Information Fusion 32 (2016) 80–92

Contents lists available at ScienceDirect

Information Fusion

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

An ensemble of patch-based subspaces for makeup-robust face recognition

Cunjian Chen [a], Antitza Dantcheva [b], Arun Ross [c,*]

[a] Department of Computer Science and Electrical Engineering, West Virginia University, Morgantown, WV 26506, USA
[b] INRIA, Team STARS, France
[c] Department of Computer Science and Engineering, Michigan State University, East Lansing, MI 48824, USA

ARTICLE INFO

Article history:
Available online 10 November 2015

Keywords:
Face recognition
Makeup
Cosmetics
Ensemble learning
Random subspace

ABSTRACT

Recent research has demonstrated the negative impact of makeup on automated face recognition. In this work, we introduce a patch-based *ensemble* learning method, which uses multiple subspaces generated by sampling patches from before-makeup and after-makeup face images, to address this problem. In the proposed scheme, each face image is tessellated into patches and each patch is represented by a set of feature descriptors, viz., Local Gradient Gabor Pattern (LGGP), Histogram of Gabor Ordinal Ratio Measures (HGORM) and Densely Sampled Local Binary Pattern (DS-LBP). Then, an improved Random Subspace Linear Discriminant Analysis (SRS-LDA) method is used to perform ensemble learning by sampling patches and constructing multiple common subspaces between before-makeup and after-makeup facial images. Finally, Collaborative-based and Sparse-based Representation Classifiers are used to compare feature vectors in this subspace and the resulting scores are combined via the sum-rule. The proposed face matching algorithm is evaluated on the YMU makeup dataset and is shown to achieve very good results. It outperforms other methods designed specifically for the makeup problem.

1. Introduction

Automated face recognition has been adopted in a broad range of applications such as personal authentication, video surveillance, image tagging, and human–computer interaction [1]. Automated face recognition systems recognize an individual by extracting a discriminative set of features from an input face image and comparing this feature set with a template stored in a database [1]. The recognition accuracy of these systems has rapidly improved over the past decade primarily due to the development of robust feature representations and matching techniques [2–5], as evidenced by significant reduction in error rates on several public benchmark databases (e.g., FRGC [6], LFW [7], YTF [8]). However, a number of challenges still remain particularly in *Heterogeneous Face Recognition* where the images to be matched are fundamentally different, e.g., visible versus thermal face images or face sketches versus photographs.

More recent research has investigated the problem of matching faces that have been altered either by plastic surgery [9] or by the application of facial cosmetics (i.e., makeup). In this work, we focus on the problem of matching face images before and after the application of makeup. These images are not acquired in a controlled environment, and hence considered as makeup in the wild. This problem is especially significant since makeup is a commonly used modifier of facial appearance. Thus, researchers in biometrics and cognitive psychology [10] are interested in understanding the effect of this modifier on face recognition.

1.1. Makeup challenge

Recent studies have demonstrated that makeup can significantly degrade face matching accuracy [11–13]. Makeup is typically used to enhance or alter the appearance of an individual's face. It has become a daily necessity for many, as reported in a recent British survey,[1] and as evidenced by a sale volume of 3.6 Billion in 2011 in the United States.[2] The cosmetic industry has developed a number of products, which can be broadly categorized as skin, eye or lip makeup. Skin makeup is utilized to alter skin color and texture, suppress wrinkles, and cover blemishes and aging spots. Lip makeup is commonly used to accentuate the lips (by altering contrast and the perceived shape) and to restore moisture. Eye makeup is widely used to increase the contrast in the periocular region, change the shape of the eyes, and

* Corresponding author.
E-mail address: rossarun@cse.msu.edu, cunjian@msu.edu (A. Ross).

http://dx.doi.org/10.1016/j.inffus.2015.09.005
1566-2535/

[1] http://www.superdrug.com/content/ebiz/superdrug/stry/cgq1300799243/surveyrelease-jp.pdf.
[2] https://www.npd.com/wps/portal/npd/us/news/press-releases/pr_120301/.

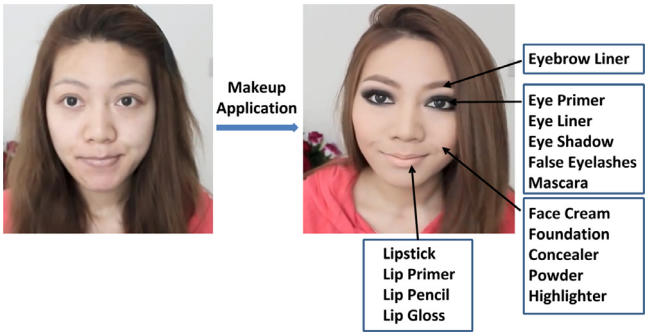


Fig. 1. An example showing how makeup can easily change the overall facial appearance, resulting in possible false non-match errors. Images are obtained from Youtube.

accentuate the eye-brows [14]. An example demonstrating the impact of applying makeup can be seen in Fig. 1.

Makeup poses a *challenge to automated face recognition* due to its potential to substantially alter the facial appearance. For example, it changes the perceived facial shape and appearance, modifies contrast levels in the mouth and eye region, and alters skin texture (see Fig. 1). Such modifications can lead to large intraclass variations, resulting in false non-matches, where a subject's face is not successfully recognized. Recent work by Dantcheva et al. [11] suggested that the recognition accuracy of both commercial and academic face recognition methods can be reduced by upto 76.21% due to the application of makeup.[3] It was concluded that non-permanent facial cosmetics can dramatically change facial appearance, both locally and globally, by altering color, contrast and texture. Existing face matchers, which rely on the cues of contrast and texture information for establishing a match, can be impacted by the application of facial makeup. It was also observed that the impact due to the application of eye makeup is considered to be more pronounced than lipstick makeup in our previous work [11]. The combination of eye and lipstick makeups poses a greater challenge than individual ones. Solutions to address this challenge are important towards developing robust face recognition systems.

1.2. Motivation and related work

To date, there is limited scientific literature on addressing the challenge of make-up induced changes. Chen et al. [15] presented an automated *makeup detection* approach, that was used to adaptively modify images prior to performing face recognition. Hu et al. [16] used canonical correlation analysis (CCA) along with a support vector machine (SVM) classifier to facilitate the matching of before-makeup and after-makeup images. Guo et al. [12] learned the mapping between features extracted from patches in the before- and after-makeup images in order to minimize the disparity between the images to be matched. The mapping was learned using CCA, rCCA (regularized CCA) and Partial Least Squares (PLS) methods. While *mapping-based methods* have been shown to be effective, they have two main limitations. First, the mapping between before-makeup and after-makeup facial images can be complex, spatially variant and nonlinear. Therefore, it is insufficient to learn a single mapping in order to describe the complex relationship between before-makeup and after-makeup samples [17]. Second, CCA and PLS methods have a tendency to overfit the training data and thus do not generalize well on unseen subjects [18].

In order to overcome these problems, we propose to use an *ensemble learning scheme [19,20] to generate multiple common semi-random subspaces for before-makeup and after-makeup samples*, instead of two separate subspaces. In random subspace methods, a set of multiple low-dimensional subspaces are generated by randomly sampling feature vectors in the original high-dimensional space [21]. It has proven to be effective in various tasks of face recognition [21–24]. For instance, Wang and Tang [21] proposed the use of Random Subspace Linear Discriminant Analysis (RS-LDA) for face recognition by randomly sampling eigenfaces. Zhu et al. [22] randomly sampled features on local image regions to construct a set of base classifiers. RS-LDA method [23,24] was also adapted for matching near-infrared images against visible images. The motivation for using a random subspace method are as follows [25]: (a) a learning algorithm can be viewed as searching for the best classifier in a space populated by different weak classifiers; (b) many weak classifiers are considered to be equally favorable when given a finite amount of training data; (c) averaging these individual classifiers can better approximate the true classifier. Therefore, a random subspace method can be used to generate multiple common subspaces, where each subspace contains a small portion of discriminative information pertaining to the identity. At the same time, by randomly selecting different patches as the input to each subspace-based classifier, the overfitting issue is avoided [21].

2. Proposed method

In this work, a patch-based ensemble learning scheme for face recognition in the presence of makeup is proposed (see Fig. 2). Given a face image, the proposed method first tessellates the image into patches and then applies multiple feature descriptors to each patch based on Local Gradient Gabor Pattern (LGGP) [26], Histogram of Gabor Ordinal Ratio Measures (HGORM) and Densely Sampled Local Binary Pattern (DS-LBP). These descriptors capture both global (LGGP and HGORM) and local (DS-LBP) information. Next, a weight learning scheme based on Fisher's separation criteria [27] is utilized to rank the significance of each patch. Then, a semi-random sampling method, based on the weights associated with the patches, is used to select multiple sets of patches and construct subspaces. This process is repeated for each of the three descriptors. Finally, Collaborative-based Representation Classifiers (CRC) and Sparse-based Representation Classifiers (SRC) are utilized in these subspaces resulting in an ensemble of classifiers for each descriptor. The scores generated by the classifiers are then fused using the sum-rule, which takes the weighted average of scores from multiple modalities [28]. The proposed method involves two levels of information fusion: the fusion of subspace classifiers corresponding to individual descriptors, and the fusion of matching scores generated by all descriptors. The rationale for the proposed method are as follows: (1) a single descriptor is not sufficient enough to describe a face image; (2) the prior knowledge about which patch is impacted by makeup is unknown; (3) semi-random sampling can increase the probability of selecting patches that are not impacted by makeup. The proposed framework is illustrated in Fig. 2. As can be seen from this figure, the sum-rule is used for first fusing the scores corresponding to the multiple subspaces and then the scores corresponding to the three descriptors.

Our approach to match two face images of the same person, acquired before and after the application of makeup, differs from previously published works with RS-LDA [21–24]. In the work of [21] and [22], the RS-LDA method was not used to handle heterogeneous face recognition. In [23] and [24], the patch sampling procedure is performed *across* different feature descriptors (SIFT and LBP), while our patch sampling is performed *within* the same feature descriptor.

Contributions:

1. We propose an ensemble framework for a face matcher that is robust to the application of makeup. The approach utilizes multiple subspaces corresponding to three different feature descriptors

[3] http://www.antitza.com/makeup-datasets-benchmark.html.

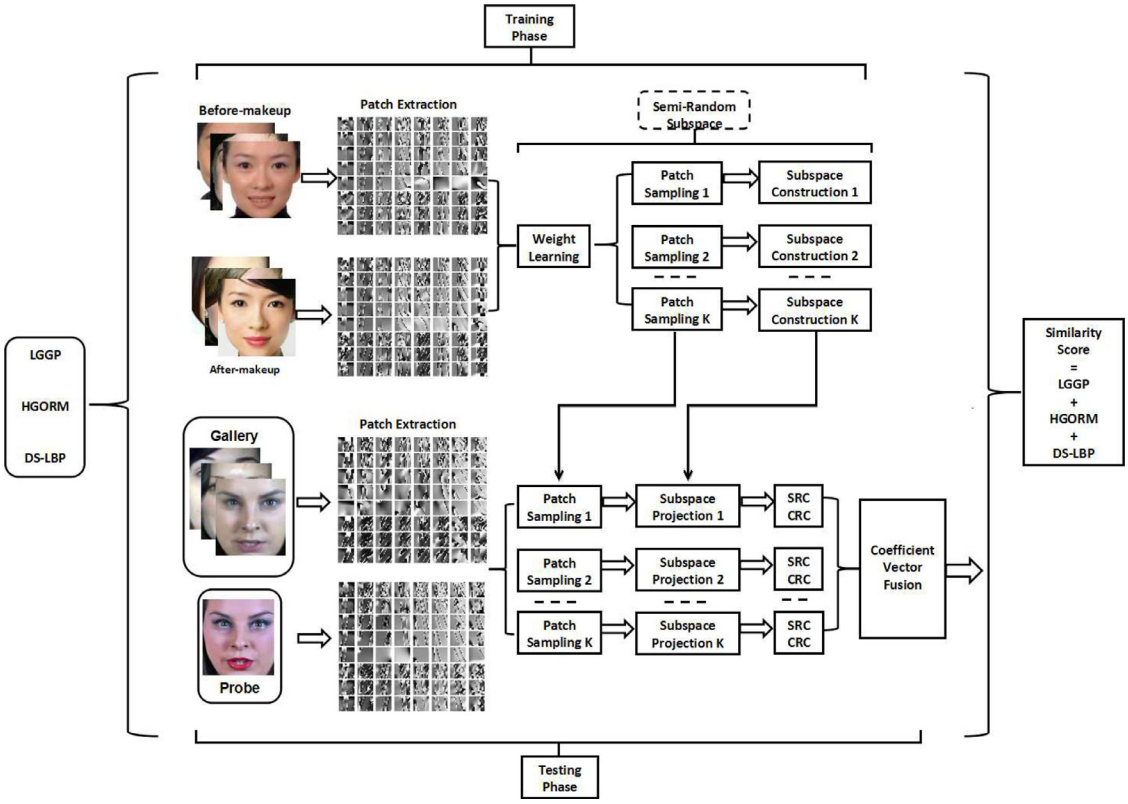


Fig. 2. Proposed framework for matching after-makeup images with before-makeup images. During the training phase, for each feature descriptor, a pool of patches is extracted, followed by weight learning, patch sampling and random subspace construction. In the testing phase, patches from an input image are projected onto the learned random subspace. A combination of SRC and CRC classifiers are used to compare feature vectors in these subspaces and generate a match score. This process is repeated for each descriptor and the matching scores corresponding to individual feature descriptors are fused to generate the final similarity score.

and multiple image patches. A combination of sparse and collaborative classifiers is used in these subspaces.

2. We propose a sampling scheme, which utilizes weight information from each patch to guide patch selection, instead of pure random sampling.

3. Feature descriptors

The design of effective face image descriptors is considered to be crucial in face recognition. In this work, the patches in each face image are represented using LGGP, HGORM and DS-LBP descriptors. Compared to Dense HOG and Dense LBP used in [23], LGGP and HGORM are derived from Gabor-filtered images and have demonstrated to be more discriminative based on empirical investigation [26]. Let $I \in \mathbb{R}^{128\times128}$ denote a face image that is either a before-makeup sample or an after-makeup sample, and let f denote a feature extractor. Moreover, z is a generic pixel position, x_c is the center of the rectangle used for coding, which is shifted both horizontally and vertically along the image. γ_c represents the intensity value in x_c. We now describe the three basic feature descriptors used in the framework.

Gabor filters: A Gabor filter can be mathematically defined as follows [27]:

$$\varphi_{\mu,\upsilon}(z) = \frac{||k_{\mu,\upsilon}||^2}{\sigma^2} e^{-\frac{||k_{\mu,\upsilon}||^2||z||^2}{2\sigma^2}} [e^{ik_{\mu,\upsilon}z} - e^{-\frac{\sigma^2}{2}}], \quad (1)$$

where μ and υ denote the orientation and scale of the Gabor filters, respectively. z denotes the pixel position and $\|\cdot\|$ denotes the norm operator [27]. σ is a constant, which has a default value of 2π. The wave vector $k_{\mu,\upsilon}$ is given by $k_{\mu,\upsilon} = k_\upsilon e^{i\phi_\mu}$, where $k_\upsilon = k_{max}/s^\upsilon$ and $\phi_\mu = \pi\mu/8$. Here, k_{max} is the maximum frequency and s is the spacing factor between kernels in the frequency domain. The Gabor response of an image is obtained by performing the convolution of the input image with Gabor kernels: $G_{\mu,\upsilon}(z) = I(z) * \varphi_{\mu,\upsilon}(z)$. The complex Gabor response has two parts: the real part $\Re_{\mu,\upsilon}(z)$ and the imaginary part $\Im_{\mu,\upsilon}(z)$. Accordingly, the Gabor magnitude $A_{\mu,\upsilon}(z)$ and phase $\theta_{\mu,\upsilon}(z)$ can be computed as:

$$A_{\mu,\upsilon}(z) = \sqrt{\Re_{\mu,\upsilon}(z)^2 + \Im_{\mu,\upsilon}(z)^2}, \quad (2)$$

and

$$\theta_{\mu,\upsilon}(z) = \arctan(\Im_{\mu,\upsilon}(z)/\Re_{\mu,\upsilon}(z)). \quad (3)$$

Both Gabor magnitude and phase responses have been proven to be useful in face recognition [27,29]. Since Gabor responses contain highly correlated and redundant information, it is essential to further encode such responses. It has been suggested that there are four different types of measurements from coarse to fine: nominal, ordinal, interval, and ratio measures [30]. A nominal measure uses numerals to denote objects for the purpose of identification. An ordinal measure uses rank-ordering to sort objects. An interval measure denotes the degree of difference between objects. A ratio measure estimates the ratio between two numerical values [31]. The ordinal and ratio measures were chosen, since they have been successfully used in biometrics. For instance, ordinal measure has been used for comparing Gabor features in iris recognition [30] and ratio measure has been used as a local image descriptor [32]. Therefore, these measures can be used to capture local variations in a face image. To code Gabor responses, we utilize both ordinal and ratio measures to develop robust feature descriptors. The process of encoding Gabor-filtered images involves the following steps: (1) apply multi-orientation (eight orientations) and multi-scale (five scales) Gabor filters on the input face image; (2) derive either ratio or ordinal measures from the magnitude and phase components of the resulting images; (3) extract statistical distributions of these measures based on local histograms.

3.1. Local Gabor Gradient Pattern (LGGP)

To code Gabor *magnitude* responses, we use a gradient descriptor defined as [26]:

$$\xi(x_c) = \arctan\left(\beta \cdot \frac{N_v}{N_h + \lambda}\right), \tag{4}$$

where N_v and N_h are the image gradients to be computed along vertical and horizontal directions, respectively. Here, the two directions are orthogonal to each other. The arctangent function (arctan), along with parameters β and λ, is used to prevent the output from increasing or decreasing too quickly [26]. Let γ_c define the intensity value of center pixel in a rectangle surrounded by neighbors equally sampled from x_0 to x_{R-1}, where R is the neighborhood size. The gradients can now be computed as:

$$N_v = \gamma_{mod(i+4,R)} - \gamma_i, \tag{5}$$

$$N_h = \gamma_{mod(i+6,R)} - \gamma_{mod(i+2,R)}. \tag{6}$$

Here, the modulo operator is denoted by *mod* and i is the index for the neighborhood pixel. In our implementation, we use $R = 8$, $\beta = 3$ and $\lambda = 1 \times 10^{-7}$. To generate LGGP features, each gradient-encoded Gabor image is divided into *non-overlapping* patches, and histogram information is extracted from each patch. The number of patches in each image is 64, where each patch size is 16 × 16. The number of histogram bins is 16. LGGP feature extraction is denoted as $f_G(I) = \{\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_P\}$, where P is the number of total patches and $\mathbf{g}_p \in \mathbb{R}^{16}$.

3.2. Histogram of Gabor Ordinal Ratio Measures (HGORM)

To code Gabor *phase* responses, we use the Ordinal Measure (OM) [30,33]. OM compares two different regions to determine which one has a larger value (e.g., mean). For instance, if region A has a larger value than region B, then the resulting code is 1, otherwise it is 0. Such a measure is used to encode the qualitative relationship between different regions. The advantages of using ordinal measures for image representation have been established in palmprint recognition [33], iris recognition [30] and face recognition [4]. To perform ordinal feature extraction, one simple approach is to compute the weighted average of dissociated image regions. This can be accomplished by the process of ordinal filtering. An ordinal filter consists of multiple positive and negative lobes, as illustrated in Fig. 3. Here, we use multi-lobe differential filters (MLDF) [4] to extract ordinal features. The positive and negative lobes are represented by Gaussian filters. MLDF can be mathematically expressed as,

$$MLDF = C_p \sum_{i=1}^{N_p} \frac{1}{\sqrt{2\pi}\,\delta_{pi}} exp\left[\frac{-(z-\mu_{pi})^2}{2\delta_{pi}^2}\right] - C_n \sum_{j=1}^{N_n} \frac{1}{\sqrt{2\pi}\,\delta_{nj}} exp\left[\frac{-(z-\mu_{nj})^2}{2\delta_{nj}^2}\right], \tag{7}$$

where z denotes the pixel position, μ and δ denote the central position and the scale of a 2D Gaussian filter, respectively. N_p is the number of positive lobes and N_n is the number of negative lobes. C_p and C_n are the constant coefficients, used to ensure that the output of MLDF is zero, i.e., $C_p N_p = C_n N_n$. MLDF is a type of differential bandpass filter. It is flexible in terms of types of lobes, spatial configuration of lobes, and number of lobes. An example of MLDF filters is shown in Fig. 3.

Unlike the work of Chai et al. [4], we also consider the ratio measure in addition to the ordinal measure. First, we construct a horizontal di-lobe ordinal filter and a vertical di-lobe ordinal filter. Then, we perform the ordinal filtering on the output of Gabor phase responses. Finally, a ratio measure is used to compute the final representation, which is denoted as OF:

$$OF = \arctan\left(\beta \cdot \frac{O_v}{O_h + \lambda}\right), \tag{8}$$

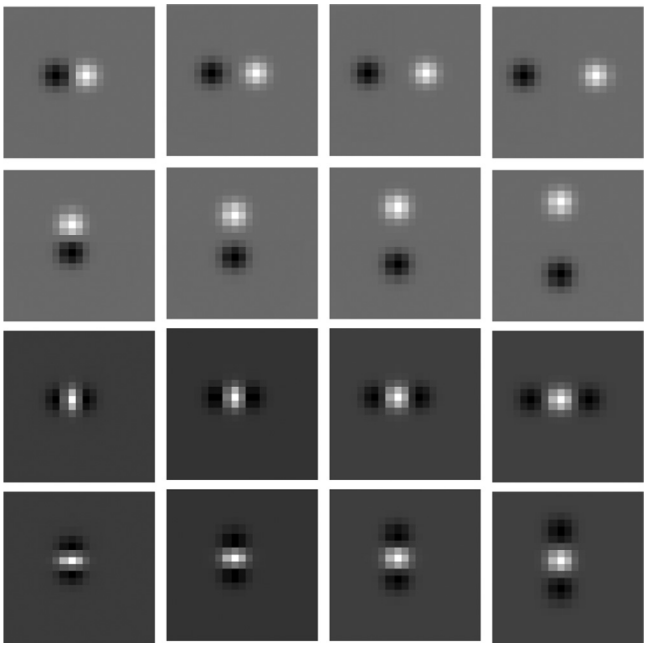


Fig. 3. Illustration of ordinal filters at different distances and orientations (horizontal, vertical). Positive and negative lobes are arranged in various configurations in terms of number of lobes and orientations between lobes.

where O_h is the convolution of horizontal di-lobe ordinal filter with the Gabor phase response and O_v is the convolution of vertical di-lobe ordinal filter with the Gabor phase response. β and λ are constant values used to stabilize the function. The domain of Eq. (8) is $[-\infty, +\infty]$. The proposed feature representation is called Histogram of Gabor Ordinal Ratio Measures (HGORM). We use the following parameters in our work: the size of the ordinal filter is 21, the distance between positive and negative lobes is 3 pixels, $\delta_{pi} = \delta_{nj} = \pi/2$, $N_p = N_n = 1$, $\beta = 3$ and $\lambda = 1 \times 10^{-7}$.

HGORM can be considered as an extension of the LGGP descriptor that we previously developed [26]. The ratio measure used in HGORM is weighted by a Gaussian kernel, thereby making it more robust to noise. To generate HGORM features, each OM-encoded Gabor image is divided into *non-overlapping* patches, where histogram information is extracted from each patch. The number of patches in each image is 64, where each patch size is 16 × 16. The number of histogram bins is 16. HGORM feature extraction is denoted as $f_O(I) = \{\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_P\}$, where P is the number of total patches and $\mathbf{o}_p \in \mathbb{R}^{16}$.

3.3. Densely Sampled Local Binary Pattern (DS-LBP)

The LBP texture descriptor [27] has been proven to be effective in capturing micro-patterns or micro-structures in the face image. It is calculated by binarizing local neighborhoods, based on the differences in pixel intensity between the center pixel and neighborhood pixels, and converting the resulting binary string into a decimal value. In this work, uniform LBP patterns are extracted from the original image, resulting in a 59-bin histogram feature vector. Uniform LBP patterns refer to those binary patterns that have at most two bitwise transitions from 0 to 1 or 1 to 0. Uniformity is an important characteristic, as it reflects micro-features such as lines, edges and corners, which are enhanced by the application of makeup. To generate DS-LBP features, each LBP coded image is divided into *overlapping* patches, and histogram information is extracted from each patch. The number of patches in each image is 256, where each patch size is 16 × 16. The number of histogram bins is 59. DS-LBP feature extraction is denoted as $f_S(I) = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_P\}$, where P is the number of total

Table 1
Summary of patch-based feature descriptors used in the proposed framework.

Feature types	LGGP	HGORM	DS-LBP
Patch size	$16 * 16$	$16 * 16$	$16 * 16$
Patch tessellation	*Non-overlapping*	*Non-overlapping*	*Overlapping*
Number of patches	2560	2560	256
Number of bins per patch	16	16	59

patches and $\mathbf{s}_p \in \mathbb{R}^{59}$. All three feature descriptors are summarized in Table 1.

3.4. Complement of feature descriptors

The choice of feature descriptors was based on the consideration that they are complementary to each other. This was established based on several empirical evaluations. First, the performance of individual descriptors does not deviate significantly from each other. Second, the subsequent experimental results on makeup dataset suggest that the combination of individual descriptors results in the best performance. Third, the performance of these descriptors and their combinations was empirically investigated on the HFB database [26], whose characteristic is different from the makeup dataset.

4. Semi-random subspace LDA (SRS-LDA)

In this section, we describe the details of the proposed SRS-LDA method for matching after-makeup images to before-makeup images. Let $I^B = \{I_i^B\}_{i=1}^c$ contain c classes of before-makeup samples, with each class $I_i^B = \{I_{i,j}^B\}_{j=1}^{n_i}$ consisting of n_i samples $I_{i,j}^B$ resulting in a total of $N = \sum_{i=1}^c n_i$ before-makeup samples in the set. Similarly, after-makeup samples consisting of c classes are denoted as $I^A = \{I_i^A\}_{i=1}^c$, where $I_i^A = \{I_{i,j}^A\}_{j=1}^{m_i}$ and $M = \sum_{i=1}^c m_i$.

In order to extract features corresponding to each descriptor, an image is divided into P patches (see Table 1). Therefore, a set of P feature vectors are extracted from a given image. Let the set of feature vectors extracted from a before-makeup image be denoted as $\mathbf{X}_{i,j}^B = f(I_{i,j}^B)$. The p-th feature vector from $\mathbf{X}_{i,j}^B$ is denoted as $\mathbf{X}_{i,j}^B(p)$, where $\mathbf{X}_{i,j}^B(p) \in \mathbb{R}^{16}$ for LGGP and HGORM features, and $\mathbf{X}_{i,j}^B(p) \in \mathbb{R}^{59}$ for DS-LBP features. Similarly, $\mathbf{X}_{i,j}^A = f(I_{i,j}^A)$ is used to denote the set of feature vectors from an after-makeup image. Without loss of generality, $\mathbf{X}_{i,j}$ is used to denote the set of feature vectors extracted from $I_{i,j}$, be it a before-makeup or an after-makeup image. For each of the three descriptors (LGGP, HGORM, and DS-LBP), the following procedure is adopted: training phase and testing phase (see Fig. 2). The resultant matching scores are then fused from the three descriptors to make a final recognition decision.

4.1. Training phase

Weight learning: Before sampling patches, we assign a weight to each extracted patch and then rank the patches based on these weights. The weights are computed based on Fisher's separation criterion [27]. Our assumption is that different facial regions may have different impact on face recognition across makeup, because makeup information is not uniformly distributed. For each patch p and its associated feature vector $\mathbf{X}_{i,j}(p)$, the mean of intra-class distance can be computed as:

$$m_1(p) = \frac{1}{c}\sum_{i=1}^{c}\frac{2}{(l_i-1)l_i}\sum_{j=1}^{l_i-1}\sum_{k=j+1}^{l_i}\phi(\mathbf{X}_{i,j}(p),\mathbf{X}_{i,k}(p)) \tag{9}$$

where, ϕ denotes the chi-squared distance between two feature vectors, $l_i = n_i + m_i$ denotes the number of samples per class. The variance of intra-class distance can be computed as:

$$var_1(p) = \sum_{i=1}^{c}\sum_{j=1}^{l_i-1}\sum_{k=j+1}^{l_i}(\phi(\mathbf{X}_{i,j}(p),\mathbf{X}_{i,k}(p)) - m_1(p))^2. \tag{10}$$

The mean of inter-class distance can be computed as:

$$m_2(p) = \frac{2}{c(c-1)}\sum_{i=1}^{c-1}\sum_{q=i+1}^{c}\frac{1}{l_i l_q}\sum_{j=1}^{l_i}\sum_{k=1}^{l_q}\phi(\mathbf{X}_{i,j}(p),\mathbf{X}_{i,k}(p)). \tag{11}$$

The variance of inter-class distance can be computed as:

$$var_2(p) = \sum_{i=1}^{c-1}\sum_{q=i+1}^{c}\sum_{j=1}^{l_i}\sum_{k=1}^{l_q}(\phi(\mathbf{X}_{i,j}(p),\mathbf{X}_{i,k}(p)) - m_2(p))^2. \tag{12}$$

Then the weight of each patch p is calculated as,

$$w(p) = \frac{(m_1(p)-m_2(p))^2}{var_1(p)+var_2(p)} \tag{13}$$

where, $(m_1(p)-m_2(p))^2$ is the difference between intra-class distance and inter-class distance. It is a measure of the discriminability of patch p, i.e., its ability to separate classes. $w(p)$ is a non-negative value and is the weight associated with the patch. In other words, the learned weight is an indication of the importance of different patches for recognition. The patches are then sorted in descending order of their weights. This is used to guide the subsequent patch sampling process for each feature descriptor.

Patch sampling: As stated earlier, multiple subspaces (K) are used to generate the ensemble of classifiers corresponding to each descriptor. Each subspace is constructed based on the semi-random sampling of the weighted patches. Here, the term semi-random is used to indicate that the probability of selecting a patch is related to its weight. For creating the k-th subspace, $k = \{1, 2, \ldots K\}$, we sample α number of patches (without replacement) pertaining to a specific descriptor from $\mathbf{X}_{i,j}^B$ and $\mathbf{X}_{i,j}^A$, where $i = \{1, \ldots, c\}$, $j = \{1, \ldots, n_i\}$, and $\alpha \in \{1, \ldots, P\}$, to obtain $\mathbf{x}_{i,j}^B \in \mathbb{R}^{D\times 1}$ and $\mathbf{x}_{i,j}^A \in \mathbb{R}^{D\times 1}$. Here, $\mathbf{x}_{i,j}^B$ and $\mathbf{x}_{i,j}^A$ are obtained by concatenating all the α feature vectors into a single vector representation, where $D = \alpha \times d$ and d is the feature dimension for each patch (see Table 1). Overall, 60% of α are selected from the first half of P and 40% of α are selected from the remaining half. This ensures that more important patches have a higher chance to be selected. The choice of K and α is empirically determined, considering a low computational complexity. The precise values are specified in Section 6.1.

Subspace construction: Since the dimensionality of D is usually much higher than the number of samples, a feature dimension reduction is performed to reduce computational time as well as to avoid the small sample size problem [21]. A common way to reduce the feature dimension is by applying Principal Component Analysis (PCA). PCA seeks to find the projection space, which can best reconstruct original vectors. To find such a subspace, mean vectors $\boldsymbol{\mu}^B$ and $\boldsymbol{\mu}^A$ are computed from before-makeup and after-makeup sampled feature vectors, respectively: $\boldsymbol{\mu}^B = \frac{1}{N}\sum_{i=1}^c\sum_{j=1}^{n_i}\mathbf{x}_{i,j}^B$, and $\boldsymbol{\mu}^A = \frac{1}{M}\sum_{i=1}^c\sum_{j=1}^{m_i}\mathbf{x}_{i,j}^A$. An overall mean vector can be computed as $\boldsymbol{\mu} = \frac{1}{N+M}\sum_{i=1}^c\sum_{j=1}^{n_i+m_i}\mathbf{x}_{i,j}$. The entire covariance matrix can be computed as:

$$\mathbf{S} = \frac{1}{N+M}\sum_{i=1}^{c}\sum_{j=1}^{n_i+m_i}(\mathbf{x}_{i,j}-\boldsymbol{\mu})(\mathbf{x}_{i,j}-\boldsymbol{\mu})^T. \tag{14}$$

We can now compute eigenvectors $\boldsymbol{W}_E$ from the covariance matrix $\boldsymbol{S}$: $\boldsymbol{S}\boldsymbol{W}_E = \lambda\boldsymbol{W}_E$. After generating $\boldsymbol{W}_E$, the before-makeup and after-makeup samples can be projected into the new subspace as:

$$\mathbf{y}_{i,j}^B = \boldsymbol{W}_E^T(\mathbf{x}_{i,j}^B - \boldsymbol{\mu}^B), \tag{15}$$

and

$$\mathbf{y}_{i,j}^A = \boldsymbol{W}_E^T(\mathbf{x}_{i,j}^A - \boldsymbol{\mu}^A). \tag{16}$$

$\mathbf{y}^B_{i,j}$ and $\mathbf{y}^A_{i,j}$ are the projected feature vectors after PCA for before-makeup and after-makeup samples, respectively. The number of eigenvectors used is $\min(N-c, M-c)$.

We use both before-makeup and after-makeup feature vectors to compute the between-class scatter and within-class scatter matrices. This ensures that the learned feature representation is less sensitive to makeup changes. The mean class vector for i-th subject when constructing the k-th subspace for a feature descriptor is calculated using both before-makeup ($\mathbf{y}^B_{i,j}$) and after-makeup ($\mathbf{y}^A_{i,j}$) projected vectors:

$$\boldsymbol{\mu}_i^{(k)} = \frac{1}{n_i + m_i}\left(\sum_{j=1}^{n_i}\mathbf{y}^B_{i,j} + \sum_{j=1}^{m_i}\mathbf{y}^A_{i,j}\right). \tag{17}$$

Then the between-class scatter matrix can be computed as,

$$\boldsymbol{S}_B^{(k)} = \sum_{i=1}^{c}(\boldsymbol{\mu}_i^{(k)} - \boldsymbol{\mu}^{(k)})(\boldsymbol{\mu}_i^{(k)} - \boldsymbol{\mu}^{(k)})^T. \tag{18}$$

where $\boldsymbol{\mu}^{(k)} = \frac{1}{c}\sum_{i=1}^{c}\boldsymbol{\mu}_i^{(k)}$. The within-class scatter matrix can be computed as,

$$\begin{aligned}\boldsymbol{S}_W^{(k)} = &\sum_{i=1}^{c}\sum_{j=1}^{n_i}\left(\mathbf{y}^B_{i,j} - \boldsymbol{\mu}_i^{(k)}\right)\left(\mathbf{y}^B_{i,j} - \boldsymbol{\mu}_i^{(k)}\right)^T \\ &+ \sum_{i=1}^{c}\sum_{j=1}^{m_i}\left(\mathbf{y}^A_{i,j} - \boldsymbol{\mu}_i^{(k)}\right)\left(\mathbf{y}^A_{i,j} - \boldsymbol{\mu}_i^{(k)}\right)^T.\end{aligned} \tag{19}$$

The objective of LDA is to seek the optimal projection $\boldsymbol{W}_F$, which can maximize the ratio between determinant of the between-class scatter matrix and determinant of the within-class scatter matrix. The optimization problem is defined as,

$$\boldsymbol{W}_F^{(k)} = \arg\max_{\mathbf{W}_F}\frac{\boldsymbol{W}_F^T\boldsymbol{S}_B^{(k)}\boldsymbol{W}_F}{\boldsymbol{W}_F^T\boldsymbol{S}_W^{(k)}\boldsymbol{W}_F}. \tag{20}$$

This is equivalent to solving the generalized eigenvalue problem of $\boldsymbol{S}_B^{(k)}\psi^{(k)} = \lambda^{(k)}\boldsymbol{S}_W^{(k)}\psi^{(k)}$, where $k = \{1, 2, \ldots, K\}$. The output of the training phase is $\boldsymbol{\mu}^B$, $\boldsymbol{\mu}^A$, $\boldsymbol{W}_F^{(k)}$ and $\boldsymbol{W}_E$ for each k (i.e., subspace). The potential importance of this proposed method lies in (a) using both before- and after-makeup patches together in the weight learning process and (b) generating the corresponding subspaces, where both before- and after-makeup feature vectors are used for learning the within- and between-class scatter matrices of LDA. These constructed subspaces, also referred to as common subspaces, are repeated for each descriptor.

4.2. Testing phase

In the testing phase, the after-makeup images of a subject in a sequestered test set are treated as probes and compared with before-makeup images that are treated as gallery images. Let $\mathbf{X}_{i,j} = f(I_{i,j})$ denote the set of feature vectors extracted from $I_{i,j}$, where $I_{i,j}$ is either a before-makeup image or an after-makeup image from test samples. The same set of α patches are selected from $\{\mathbf{X}_{i,j}(p) : \mathbf{X}_{i,j}(p) \in \mathbf{X}_{i,j}, 1 < p < P\}$, and concatenated to form a single feature vector $\mathbf{x}^B_{i,j}$ or $\mathbf{x}^A_{i,j}$. The location and order of patches in the training set and the test set are the same.

Subspace projection: For each derived subspace $k = \{1, 2, \ldots, K\}$, the representation for test samples of before-makeup and after-makeup are obtained as follows:

$$\mathbf{y}^B_{i,j} = \boldsymbol{W}_F^T\boldsymbol{W}_E^T\left(\mathbf{x}^B_{i,j} - \boldsymbol{\mu}^B\right), \tag{21}$$

and

$$\mathbf{y}^A_{i,j} = \boldsymbol{W}_F^T\boldsymbol{W}_E^T\left(\mathbf{x}^A_{i,j} - \boldsymbol{\mu}^A\right), \tag{22}$$

where $\mathbf{y}^B_{i,j}$ and $\mathbf{y}^A_{i,j}$ are the final projected feature vectors for before-makeup and after-makeup test samples, respectively. In case the makeup information is unknown, a makeup detection scheme [15] can be employed to make the distinction. An overall mean vector $\boldsymbol{\mu}$ can also be used for projection, resulting in a *slight* decrease in matching accuracy (< 1% verfication rate).

SRC and CRC classification: As stated earlier, the before-makeup samples are used as gallery, and the after-makeup samples are used as probe. Let $\mathbf{Y}^B = \{\mathbf{y}^B_{1,1}, \ldots, \mathbf{y}^B_{1,n'_1}, \ldots, \mathbf{y}^B_{i,1}, \ldots, \mathbf{y}^B_{i,n'_i}, \ldots, \mathbf{y}^B_{c',1}, \ldots, \mathbf{y}^B_{c',n'_c}\}$ denote the gallery feature vectors of before-makeup samples, where c' is the number of subjects in the gallery and n'_i is the number of samples for the i-th subject. Let $\mathbf{y}^A_{i,j}$ denote an after-makeup probe sample. Distance scores can be computed by matching a probe sample against a gallery, thereby obtaining a similarity score between before-makeup and after-makeup samples. This is accomplished by using the principles of *sparse* and *collaborative* representation. Wright et al. [42] demonstrated the robustness of sparse representation based classification (SRC) to occlusions and noise. The application of makeup can be viewed as the addition of noise, i.e., pixel corruption. Zhang et al. [34] demonstrated that it is the collaborative representation based classification (CRC), rather than just sparsity, that lends robustness to face recognition. Thus, the collaborative role of face images from other subjects in the classification process cannot be undermined. SRC and CRC have their own merits and can provide complementary information in classification [34]. Therefore, we develop two classifiers for each subspace – one based on SRC and the other on CRC – and combine their outputs. In this way, we exploit the complementary information provided by both of them.

The coefficient of projection of an after-makeup sample can be obtained using both ℓ_1 (SRC) and ℓ_2 (CRC) solutions:

$$\boldsymbol{\rho}_1^{(k)} = \arg\min_{\boldsymbol{\rho}}||\mathbf{Y}^B\boldsymbol{\rho} - \mathbf{y}^A_{i,j}||_2^2 + \lambda_1||\boldsymbol{\rho}||_1, \tag{23}$$

and

$$\boldsymbol{\rho}_2^{(k)} = ((\mathbf{Y}^B)^T\mathbf{Y}^B + \lambda_2\mathbf{I})^{-1}(\mathbf{Y}^B)^T\left(\mathbf{y}^A_{i,j}\right). \tag{24}$$

The ℓ_1 solution is obtained using Least Angle Regression (LARS) algorithm [35]. LARS algorithm is a variant for solving the Lasso based on model selection. The dimension of both $\boldsymbol{\rho}_1^{(k)}$ and $\boldsymbol{\rho}_2^{(k)}$ is $N' = \sum_{i=1}^{c'} n'_i$. The coefficient vector for the k-th subspace is $\boldsymbol{\rho}^{(k)} = \boldsymbol{\rho}_1^{(k)} + \boldsymbol{\rho}_2^{(k)}$. The coefficient vectors pertaining to all subspaces are summed together (see Fig. 4):

$$\boldsymbol{\rho} = \sum_{k=1}^{K}\boldsymbol{\rho}^{(k)}. \tag{25}$$

$\boldsymbol{\rho}$ is the final coefficient (score) vector generated when matching an after-makeup feature vector $\mathbf{y}^A_{i,j}$ (probe) to a set of feature vectors $\mathbf{Y}^B$ (gallery). The proposed framework fuses the output of three different feature descriptors, i.e., LGGP, HGORM, and DS-LBP. So there are three coefficient vectors: $\boldsymbol{\rho}_G$, $\boldsymbol{\rho}_O$ and $\boldsymbol{\rho}_S$. These are combined to get a single coefficient vector $\boldsymbol{\rho}_F = (\boldsymbol{\rho}_G + \boldsymbol{\rho}_O + \boldsymbol{\rho}_S)/3$. Here, same weight is used for different modalities during the sum-rule fusion. The match score between the r-th entry in the gallery and the probe image corresponds to the r-th element in $\boldsymbol{\rho}_F$. Fig. 4 illustrates this. Given a probe sample, the red circle denotes the computed coefficient (similarity value) associated with the first gallery sample in each subspace. In this example, the first gallery sample corresponds to the same identity as the probe sample. The rest of the coefficients are associated with other gallery samples. The identity of the probe sample is correctly inferred after summing all the coefficients in the red circle. Even though the first gallery sample is not associated with the largest coefficient value in individual subspaces, the sum of each of these coefficients results in the maximum value among all samples.

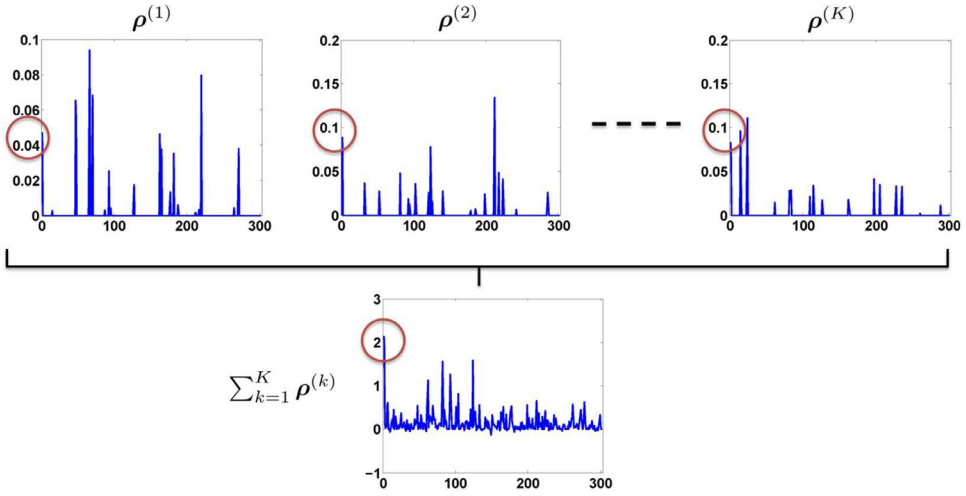


Fig. 4. An example showing how SRC is used for identity classification during ensemble learning. CRC follows the same principle, except that coefficients are more widely spread. $\rho^{(k)}$ denotes the coefficients generated for the k-th subspace of a particular descriptor. Horizontal axis refers to the number of gallery samples and vertical axis refers to the coefficients for each sample as computed in Eq. (23). The red circle denotes the coefficient associated with the first gallery sample in each subspace. The identity of a probe sample is correctly inferred after summing all the coefficients in the red circle. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

5. Baseline algorithms

The accuracy of the proposed SRS-LDA method is compared against several face recognition algorithms. They are used as baselines in our analysis.[4]

Commercial Off-The-Shelf (COTS) Systems: Three commercial face recognition systems were evaluated in this study. To anonymize results obtained by these commercial matchers, they are referred to as COTS-1, COTS-2, and COTS-3. There are several reasons why COTS are used in this work.

1. These are three of the 10 matchers that were evaluated in the NIST-organized Multi-Biometrics Evaluation [36].
2. These matchers are representative of state-of-the-art performance in face recognition and are the top performers in various face recognition evaluations [36,37].
3. Comparing the accuracy of the proposed method against leading COTS systems provides an unbiased and objective baseline [24,36].
4. COTS have further room for improvement and the proposed method is shown to provide complementary information.

OpenBR: OpenBR [38] is a publicly available toolkit for biometric recognition and evaluation. The default face recognition algorithm in OpenBR is developed based on the Spectrally Sampled Structural Subspaces Features (4SF) [38]. An input face image is represented by extracting histograms of local binary pattern (LBP) and scale-invariant feature transform (SIFT) features, computed on a dense grid of patches. The histograms from each patch are then projected onto a subspace generated using PCA, in order to obtain a feature vector. LDA is then applied on each random sample to learn the discriminative subspace. The efficacy of the OpenBR face matching algorithm is elaborated in [38].

Histogram of Monogenic Binary Pattern (HMBP): HMBP [39] is established by concatenating the histograms of monogenic binary pattern (MBP) from all subregions. MBP is computed based on the output from monogenic signal analysis. The magnitude, phase and orientation of a 2D image is derived and local histogram is built from each non-overlapping subregion. The number of bins is 512. The final feature dimension for HMBP is 153,600.

Partial Least Square (PLS): Partial least square (PLS) is a statistical learning technique originally proposed in the field of chemometrics [12] as an alternative to ordinary least square regression. It maps input vectors (regressors) and corresponding output vectors (responses) into a common feature space such that the covariance between the projected input and output vectors is maximized. The number of basis vectors is 70.

6. Experiments

The following experiments were designed with the primary goal of exploring the effectiveness of the SRS-LDA method in matching after-makeup to before-makeup face samples.

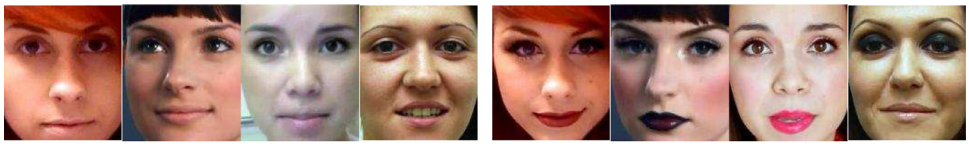
6.1. Makeup datasets

We utilized the YMU-dataset consisting of 151 subjects, specifically Caucasian females, from YouTube makeup tutorials.[5] Images of the subjects before and after the application of makeup were captured. There are four samples per subject: two samples before the application of makeup and two samples after the application of makeup. The makeup in these face images varies from subtle to heavy. The cosmetic alteration is mainly in the ocular area, where the eyes have been accentuated by diverse eye makeup products. Additional changes are on the quality of the skin, due to the application of foundation and change in lip color. The database is relatively unconstrained, exhibiting variations in facial expression, pose and resolution. Although there are other intra-class variations in this dataset, it was determined in [11] that the drop in matching accuracy was primarily due to the application of makeup. Some examples of YMU dataset are shown in Fig. 5. The face images are geometrically normalized using an affine transformation, based on the eye landmarks, in order to reduce variations due to scale and pose. All normalized face images are cropped and resized to a dimension of 128 × 128. Images were converted from RGB to grayscale [11].

In addition to the aforementioned dataset, we assembled another makeup dataset for the purpose of training the proposed face matcher. The training dataset consists of a subset of female subjects from the FAM database [16], a female Asian makeup dataset from

[4] We also tested the CCA method proposed in [12]. The performance due to CCA was worse than the other matchers and, hence, not included in this paper.

[5] Available at: http://www.antitza.com/makeup-datasets.html.



(a) Before-makeup samples (b) After-makeup samples

Fig. 5. The example images after alignment and cropping. Substantial change in facial appearance is observed after the application of makeup.

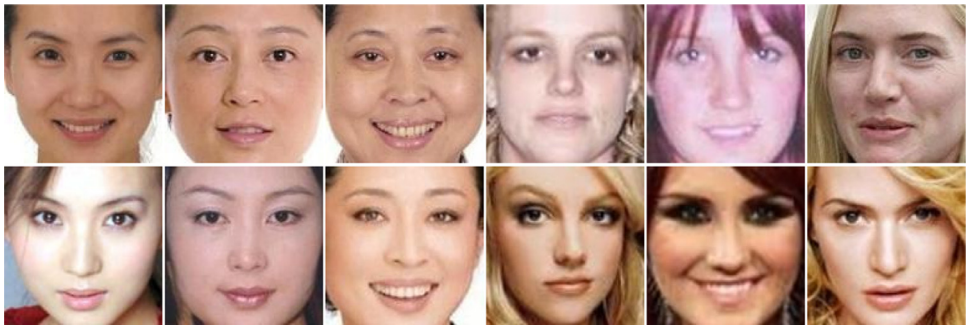


Fig. 6. Sample images from the T-makeup training dataset. Top row shows images before the application of makeup and bottom row shows images after the application of makeup.

Youtube, and the entire MIAA dataset [40] (see Fig. 6). The total number of samples in this training dataset is 796, corresponding to 398 subjects. Each subject here has one before-makeup and one after-makeup sample. We refer to this dataset as "T-makeup". Since the T-makeup dataset has limited number of samples per subject, we use the facial symmetry property to generate mirrored face samples. In this way, the size of the training dataset is doubled, which helps in the construction of more robust subspaces. It should be noted that there is no overlap between training and testing subjects since they belong to two different databases. The following parameter values were used in the experiment: the number of subspaces, K, for each descriptor is 75; $\lambda_1 = 0.15$, $\lambda_2 = 0.1$, $\alpha = 180$ for HGORM and LGGP, $\alpha = 80$ for LBP. The number of dimensions of the SRS-LDA feature vector is 220.

6.2. Experiment on the YMU database

This section discusses experiments performed to demonstrate the merits of SRS-LDA for face recognition with makeup variations. In order to evaluate the performance of the proposed face matcher, genuine and impostor scores were generated according to the following protocol:

- Match B against B (B vs. B): Both the images to be compared are before-makeup samples.
- Match A against A (A vs. A): Both the images to be compared are after-makeup samples.
- Match A against B (A vs. B): One of the images to be compared is after-makeup sample while the other is before-makeup sample.

The EERs (Equal Error Rates) of the matching scenarios considered in the YMU database are summarized in Table 2. COTS-1, COTS-2 and COTS-3 are commercial face recognition software, which represent state-of-the-art performances in the task of face recognition. OpenBR [38], LGBP [27], LGGP [26] and HMBP [39] are recent face recognition algorithms proposed in the academic field. The assessed algorithms all have significantly higher EERs, when matching after-makeup to before-makeup samples. The EER of the proposed method for face matching scenario A vs. B is 7.59%. We have significantly reduced the EER from over 20% (see OpenBR, HMBP) to 7.59%. Further, the performance is better than all the three COTS matchers. The proposed method achieves the best EER for the A vs. B case. PLS,[6] on the other hand, obtains an EER of 23.91% on A vs. B. The PLS model was trained with the same feature descriptors as SRS-LDA.

Table 2
Equal error rates (%) corresponding to the eight face matchers and three matching scenarios (B vs. B: matching of before-makeup images, A vs. A: matching of after-makeup images, and A vs. B: one of the images is after-makeup, the other is before-makeup) on YMU database (151 subjects and 604 images). The direct comparison can only be made vertically as they belong to the same matching case. The lower the EER value, the better the performance. The best performance in each matching case has been bolded.

Algorithms	B vs. B	A vs. A	A vs. B
COTS-1	3.85	7.08	12.04
COTS-2	0.69	**1.33**	7.69
COTS-3	0.11	3.29	9.18
OpenBR	6.87	16.44	25.20
LGBP	5.35	8.77	19.71
LGGP	5.36	8.01	19.70
HMBP	6.25	10.87	21.54
Proposed	**0.62**	1.99	**7.59**

We also considered fusing the proposed method with COTS,[7] to further improve the matching performance. Individual match scores generated by different matchers are normalized based on min–max rule, followed by the simple sum rule. As can be seen from Fig. 7, the fused matchers significantly improve the face matching performance in terms of both EER and Genuine Accept Rate (GAR). It is apparent that the proposed method and COTS provide complementary information. As reported in Table 3, COTS-1, COTS-2 and COTS-3 obtain EERs of 12.04%, 7.69%, and 9.18%, respectively. COTS-1, COTS-2 and COTS-3 obtain GARs (GAR: 0.1% FAR) of 48.86%, 76.15%, and 58.48%, respectively. The proposed method achieves EER of 7.59% and GAR of 69.24%. When fusing the proposed method with COTS-2 and COTS-3, we obtain the best results: 83.61% GAR and 5.19% EER, which is better than the fusion of COTS-2 and COTS-3. This clearly indicates that commercial systems have further room for improvement and that the proposed method addresses this issue. It must be noted that

[6] Code used: http://www.cs.umd.edu/~djacobs/pubs_files/PLS_Bases.m.

[7] Fusing with COTS-1 results in poor performance, and is thus not used in the analysis.

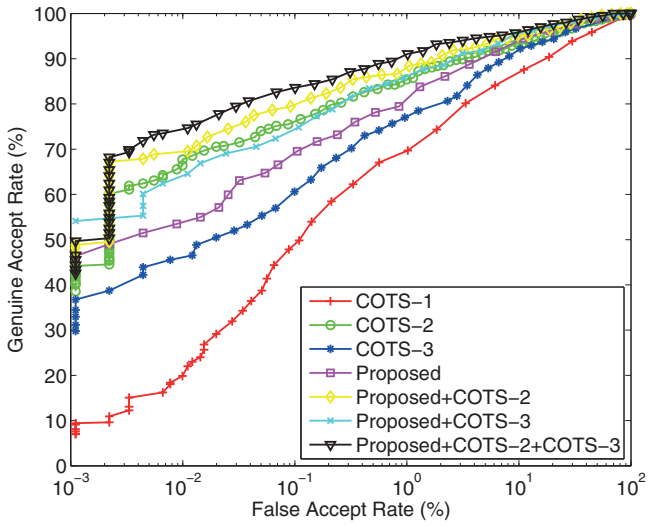
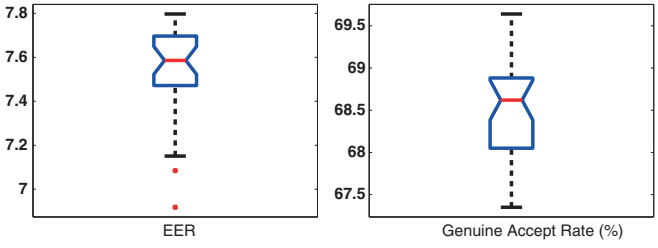


Fig. 7. Performance evaluation of the proposed method and different COTS systems with reported ROC curves (GARs at different FAR: *A* vs. *B*) on YMU database. *Y*-axis is the true positive (i.e., Genuine Accept Rate and *x*-axis is the false positive (i.e., False Accept Rate). GAR is reported at different thresholds of FAR. The higher the GAR, the better the performance under this threshold.

Table 3
The result of fusing the proposed method with different COTS systems on the YMU database. GAR performance at a FAR of 0.1% is reported.

Algorithms	GAR (%)	EER (%)
COTS-1	48.86	12.04
COTS-2	76.15	7.69
COTS-3	58.48	9.18
Proposed	69.24	7.59
Proposed+COTS-2	79.88	5.98
Proposed+COTS-3	74.44	6.59
COTS-2+COTS-3	80.54	5.98
Proposed+COTS-2+COTS-3	**83.61**	**5.19**



(a) EER (b) GAR

Fig. 8. Demonstration (boxplot) of stability of the SRS-LDA method on YMU database: 7.52 ± 0.23 (EER), 68.51 ± 0.59 (GAR at FAR $= 0.1\%$).

the proposed method focuses only on robust feature extraction and matching; this is in contrast to end-to-end COTS matchers which have the advantage of many years of research and are, therefore, likely to have advanced pre-processing and post-processing routines. Inspite of this, the proposed method is observed to be very competitive.

The semi-random sampling scheme resulted from weight learning improves GAR by approximately 4% over random sampling on *A* vs. *B*. Due to the use of semi-random sampling scheme, it is evident that the method is not deterministic. In spite of that, we demonstrate that the solution is rather stable. In order to verify the stability of the SRS-LDA method, we repeat the experiments 31 times and report the distributions of both EER and GAR. As seen from Fig. 8, the EER values range from 6.92% to 7.80% and the GARs at a FAR of 0.1% range from

Table 4
Performance of individual feature descriptors on YMU database. GAR performance at a FAR of 0.1% is reported.

Algorithms	GAR (%)	EER (%)
LGGP	32.82	20.48
HGORM	37.99	20.24
DS-LBP	30.26	19.65
SRS-LDA + LGGP	57.99	11.01
SRS-LDA + HGORM	**63.64**	**10.21**
SRS-LDA + DS-LBP	58.06	11.35

67.35% to 69.64%. In our experiments, we use the median value to report the result. A one-sided hypothesis test at the 5% significance level indicates that the data comes from a population with a mean less than 7.69% EER.

6.2.1. Analysis of individual features

In the proposed framework, we utilize three types of features (LGGP, HGORM and DS-LBP). However, the performance of individual feature descriptors vary (see Table 4). As seen in Table 4, if we use descriptor-based methods only, then the matching performance of *A* vs. *B* is very poor. LGGP, HGORM and DS-LBP obtain EERs of 20.48%, 20.24% and 19.65%, respectively. Here, the LGGP result is reported with slightly different parameters as in Table 2, which is used as a benchmark performance and already revealed to the public. These results are consistent with other descriptor-based methods reported in Table 2. This demonstrates the necessity to use an ensemble learning scheme, to further improve the performance. After adopting the SRS-LDA method, the matching performance increases significantly. The choice of these features (LGGP, HGORM and DS-LBP) are based on empirical testing. We have also analyzed the matching performance of LGBP and HMBP. However, we cannot simply fuse all feature descriptors together due to implicit correlation between some of the feature descriptors. After rigorous invesigation, we found that the fusion of LGGP, HGORM and DS-LBP gives the best result, thereby justifying the use of these features in the proposed framework. HGORM achieves the best overall performance among individual feature descriptors.

6.2.2. Number of subspaces

An important parameter to be considered in SRS-LDA is the number of subspaces used (K). For this purpose, we conducted another experiment to analyze the convergence property of the proposed algorithm. We gradually increase the number of subspaces and compute the corresponding Rank-1 accuracies. As illustrated in Fig. 9, the performance of the proposed algorithm first increases with the number of subspaces, and then stabilizes. HGORM reaches its peak in 26 iterations, while DS-LBP and LGGP reach their respective peaks after 43 and 54 iterations.

In our experiment, we simply set the number of subspaces to be 75. We notice that after this, the performance stabilizes. This illustrates that the proposed method is not very sensitive to the number of subspaces selected. This stabilization aspect is one of the characteristics of the proposed random subspace method.

6.2.3. Computational complexity

The proposed SRS-LDA method is computationally efficient to meet the requirements of practical face recognition systems. Since the random subspaces generated by the ensemble learning algorithm are independent of each other, they can be processed in parallel, making them suitable for large scale face recognition. The training phase is done offline. Hence, only the feature extraction and classification of the test samples impact the computational time during real-time operation. The feature extraction processes (Eqs. (21) and (22)) are

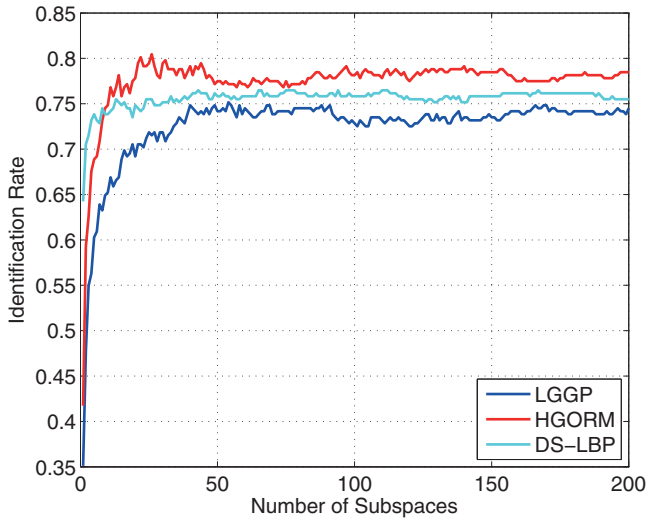


Fig. 9. The impact of number of subspaces on the SRS-LDA algorithm for different feature descriptors (YMU database).

Table 5
The computational time (s) for the proposed algorithm in terms of feature extraction and classification steps.

Algorithms	LGGP	HGORM	DS-LBP	SRC	CRC
Time (s)	4.18	3.00	0.09	0.54	0.19

Table 6
Rank-1 accuracies (%) of different face matchers before and after adding FRGC mugshots to the gallery.

Gallery →	YMU	YMU + FRGC	YMU + FRGC + MIW
Proposed	80.46	78.48	77.81
COTS-2	86.42	85.43	84.44
COTS-3	75.83	74.50	72.19
Proposed + COTS-2	88.41	86.42	85.43
Proposed + COTS-3	85.76	81.46	80.79
COTS-2 + COTS-3	88.41	88.08	86.75
Proposed + COTS-2 + COTS-3	**89.40**	**88.74**	**87.75**

simple linear operations and introduce very limited overhead. The computational complexity is indicated in Table 5.

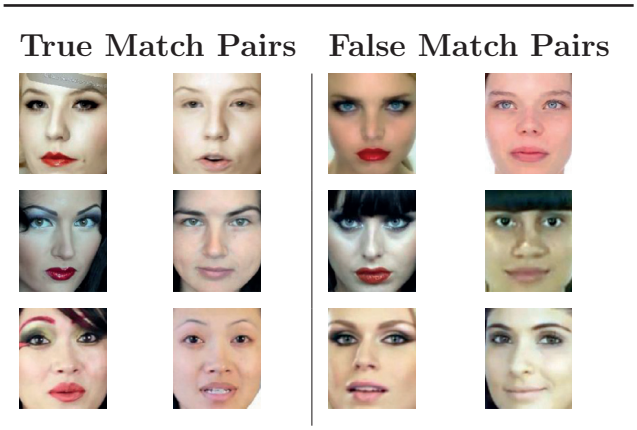
Experiments were conducted using Matlab R2012b on a 64 bit windows operating system with Intel Core i7-2600 CPU at 3.40 GHz and 8GB RAM.[8]

6.3. Large-scale identification experiment

To demonstrate a practical face retrieval experiment, we augment the before-makeup samples in the gallery with a subset of images from the FRGC database [6]. In this experiment, the after-makeup samples from the YMU database are used as probes. A subset of 10,000 (10K: 4574 females + 5426 males) mugshot images are selected from the FRGC database. So the gallery comprises of of 10,302 images (302 before-makeup YMU images and 10K FRGC images). The selected FRGC subset does not contain evident makeup information. We use the exact same matching algorithm as described in Section 6.2. As seen in Table 6, the Rank-1 accuracy drops only marginally in the case of the proposed algorithm in spite of expanding the size of the gallery. This suggests that the proposed method is scalable over a large database.

[8] DS-LBP is implemented in MEX which makes it faster than pure MATLAB: http://www.mathworks.com/matlabcentral/fileexchange/29800-scenesobjects-classification-toolbox.

Table 7
Examples of true and false matches at Rank-1. The selection of these matching pairs (*A* vs. *B*) is based on the face identification scenario.



Furthermore, we conduct another experiment where a set of 112 face images with makeup from the MIW dataset [15] are added to the gallery, resulting in a total of 10,414 gallery samples. The identities of these after-makeup samples in the gallery do not overlap with that of after-makeup samples in the probe. Fig. 10 shows the Cumulative Match Characteristic (CMC) curve, before and after adding the mugshots. The performance of the proposed algorithm is observed to be better than the commercial algorithms on the specified task. The performance is further improved after fusing the proposed method with the COTS matchers.

Examples of successful and unsuccessful matching can be found in Table 7. As illustrated in the examples of false matched pairs, the proposed algorithm sometimes inadvertently learns inter-class variations such as specific hair-styles and poses. Additionally, examples of successful and unsuccessful matching where (a) COTS-2 failed; (b) COTS-3 failed; (c) the proposed method succeeded; (d) and the fusion succeeded, are shown in Table 8.

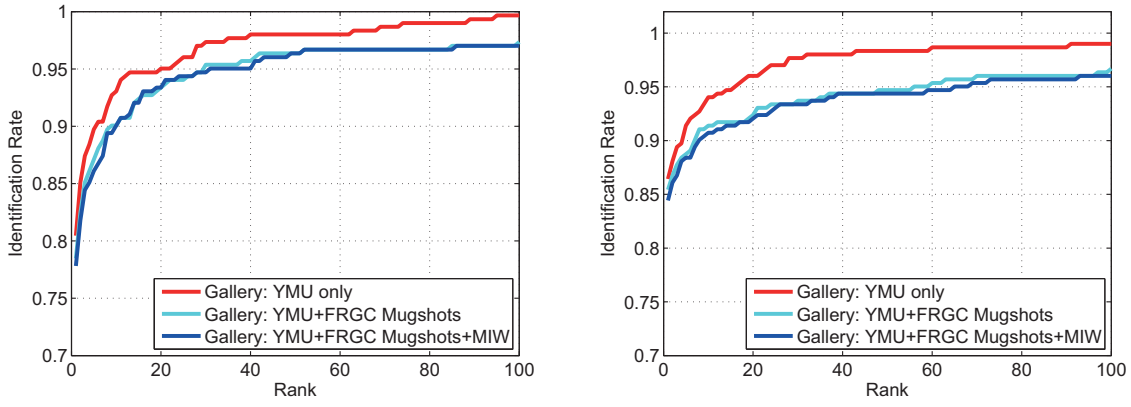
6.4. Comparison against HFR matcher

The proposed method is also evaluated against a state-of-the-art Heterogeneous Face Recognition (HFR) matcher that is used in [23] and [41]. The HFR matcher is a random subspace method that is constructed based on the random sampling of dense LBP (Dense-LBP) and dense SIFT features (Dense-SIFT) [41]. The classification is performed using the Nearest Neighbor Classifier (NNC) or the Sparse Representation Classifier (SRC). We reimplement the algorithm[9] and set the following parameters: $K = 75$ and $\alpha = 180$. The matcher is trained on the T-makeup dataset and tested on the YMU makeup dataset. For NNC and SRC, the selected patches in each subspace are the same. As seen in Table 9, the proposed method is much better than the HFR matcher.

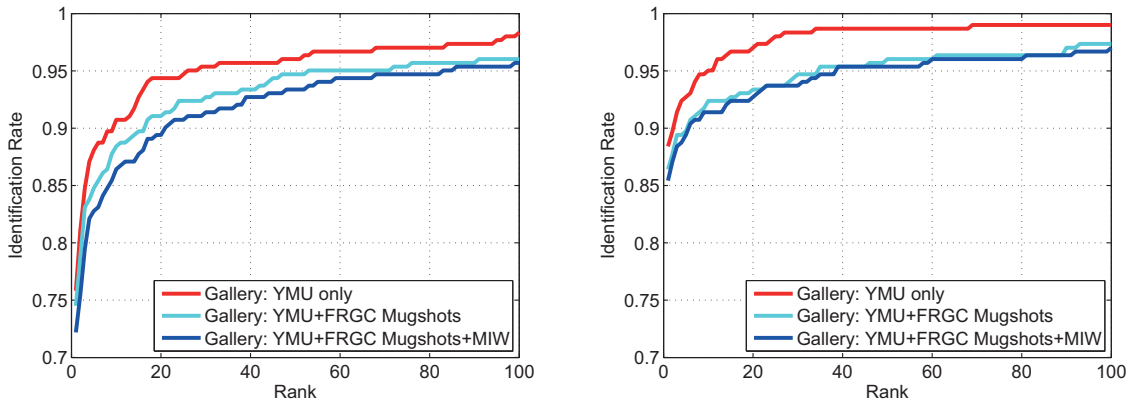
7. Discussions

In this section, we first characterize the datasets used in order to understand their role in the assessment of proposed method for face recognition across makeup. To ensure an objective evaluation, the proposed matcher is pre-trained on the T-makeup dataset, whose subjects are not overlapped with test datasets. The first test set is the public benchmark dataset-YMU, which is used to demonstrate the merits of SRS-LDA for face recognition with makeup variations. Our experimental result shows that the proposed method achieves

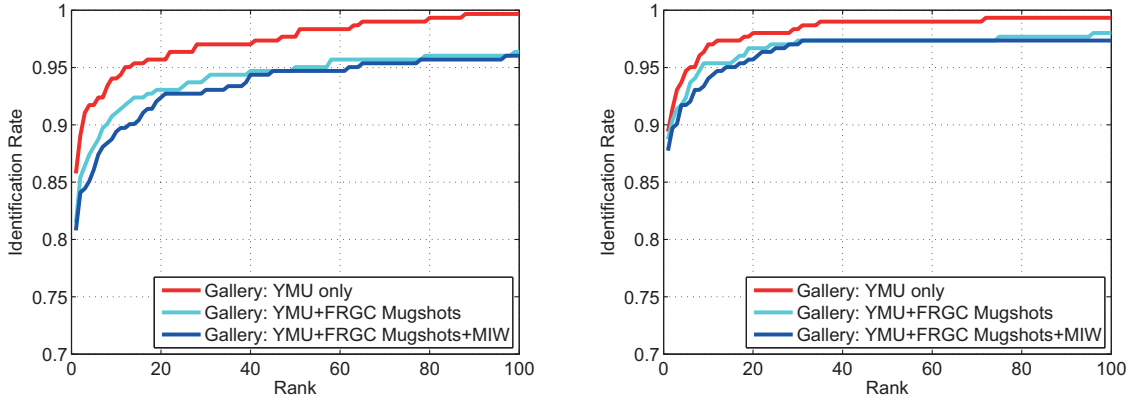
[9] It is not an exact reimplementation and some details may vary.



(a) Proposed (SRS-LDA) (b) COTS-2



(c) COTS-3 (d) Proposed+COTS-2



(e) Proposed+COTS-3 (f) Proposed+COTS-2+COTS-3

Fig. 10. The CMC curves of different face matchers before and after adding mugshots. YMU only: 302 after-makeup samples from YMU are used as probes and 302 before-makeup samples from YMU are used in the gallery; YMU + FRGC Mugshots: 302 after-makeup samples from YMU are used as probes, while 302 before-makeup samples from YMU and 10K FRGC mugshots are used in the gallery; YMU + FRGC Mugshots + MIW: 302 after-makeup samples from YMU are used as probes, while 302 before-makeup samples from YMU, 112 after-makeup samples from MIW [15], and 10K mugshots from FRGC are used in the gallery.

comparable (or superior) performance against leading commercial matchers. The second test set is the FRGC subset, which is used to augment the gallery dataset in order to demonstrate a practical large-scale face identity retrieval experiment. Our experimental result indicates only a marginal drop in Rank-1 accuracy, suggesting that the proposed method is scalable over a large database. The third test set is the MIW dataset, which is used to populate the gallery dataset (including FRGC subset) with makeup samples. Our experimental result reveals that the proposed method results in the smallest drop in performance due to the addition of makeup. That validates the observation that our proposed SRS-LDA method is more resilient to makeup.

Then, we summarize the main observations made from the experimental results.

- The impact of makeup on face recognition is significantly reduced by using the proposed SRS-LDA method. We obtain 7.59% EER, 69.24% GAR (0.1% FAR), and 80.46% Rank-1 accuracy on the YMU database.
- SRS-LDA lends itself to parallel processing; further, SRS-LDA is flexible, where other types of feature descriptors can easily be incorporated into the framework.

Table 8
Examples of successful Rank-1 matches that demonstrate the effectiveness of fusion (Proposed + COTS-2 + COTS-3).

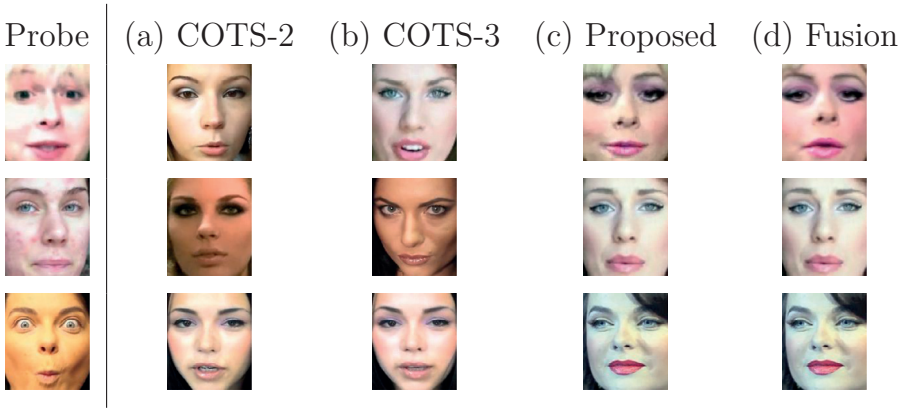


Table 9
Comparison against a HFR matcher [41]. The proposed method uses both SRC and CRC classifiers. GAR performance at a FAR of 0.1% is reported.

Algorithms	GAR (%)	EER (%)	Rank-1 (%)
Proposed	69.24	7.59	80.46
HFR+NNC	39.80	16.39	68.21
HFR+SRC	46.36	19.52	72.52

- Fusion of SRS-LDA with COTS can further improve the matching performance. This clearly demonstrates that SRS-LDA can provide complementary information, which can be utilized to improve the matching accuracy of commercial matchers. We obtain 5.19% EER, 83.61% GAR (0.1% FAR), and 89.40% Rank-1 accuracy on the YMU database.

Though the proposed method is not very complex, it is proved to be effective in addressing the makeup-robust face recognition problem.

8. Summary

This paper presents a method for matching after-makeup images with their before-makeup counterparts. We provide extensive evidence that the proposed method is competitive and outperforms several state-of-the-art descriptor-based methods and commercial matchers. The proposed method uses a patch-based ensemble learning scheme, where multiple subspaces are generated for three different descriptors. The Fisher's separation criteria is used to guide the patch sampling process prior to generating the subspaces. Both SRC and CRC classifiers are utilized for classification in the semi-random subspaces. The final output is the fusion of matching scores from individual descriptors. Experimental results on the YMU database demonstrate the effectiveness of the proposed method. The fusion of proposed method with COTS further improves matching accuracy.

References

[1] A. Jain, A. Ross, K. Nandakumar, Introduction to Biometrics, Springer, 2011.
[2] Z. Liu, C. Liu, Fusion of color, local spatial and global frequency information for face recognition, Pattern Recognit. 43 (8) (2010) 2882–2890.
[3] H. Han, S. Shan, X. Chen, W. Gao, A comparative study on illumination preprocessing in face recognition, Pattern Recognit. 46 (6) (2013) 1691–1699.
[4] Z. Chai, Z. Sun, H.M. Vazquez, R. He, T. Tan, Gabor ordinal measures for face recognition, IEEE Trans. Inf. Forensics Secur. 9 (1) (2014) 14–26.
[5] Y. Taigman, M. Yang, M. Ranzato, L. Wolf, Deepface: closing the gap to human-level performance in face verification, in: Proceedings of Conference on Computer Vision and Pattern Recognition, CVPR, 2014, pp. 1701–1708.
[6] P.J. Phillips, P.J. Flynn, T. Scruggs, K.W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, W. Worek, Overview of the face recognition grand challenge, in: Proceedings of Conference on Computer Vision and Pattern Recognition, CVPR, 2005, pp. 947–954.
[7] G.B. Huang, M. Ramesh, T. Berg, E. Learned-Miller, Labeled faces in the wild: a database for studying face recognition in unconstrained environments, Technical Report 07-49, University of Massachusetts, Amherst, 2007.
[8] L. Wolf, T. Hassner, I. Maoz, Face recognition in unconstrained videos with matched background similarity, in: Proceedings of Conference on Computer Vision and Pattern Recognition, CVPR, 2011, pp. 529–534.
[9] M.D. Marsico, M. Nappi, D. Riccio, H. Wechsler, Robust face recognition after plastic surgery using region-based approaches, Pattern Recognit. 48 (4) (2015) 1261–1276.
[10] S. Ueda, T. Koyama, Influence of make-up on facial recognition, Perception 39 (2010) 260–264.
[11] A. Dantcheva, C. Chen, A. Ross, Can facial cosmetics affect the matching accuracy of face recognition systems? in: Proceedings of Conference on Biometrics: Theory, Applications, and Systems, BTAS, 2012, pp. 391–398.
[12] G. Guo, L. Wen, S. Yan, Face authentication with makeup changes, IEEE Trans. Circuits Syst. Video Technol. 24 (99) (2014) 814–825.
[13] E. Marie-Lena, K. Neslihan, D. Jean-Luc, Facial cosmetics database and impact analysis on automatic face recognition, in: Proceedings of Multimedia Signal Processing, MMSP, 2013.
[14] R. Feng, B. Prabhakaran, Quantifying the makeup effect in female faces and its applications for age estimation, in: Proceedings of International Symposium on Multimedia, ISM, 2012, pp. 108–115.
[15] C. Chen, A. Dantcheva, A. Ross, Automatic facial makeup detection with application in face recognition, in: Proceedings of International Conference on Biometrics, ICB, 2013.
[16] J. Hu, Y. Ge, J. Lu, X. Feng, Makeup-robust face verification, in: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, 2013, pp. 2342–2346.
[17] S. Wang, D. Zhang, Y. Liang, Q. Pan, Semi-coupled dictionary learning with applications to image super-resolution and photo-sketch synthesis, in: Proceedings of Conference on Computer Vision and Pattern Recognition, CVPR, 2012, pp. 2216–2223.
[18] D. Yi, R. Liu, R. Chu, Z. Lei, S.Z. Li, Face matching between near infrared and visible light images, in: Proceedings of International Conference on Biometrics, ICB, 2007, pp. 523–530.
[19] G. Ma, M. Liu, L. Wu, H.R. Karimi, Local patch-based subspace ensemble learning via totally-corrective boosting for gait recognition, in: Proceedings of International Conference on Machine Learning, ICML Workshop, 2013.
[20] P. Li, K.L. Chan, S.M. Krishnan, Learning a multi-size patch-based hybrid kernel machine ensemble for abnormal region detection in colonoscopic images, in: Proceedings of Conference on Computer Vision and Pattern Recognition, CVPR, 2005, pp. 670–675.
[21] X. Wang, X. Tang, Random sampling for subspace face recognition, Int. J. Comput. Vis. 70 (1) (2006) 91–104.
[22] Y. Zhu, J. Liu, S. Chen, Semi-random subspace method for face recognition, Image Vis. Comput. 27 (9) (2009) 1358–1370.
[23] B. Klare, A.K. Jain, Heterogeneous face recognition: matching NIR to visible light image, in: Proceedings of International Conference on Pattern Recognition, ICPR, 2010, pp. 1513–1516.
[24] D. Kang, H. Han, A.K. Jain, S.-W. Lee, Nighttime face recognition at large standoff: cross-distance and cross-spectral matching, Pattern Recognit. 47 (12) (2014) 3750–3766.

[25] T. Dietterich, Ensemble methods in machine learning, in: Proceedings of the First International Workshop on Multiple Classifier Systems, vol. 1857, 2000, pp. 1–15.
[26] C. Chen, A. Ross, Local gradient gabor pattern (LGGP) with applications in face recognition, cross-spectral matching and soft biometrics, in: Proceedings of SPIE Biometric and Surveillance Technology for Human and Activity Identification, 2013.
[27] W. Zhang, S. Shan, W. Gao, X. Chen, H. Zhang, Local gabor binary pattern histogram sequence (LGBPHS): a novel non-statistical model for face representation and recognition, in: Proceedings of International Conference on Computer Vision, ICCV, 2005, pp. 786–791.
[28] A. Ross, A. Jain, Information fusion in biometrics, Pattern Recognit. Lett 24 (13) (2003) 2115–2125.
[29] B. Zhang, S. Shan, X. Chen, W. Gao, Histogram of Gabor Phase Patterns (HGPP): a novel object representation approach for face recognition, IEEE Trans. Image Process. 16 (1) (2007) 57–68.
[30] Z. Sun, T. Tan, Ordinal measures for iris recognition, IEEE Trans. Pattern Anal. Mach. Intell. 31 (12) (2009) 2211–2226.
[31] S.S. Stevens, On the theory of scales of measurement, Science 103 (2684) (1946) 677–680.
[32] J. Chen, S. Shan, C. He, G. Zhao, P. Matti, X. Chen, W. Gao, WLD: a robust local image descriptor, IEEE Trans. Pattern Anal. Mach. Intell. 32 (9) (2010) 1705–1720.
[33] Z. Sun, T. Tan, Y. Wang, S. Li, Ordinal palmprint representation for personal identification, in: Proceedings of Conference on Computer Vision and Pattern Recognition, CVPR, vol. 1, 2005, pp. 279–284.
[34] L. Zhang, M. Yang, X. Feng, Sparse representation or collaborative representation: which helps face recognition? in: Proceedings of International Conference on Computer Vision (ICCV), IEEE, 2011, pp. 471–478.
[35] J. Mairal, F. Bach, J. Ponce, G. Sapiro, Online learning for matrix factorization and sparse coding, J. Mach. Learn. Res. 11 (2010) 19–60.
[36] B. Klare, M.J. Burge, J.C. Klontz, R. Bruegge, A.K. Jain, Face recognition performance: role of demographic information, IEEE Trans. Inf. Forensics Secur. 7 (6) (2012) 1789–1801.
[37] P.J. Grother, G.W. Quinn, P.J. Phillips, MBE 2010: Report on the Evaluation of 2D Still-image Face Recognition Algorithms, 7709, National Institute of Standards and Technology, 2010, pp. 1–61.
[38] J. Klontz, B. Klare, S. Klum, E. Taborsky, M. Burge, A. Jain, Open source biometric recognition, in: Proceedings of Conference on Biometrics: Theory, Applications and Systems, BTAS, 2013.
[39] M. Yang, L. Zhang, L. Zhang, D. Zhang, Monogenic binary pattern (MBP): a novel feature extraction and representation model for face recognition, in: Proceedings of International Conference on Pattern Recognition, ICPR, 2010, pp. 2680–2683.
[40] C. Chen, A. Dantcheva, A. Ross, Impact of facial cosmetics on automatic gender and age estimation algorithms, in: Proceedings of Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, VISAPP, 2014, pp. 182–190.
[41] B. Klare, A. Jain, Heterogeneous face recognition using kernel prototype similarities, IEEE Trans. Pattern Anal. Mach. Intell. 35 (6) (2013) 1410–1422.
[42] J. Wright, A. Yang, A. Ganesh, S. Sastry, Y. Ma, Robust face recognition via sparse representation, IEEE Trans. Pattern Anal. Mach. Intell. 31 (2) (2009) 210–227.