L. Wolf, T. Hassner, and Y. Taigman. Effective unconstrained face recognition by combining multiple descriptors and learned background statistics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(10):1978–1990, 2011.

[30] S. Xie, S. G. Shan, X. L. Chen, and J. Chen. Fusing local patterns of Gabor magnitude and phase for face recognition. *IEEE Transactions on Image Processing*, 19(5):1349–1361, 2010.

[31] 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*, 16(1):57–68, 2007.

[32] W. Y. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM Computing Surveys*, 35(4):399–458, 2003.