This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2016.2535284, IEEE Transactions on Image Processing

Aging Face Recognition: A Hierarchical Learning Model Based on Local Patterns Selection

Zhifeng Li, *Senior Member, IEEE*, Dihong Gong, Xuelong Li, *Fellow, IEEE*, and Dacheng Tao, *Fellow, IEEE*

***Abstract*—Aging face recognition refers to matching the same person's faces across different ages, e.g. matching a person's older face to his (or her) younger one, which has many important practical applications such as finding missing children. The major challenge of this task is that facial appearance is subject to significant change during the aging process. In this paper, we propose to solve the problem with a hierarchical model based on two-level learning. At the first level, effective features are learned from low-level microstructures, based on our new feature descriptor called Local Pattern Selection (LPS). The proposed LPS descriptor greedily selects low-level discriminant patterns in a way such that intra-user dissimilarity is minimized. At the second level, higher-level visual information is further refined based on the output from the first level. To evaluate the performance of our new method, we conduct extensive experiments on the MORPH dataset (the largest face aging dataset available in the public domain), which show a significant improvement in accuracy over the state-of-the-art methods.**

Keywords*—Face recognition, aging faces, feature descriptor.

I. Introduction

AUtomatic face recognition is an important yet challenging problem, and has gained great progress in recent years due to better feature representation methods [21], [23], [24], [41], [47], [50], [51], [52], [53], [54], [55], [56], [57], [58], [59], [60] and feature classification models[61], [62], [63], [64], [65], [66], [67], [68], [69], [70], [71]. An emerging research topic in face recognition community is aging face recognition, which has many useful real-world applications, e.g., finding missing children and identifying criminals based on photographs or identity 'mug shots' [1][2]. While considerable progresses have been made on face recognition, aging face recognition still remains as a major challenge in real world application of face recognition systems. This challenge is mostly attributed to the significant intra-personal variations caused by the aging process. As illustrated in Figure 1, the cross-age faces (for one of the subjects in MORPH database [30]) contain significant intra-personal variations.

Actually, age related face image analysis has only been studied in recent years. Most existing works focus on age estimation [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [28], [48], and aging simulation [14], [15], [16], [17], [18]. There are very limited amount of works directly on aging face recognition. A typical aging face recognition approach is to use face modeling to synthesize and render the face images to the same age as the gallery image before recognition [4], [14], [18], [33]. Due to the strong parametric assumptions and the complexity of the algorithm, these methods are expensive to compute and the results are often unstable for real world face recognition. Recently, discriminative methods are proposed for aging face recognition [19], [20], [34], [35], [43], [44], [49]. The method in [19] uses gradient orientation pyramid (GOP) for feature representation, combined with support vector machine for verifying faces across age progression. The method in [20] combines both Scale Invariant Feature Transform (SIFT) [23] and Multi-scale Local Binary Pattern (MLBP) [26] with a random sampling based fusion framework to improve the performance of aging face recognition. Some variants of random sampling LDA approach has also been proposed in [34], [35] to address the face aging problem in face recognition. They are shown to be much more robust with fewer requirements on parameters and the training data and have demonstrated better results than previous methods. More recent works on aging face recognition include [43], [44], [49], which has notably improved the performance of aging face recognition.

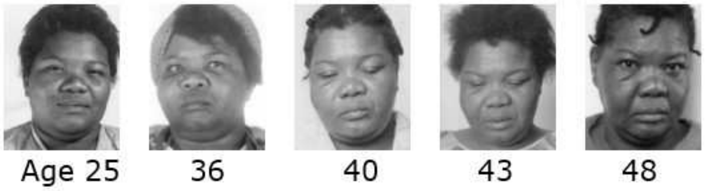

Fig. 1: Example faces for one of the subjects in the MORPH database [30]

In this paper, we propose a two-level hierarchical learning model to address this problem. In this model, effective features are first learned from low-level pixel structures, based on our new feature extraction algorithm called Local Pattern Selection (LPS). Low-level common information is widely believed to be very beneficial to cross-age face recognition, and the LPS algorithm maximizes this information between cross-age faces. At the second level, higher-level visual information is refined by learning subspace analysis algorithms. The advantage of this model is that, when compared with traditional paradigms where learning happens only in higher levels via classification algorithms, our model has better learning capabilities and is thus able to adaptively capture more useful information. Also note that strong model learning capability is essential to a successful face recognition algorithm, as confirmed by the recent success in deep learning [46], [47]. Extensive experiments are conducted on the MORPH dataset (Album 2), the largest publicly available facial aging dataset, to validate the effectiveness of our new approach over the state-of-the-art ones.

The rest of this paper is organized as follows. In Section II, we introduce our LPS algorithm for learning low-level visual structures. In Section III, we present an efficient feature refinement framework to enhance high-level visual information. In Section IV, we introduce the experimental results. We conclude this paper in Section V.

II. Learning Low-Level Visual Structures

In this section, we present a novel algorithm, called Local Pattern Selection (LPS), for learning low-level visual structures. First, we introduce the motivation behind the LPS algorithm, and then formulate the algorithm. Lastly, we provide a detailed discussion of the algorithm.

A. The Motivation

Facial appearance changes significantly in the human aging process, which makes aging face recognition highly challenging. The difficulty can be attributed to the large intra-user dissimilarity caused by aging. To overcome this challenge, we propose to learn the feature encoder, which can reduce intra-user dissimilarity at the micro-structure level. As illustrated in Figure 2, we encode pixels by converting them into integer codes. The code assignment is determined by which partition (leaf node) the pixel falls into. By encoding each pixel in such a manner that corresponding pixels of the same subject at different ages fall into the same partition (pixels falling into the same partition are assigned the same code), the intra-user dissimilarity can be effectively reduced. This intuition may not work properly without regularization, however. Consider a case in which all the pixels fall into the same partition; this trivial case does give an optimal solution consistent with our intuition. To avoid this trivial solution, we regularize the problem by requiring partition size to be distributed as evenly as possible.

B. Terms and Definitions

To facilitate understanding, we first introduce several essential terms and definitions. In this paper, all the face images we use are gray scale images cropped to the same size of 200 × 150, with faces properly aligned to ensure good correspondence between pixels [1].

Definition 1 (Pixel feature)
Each pixel is associated with a corresponding pixel feature ($8-dimensional$ vector) that is formed by sampling its eight neighbors at radius r, and centered by subtracting the center pixel.

Figure 2 illustrates different sampling patterns as well as how pixel features are extracted.

Definition 2 (Matching pixel features)
For each pair of pixel features, we call them matching pixel features if they are from two different images of the same subject and extracted at the same location.

The 'same subject' means the same person, and the 'same location' means that the pixel locations in the images are the same (e.g. we may refer a pixel by its location with 2D coordinate (r, c) for the pixel at row r and column c of the image). The correspondence between pixels is ensured by proper alignment of face images.

Definition 3 (Encoding tree)
An encoding tree is a binary decision tree with internal nodes and leaf nodes, where internal nodes are associated with (attribute, threshold) and leaf nodes are associated with distinct decimal codes.

The encoding tree is used to encode each pixel into a decimal code. Given a pixel, we convert it into an integer code by first extracting its pixel feature, followed by passing this pixel feature through the encoding tree, and the code is determined based on which leaf node the pixel feature reaches, as illustrated in Figure 3. When a pixel feature passes through an encoding three, the pixel feature is directed to either left or right branch of a node by $(attribute, threshold)$ pair associated with that node, where the attribute refers to the element of the pixel feature to be encoded. For example, if the $attribute$ of the pixel feature is less than the $threshold$, then this pixel feature will be directed to the left branch (right branch otherwise).

Definition 4 (Support pixel features)
In the phase of encoding tree training, for a given leaf node, its support pixel features are the set of pixel features that can reach this leaf node.

The encoding tree divides the entire pixel feature space into partitions, with each leaf node corresponding to one partition, as illustrated in Figure 3. The support pixel features of a leaf node are visually the training samples that fall into the partition corresponding to that leaf node.

C. The Formulation

With the above definitions, we are now ready to formulate the algorithm. As described previously, the training aim of LPS algorithm is to learn an encoding tree (see Definition 3) that can capture valuable common information among cross-age faces. Formally, suppose we are given N pairs of training face images of size $H \times W$, with each pair of face images from the same subject at different ages. The corresponding pixel features (see Definition 1) of these image pairs are denoted as:

$$A = \{(\vec{x}^n_m, \vec{y}^n_m) | m = 1, \dots, M; n = 1, \dots, N\} \qquad (1)$$

where $M = H \times W$ is the total number of pixels in an image. We say $\vec{x}^n_m$ and $\vec{y}^n_m$ are matching pixel features (see Definition 2), as they belong to the same pixel location in the images of the same subject. For any encoding tree T of L leaf nodes, suppose the T assigns each pixel feature in A with a code based on which leaf node that pixel finally reaches, resulting in an encoded set:

$$C = \{(u^n_m, v^n_m) | m = 1, \dots, M; n = 1, \dots, N\} \qquad (2)$$

[1] Strict pixel-level correspondence is not required. By training with large amount of pixels, misalignment effect can be alleviated.

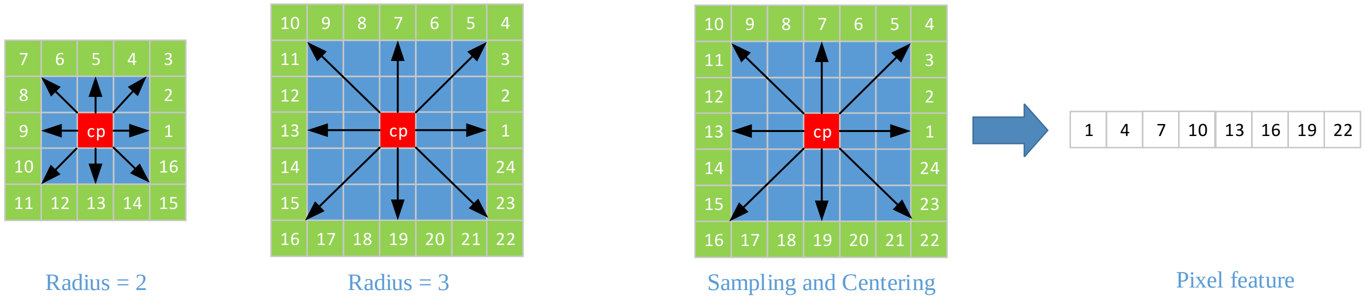

Fig. 2: The illustration for different sampling patterns and the formation of pixel features. The pixel feature of cp is formed by first sampling its eight neighbors at specific radius, followed by subtracting itself so that pixel feature is centered.

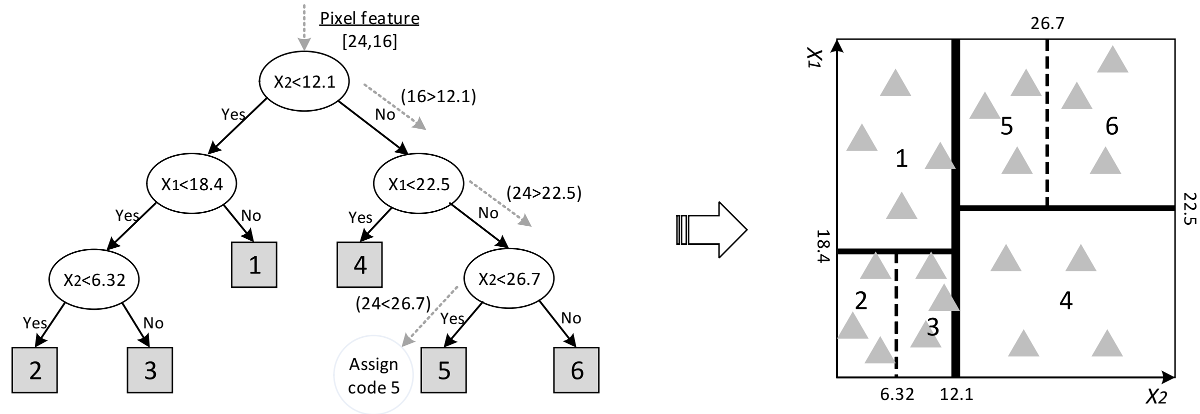

Fig. 3: The illustration for encoding tree. The internal nodes direct each pixel feature to either left or right branch by comparing it to the threshold: if its component is less than the threshold, then direct the pixel feature to left branch, otherwise right branch. Once a pixel feature reaches a leaf node, we encode the pixel by assigning a code associated with that leaf node. The right figure visualizes the partitions corresponding to the encoding tree on left. $X1$ and $X2$ represent two different attributes.

We measure the performance of T by:

$$U = \alpha \frac{\sum_{n=1}^{N}\sum_{m=1}^{M}\delta(u_m^n, v_m^n)}{M \times N} + (1-\alpha)\frac{\sum_{l=1}^{L} p_l log\frac{1}{p_l}}{logL} \qquad (3)$$

Where α is a tradeoff factor between the two terms, while $\delta(x, y)$ is a function takes value 1 only if $x = y$ and 0 otherwise. Finally, the p_l is interpreted as the fraction of pixel features falling into the $l - th$ leaf node, given by:

$$p_l = \frac{\sum_{n=1}^{N}\sum_{m=1}^{M}\delta(u_m^n, l) + \sum_{n=1}^{N}\sum_{m=1}^{M}\delta(v_m^n, l)}{2 \times M \times N} \qquad (4)$$

The U defined in Eqn 3 is referred as *utility* of an encoding tree for the rest of this paper. We explain the two terms in Eqn 3 as follows. The first term is interpreted as common information between cross-age faces. Larger value indicates higher fraction of matching pixel features having the same code, and thus higher common information. The second term serves as regularization. This regularization encourages even partitioning for the pixel feature space. It measures the entropy of the pixel features distribution among leaf nodes. The higher value means higher entropy, and thus evener distribution. The use of evenness for regularization is motivated by [45]. In fact, according to the information theory, the evener the distribution is, the more informative the encoding becomes.

D. Learning the encoding tree

In this part, we elaborate the LPS algorithm based on formulations in section II-C. Specifically, we grow an encoding tree such that the utility given in Eqn 3 is maximized. The basic idea of our algorithm is to grow the encoding tree incrementally until the expected number of leaf nodes L is reached. At each step, we select the best node that maximizes the increase in utility for expansion.

Let's denote the tree with K leaf nodes ($1 \leq K < L$) as T_K and the corresponding utility as U_K. Now we extend T_K into T_{K+1} by splitting node w into two children nodes w_l and w_r. Denote the sets of support pixel features (see Definition 4) of node w as:

$$S_w^1 = \{I_1^{(1)}, \dots, I_{n_1}^{(1)} | I_i^{(1)} \in 1, \dots, M \times N\} \qquad (5)$$

$$S_w^2 = \{I_1^{(2)}, \dots, I_{n_2}^{(2)} | I_i^{(2)} \in 1, \dots, M \times N\} \qquad (6)$$

Algorithm 1: Local Patterns Selection

```
Input: The number of leaf nodes $L$, the tradeoff factor $\alpha$,
       and training image pairs $\{(I_n^1, I_n^2)|n = 1, \ldots, N\}$.
Output: Encoding tree $T$.
/* Pixel features extraction.                                   */
begin
    Convert images into a set of pixel features as described
    in Eqn 1:
        $A = \{(\vec{x}_m^n, \vec{y}_m^n)|m = 1, \ldots, M; n = 1, \ldots, N\}$
/* Encoding tree initialization.                                */
begin
    Initialize encoding tree $T$ by adding one leaf node $w$,
    whose indices of support pixel features are:
        $S_w^1 = \{I_1^{(1)}, \ldots, I_{n_1}^{(1)}|n_1 = M \times N\}$
        $S_w^2 = \{I_1^{(2)}, \ldots, I_{n_2}^{(2)}|n_2 = M \times N\}$
    $w.a \leftarrow 0$, $w.t \leftarrow 0$, and $w.\Delta u \leftarrow -\inf$.
/* Encoding tree learning                                       */
begin
    for $step = 2 \rightarrow L$ do
        for each leaf node $w$ do
            if $w.\Delta u \neq -\inf$ then
                /* Node has been evaluated.                     */
                continue;
            else
                for $k = 1 \rightarrow 8$ do
                    for $z = min(S_w^1, S_w^2) \rightarrow max(S_w^1, S_w^2)$ do
                        Evaluate increase of utility $\Delta u$ with
                        $attribute = k$ and $threshold = z$.
                Let maximum $\Delta u^*$ is achieved at $(k^*, z^*)$.
                Update: $w.\Delta u = \Delta u^*$, $w.a \leftarrow k^*$, $w.t \leftarrow z^*$.
        Let $w^*$ has maximum $\Delta u$ over all leaf nodes.
        Split $w^*$ into two children nodes $l$ and $r$.
        $l.a \leftarrow 0$, $l.t \leftarrow 0$, and $l.\Delta u \leftarrow -\inf$.
        $r.a \leftarrow 0$, $r.t \leftarrow 0$, and $r.\Delta u \leftarrow -\inf$.
        Update $S_l^1$, $S_l^2$, $S_r^1$, $S_r^2$ based on Eqn 5, 6.
    Assign distinct codes to leaf nodes, and return $T$.
```

Where S_w^1 is a set of indices of pixel features from the first image of training face pairs, while the S_w^2 is from the second image. Consider that node w is split into two nodes at attribute a and threshold t, then the support pixel features in that node will be partitioned into the left child node if attribute a of features is less than t (into right child otherwise), resulting in two new sets of support pixel features for node w_l: $S_l^1 = \{I_1^{(1)}, \ldots, I_{n_{l_1}}^{(1)}\}$ and $S_l^2 = \{I_1^{(2)}, \ldots, I_{n_{l_2}}^{(2)}\}$. Similarly, for node w_r we also have two new sets $S_r^1 = \{I_1^{(1)}, \ldots, I_{n_{r_1}}^{(1)}\}$ and $S_r^2 = \{I_1^{(2)}, \ldots, I_{n_{r_2}}^{(2)}\}$. The utility of the new tree T_{K+1} can thus be formulated as:

$$U_{K+1} = U_K + \alpha \frac{\Delta A}{M \times N} + (1 - \alpha) \frac{\Delta E}{logL} \tag{7}$$

Where ΔA is the increased number of matching pixels (negative) reaching into the same leaf node, given by:

$$\Delta A = \sum_{x \in S_l^1} \sum_{y \in S_l^2} \delta(x, y) + \sum_{x \in S_r^1} \sum_{y \in S_r^2} \delta(x, y) - \sum_{x \in S_w^1} \sum_{y \in S_w^2} \delta(x, y) \tag{8}$$

The ΔE is amount of increase in entropy, given by:

$$\Delta E = p_{w_l} log \frac{1}{p_{w_l}} + p_{w_r} log \frac{1}{p_{w_r}} - p_w log \frac{1}{p_w} \tag{9}$$

In expanding T_K to T_{K+1}, the LPS algorithm greedily selects the best node w^*, such that the following increased amount of utility is maximized:

$$\Delta U = U_{K+1} - U_K \tag{10}$$

Note that the number of expected leaf nodes L and tradeoff factor α are determined by cross validation. The Algorithm 1 describes the detailed steps in learning the encoding tree.

E. LPS-based feature extraction

The Algorithm 1 learns an encoding tree that encodes given image by converting each pixel into decimal codes based on the leaf node that pixel reaches in the tree. In this part, we briefly introduce how we extract over-completed features based on encoded images. The techniques we use include multiple scaling and dense sampling [42]. Specifically, we first train multiple encoding trees based on different sampling radii (e.g. $1, 3, 5, 7$) as illustrated in Figure 2. Then for each encoded image, we extract local features by calculating the histograms of small patches formed by dividing image into overlapping (with overlapping factor 0.5) fixed size (e.g. 16×16) areas. The final features of an image are formed by concatenating local features at all sampling radii.

F. Discussion

The major advantage of the proposed LPS feature descriptor, compared with the state-of-the-art descriptors, is that it can adaptively learn an encoding tree that discovers useful common information between cross-age faces. In addition, another interesting discovery is that, by introducing the regularization term in Eqn 3, we can obtain much evener code distribution than the handcrafted descriptors such as uniform Local Binary Patterns (LBP) [21]. The evenness of code distribution usually encourages informative codes, as noted previously. In this part we will give a detailed analysis of these concepts as well as the training time complexity of the algorithm.

1) Common information analysis: We compare the common information of cross-age faces against the uniform LBP descriptor. The uniform LBP is one of the most successful feature descriptors in the literature, whose design is based on an empirical observation that for most of the real-world images, some codes occur consistently with much higher frequency than others. The uniform LBP then assigns distinct values (from 1 to 58) to the top 58 codes with the highest frequency and a single value to all the rest, to convert 256 codes into 59 codes. To compare the common information between these

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2016.2535284, IEEE Transactions on Image Processing

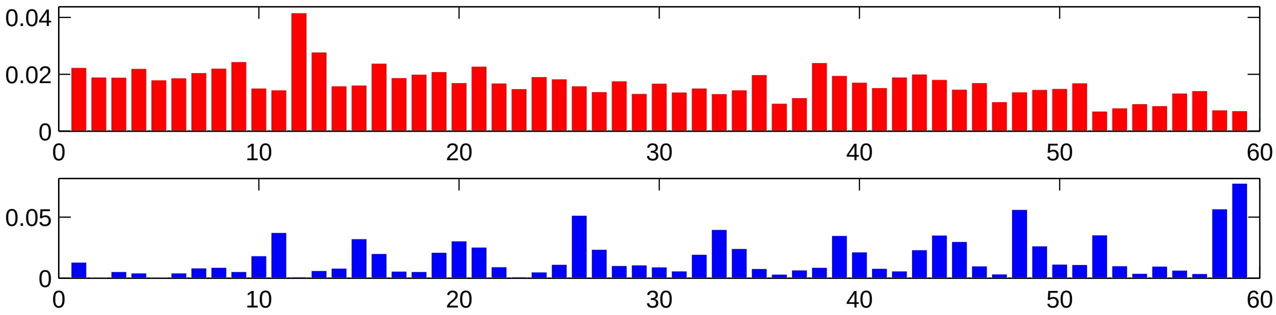

Fig. 4: Code emergence frequency of our method (top) and the uniform LBP (bottom)

TABLE I: The common information comparison

Radius	1	3	5	7
uniform LBP	0.1269	0.0659	0.1633	0.0978
LPS	0.1839	0.1200	0.2397	0.1483

two methods, we first formally define a metric for common information between two images of size $H \times W$ as follows:

$$\frac{\sum_{i=1}^{H\times W} \delta(C_1(i), C_2(i))}{H \times W} \tag{11}$$

Where C_1 and C_2 are the encoded images of the first and second image, respectively. Then we compute the averaged common information of 100 pairs of testing face images at four different sampling radii, the results of which are shown in the Table I. From these numbers, we can conclude that the common information of our method is consistently higher than the handcrafted uniform LBP descriptor in all of the sampling radii. This reveals that our LPS algorithm is able to learn an encoding tree that increases the amount of common information contributes to higher recognition accuracy.

2) Code distribution analysis: The evenness of code distribution usually contributes to more discriminative codes as suggested by [45]. In this part we analyse the code distribution of the LPS algorithm by comparing it against the uniform LBP as before. From the design of uniform LBP, we can see that it also encourages evenness, based on an assumption that some codes have consistently higher emergence frequency in all images.

To compare code distribution between these two methods, we plot their code emergence frequencies based on 1000 cross-age face images at sampling radius 1. As illustrated in Figure 4, the distribution of codes generated by the LPS algorithm has much better evenness than the uniform LBP method. Thanks to the regularization term in Eqn 3, the LPS algorithm encourages evenness in addition to higher amount of common information.

3) Time complexity analysis: At the end of this section, we briefly analyse the time complexity of our algorithm. The Algorithm 1 tends to generate a balanced binary tree (so that codes are evenly distributed), with totally L leaf nodes. Thus the tree has height $O(logL)$. To split all nodes at level k, we need to evaluate Eqn 10 for $O(N)$ times, where N is the number of training image pairs. Thus, to grow a tree of $O(logL)$ levels, the overall time complexity of the Algorithm 1 is $O(NlogL)$. On a single desktop computer, it takes around 20 seconds to train an encoding tree with $L = 64$, and $N = 1,000$ pairs of images of size 200×150. This training time grows linearly as N increases, and logarithmically as L increases.

III. High-Level Visual Information Refinement

In the previous section, we present a novel algorithm for learning low-level visual structures. The features extracted based on LPS algorithm are usually of very high dimension (e.g. $100K$) due to the employment of both multiple scaling and dense sampling techniques. The extremely high dimension of the facial features are not good for both storage and face matching. In addition, there might be a lot of redundant information as well as noise among these features. Thus, there is a need for further refinement for low-level visual features.

The most popular feature refinement techniques such as Fisher linear discriminant analysis or universal subspace analysis [42] involve eigendecomposition of a matrix of size $min(N, D)$ where N is the number of training image pairs and D is the dimension of features. The time complexity of eigendecomposition is around $min\{O(N^3), O(D^3)\}$. Thus, it is almost infeasible to do an eigendecomposition with very large N and D. In this part we present a scalable high-level visual information refinement framework to efficiently reduce the noise as well as redundant information, and preserve only the most crucial information for face recognition. The Figure 5 provides an illustration for our framework.

A. Bootstrap aggregating: divide data by sampling

The Bootstrap aggregating, also known as Bagging, is an ensemble meta-algorithm designed to improve the stability and accuracy of machine learning algorithms. The Bagging repeatedly selects a series of training subsets of M samples ($M < N$) and based on these subsets it trains a series of classifiers, which will be fused into a unified classifier. In our implementation, we set $M = 2,000$ (where $N = 10,000$) and create $K = 20$ training subsets for classifier training. In

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2016.2535284, IEEE Transactions on Image Processing

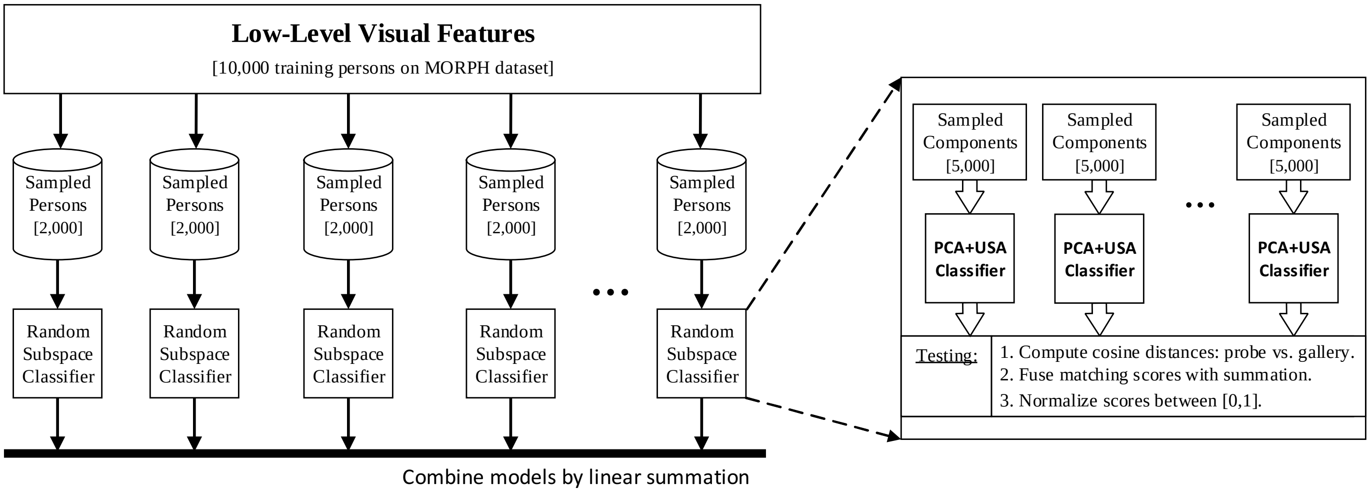
Low-Level Visual Features
[10,000 training persons on MORPH dataset]
Sampled Persons [2,000]
Sampled Persons [2,000]
Sampled Persons [2,000]
Sampled Persons [2,000]
Sampled Persons [2,000]
Random Subspace Classifier
Random Subspace Classifier
Random Subspace Classifier
Random Subspace Classifier
Random Subspace Classifier
Combine models by linear summation
Sampled Components [5,000]
Sampled Components [5,000]
Sampled Components [5,000]
PCA+USA Classifier
PCA+USA Classifier
PCA+USA Classifier
Testing:
1. Compute cosine distances: probe vs. gallery.
2. Fuse matching scores with summation.
3. Normalize scores between [0,1].

Fig. 5: The illustration of the high-level feature refinement framework. At the training stage, we first generate 20 training subsets with each consisting of 2,000 samples using Bagging techniques, based on which we train 50 random subspace classifiers using PCA+USA as base classifier. Each PCA+USA base classifier is trained based on 5,000 randomly sampled components. At the testing stage, the matching scores from probe person to gallery persons are the fused scores of all the random subspace classifiers, where matching scores of random subspace classifiers are the again the fused cosine scores of all the PCA+USA classifiers with scores normalized between $[0, 1]$ (best matching corresponding to 1 while worst matching corresponding to 0).

this manner, we have efficiently limit the number of training samples for each sub-classifier, so that we can scale up to larger dataset. Note that for dataset of extremely large number of training samples (e.g. $N > 10,000$), we may need to increase the K accordingly. But the training time grows linearly as K increases, instead of cubically.

B. Random subspace: divide features by sampling

The Bagging can efficiently reduce the computational cost as noted previously. To further improve the stability of the final classifier, we incorporate the random subspace sampling technique [32] into our framework. For a given training subset, the random space algorithm repeatedly samples subsets of features (e.g. take $5K$ our of $100K$) randomly, as illustrated in Figure 5. In our implementation, we create 50 subsets based on random subspace sampling, and use the Universal Subspace Analysis (USA) as base classifiers.

C. Discussion

Our second level classification framework is designed to be scalable. As described in Section III-A, as the number of training samples increases, the training time increases linearly. Specifically, for the Morph 2 dataset, the major computational cost involves 2000x2000 square matrix eigendecomposition. There are K=20 of these decompositions, and these decompositions can be computed in parallel. On a 6-core machine it takes 3.64 seconds for each of the decomposition, and 74.31 seconds in total for the second level (including other overhead). We had focused on training time analysis, since for testing, it only involves matrix-vector multiplication, and thus it is very fast.

IV. Experiment

In this section we conduct extensive experiments to evaluate the effectiveness of our method. There are two well-known public domain datasets for aging faces: FGNET [36] and MORPH [30]. The FGNET dataset is a relatively small dataset consists of 1002 face images from 82 different persons. The MORPH dataset has two separate versions: Album 1 and Album 2. The MORPH Album 1 contains 1690 face images from 625 different persons. The MORPH Album 2 contains $78,000$ images from $20,000$ different people with each person having at least two cross-age face images. Given that the Album 2 has the largest size and there have been some benchmark results on it, we will thus focus on the MORPH Album 2 dataset in this paper. Following the benchmark conventions, we use two images (with maximum age gap) for each of the $20,000$ persons.

We partition the Album 2 dataset following the same scheme in [43]: the first $10,000$ pairs of faces are reserved for testing, and the last $10,000$ pairs are used for training. There is no overlapping subjects between training and testing sets. All the face images are automatically preprocessed through the following steps: (1) rotate the face images to align to the vertical face orientation; (2) scale the face images so that the distance between the two eyes is the same for all the images; (3) crop the face images into 200×150 tightly to remove the background and the hair region.

A. Parameter exploration

In the first experiment, we show how to determine the model parameters in Eqn 3 using cross validation. Particularly, during the training phase the $10,000$ training face pairs are divided

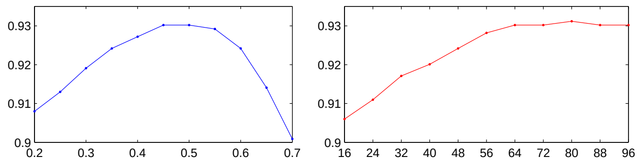

Fig. 6: Validation accuracies vs. α (left) and vs. L (right).

TABLE II: Low-level features comparison

Feature descriptors	Raw feature accuracies
uniform LBP	40.04%
MLBP	43.75%
HOG	44.12%
SIFT	42.26%
Multi-scale SIFT	44.19%
SIFT-Rank	42.37%
Bio-Inspired-Features	38.72%
LPS	**48.53%**

into 10 subsets of equal size. A single subset is used for validation and the rest 9 subsets are used for training. We gradually increase the L from 16 to 96 with step 8. For each L value, we search for the optimal α value such that the accuracy is maximized at the validation subset. This process is repeated for 10 times, and the optimal (L, α) combination is set based on averaged validation performance. Figure 6 illustrates the validation process. Based on the validation result, we set $L = 64$ and $\alpha = 0.5$ for the result of our paper.

B. Low-level features comparison

In this experiment we compare the effectiveness of the proposed LPS feature descriptor against the popular feature descriptors in face recognition community. Specifically, we use the extracted raw features (before refinement) directly for face recognition. The matching score between probe face and gallery face is defined as:

$$score = \frac{< \vec{x}, \vec{y} >}{\parallel \vec{x} \parallel \parallel \vec{y} \parallel}$$

The comparative results are reported in Table II. The LBP feature descriptor is the original LBP descriptor. The Multi-scale LBP (MLBP) feature descriptor is an extension of LBP, by computing the LBP descriptor at four different sampling radii $\{1, 3, 5, 7\}$. Note that for fairness, we use the same sampling patterns with MLBP for our method. The HOG features are extracted with the suggested settings in this paper [43], where face images are processed at three different scales. The SIFT feature descriptor [23] quantizes both the spatial location and orientation of image gradient within an image patch, and computes a histogram in which each bin corresponds to a combination of specific spatial location and gradient orientation. The SIFT-Rank [40] algorithm is a revised version of SIFT, which uses the ranking of the SIFT values as features.

TABLE III: High-level feature refinement comparison

# Training subjects	LFDA	Proposed
1000	85.64%	84.37%
2000	87.10%	86.25%
3000	88.21%	88.01%
4000	89.15%	89.72%
5000	89.61%	90.13%
6000	89.73%	90.55%
7000	89.84%	91.09%
8000	89.88%	91.82%
9000	90.01%	91.96%
10,000	90.08%	92.11%

In our experiment, the SIFT features are extracted on the same landmark points as our descriptor, with each landmark point giving a 128-dimensional local features corresponding to 4 cells and 8 bins. For fair comparison, we also include the multi-scale version of SIFT with sampling scales $\{1, 3, 5, 7\}$. Bio-Inspired Features (BIF) is a recently developed descriptor that has been successfully applied to face aging community such as face-based human age estimation [6].

From the results listed in Table II we can see that our method has a clear advantage over all the popular descriptors, which confirms the effectiveness of our approach in matching cross-age faces.

C. High-level feature refinement comparison

In this experiment, we investigate the effectiveness of our high-level feature refinement framework by comparing it against the LFDA framework [42]. The LFDA is a state-of-the-art approach to handle the high dimensional feature vector. In the experiment, we fix the testing set as $10,000$ pairs of images, and gradually enlarge the size of training set from $1,000$ to $10,000$ with $1,000$ increment at each step. For fair comparison, we use the same low-level features extracted by LPS algorithm for both frameworks.

We compare the recognition accuracy, the results of which are shown in Table III. These results reveal that as we enlarge the training dataset, the recognition performance of LFDA improves very slowly after a certain point (e.g. $5,000$). The recognition performance of our approach keeps improving, however, up to $10,000$ subjects. This shows that our high-level refinement framework is able to make efficient use of large volume of training samples.

D. Overall benchmark comparison

In the last experiment, we present our benchmark result on the MORPH Album 2 dataset, and compare it against the *state-of-the-art* approaches for aging face recognition. To ensure a fair comparison, all the methods listed in Table IV are tuned to the best settings according to their original papers, and all the methods are using the same training and testing protocol: $10,000$ pairs of faces (with the largest age gap) used for

TABLE IV: Overall benchmark comparison

Algorithms	Recognition Accuracies
FaceVACS [31]	78.90%
Park et al. [18]	79.80%
Du et al. [33]	79.24%
Li et al. [20]	83.90%
Klare et. al. [34]	79.08%
Otto et al. [35]	81.27%
Zhen et. al. [37]	86.12%
Gong et. al. [43]	91.14%
LPS	**92.11%**
HOG+LPS	**94.20%**
LPS+HFA	**94.87%**

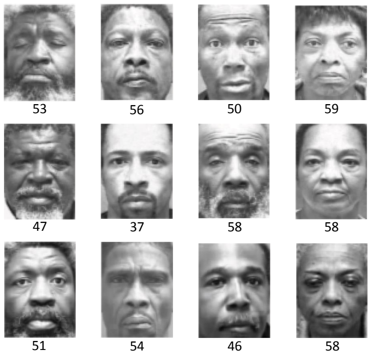

Fig. 7: Examples of failed retrievals for MORPH Album 2 dataset. The first row shows the probe faces, the second row shows the incorrect retrieval results given by our system, and the third row shows the ground-truth faces.

training and the other $10,000$ pairs of faces (with the largest age gap) are reserved for testing.

The comparative results are reported in Table IV. It is encouraging to see that our approach (LPS) outperforms the existing methods in the literature. This result is rather encouraging, given that we are only using a single descriptor (LPS). One thing to note is that different feature descriptors usually provide complementary information that is beneficial to higher accuracy. By combining our descriptor with the HOG descriptor, we can obtain a better result (94.20%). Moreover, by combining our approach with the state-of-the-art approach (HFA [43]), which demonstrates a new *state-of-the-art* (94.87%). Considering the simplicity of our method and the difficulty of this dataset, this is a very encouraging result.

Finally, Figure 7 shows some examples of failed retrievals. We can see that the incorrect retrieved face images appeared highly similar to the probe images.

V. Conclusion

In this paper we present a two-level hierarchical learning model for aging face recognition. At the first level, effective features are extracted by adaptively selecting the local patterns that optimize the common information. At the second level, the output from the first level are further refined using our scalable high-level feature refinement framework to form a final powerful face representation. Extensive comparison experiments based on the MORPH Album 2 dataset reveals a significant improvement over the *state-of-the-art*.

VI. Acknowledgment

This work was supported by grants from Natural Science Foundation of Guangdong Province (2014A030313688), Key Laboratory of Human-Machine Intelligence-Synergy Systems, Chinese Academy of Sciences, National Natural Science Foundation of China (61103164), and Australian Research Council Projects (DP-140102164 and FT-130101457), Shenzhen Basic Research Program (JCYJ20120617114614438), and project MMT-8115038 of the Shun Hing Institute of Advanced Engineering, The Chinese University of Hong Kong.

References

[1] Andreas Lanitis, ”survey of the effects of aging on biometric identity verification,” Int. J. Biometrics, vol. 2, no. 1, pp. 34-52, Dec. 2010.

[2] Narayanan Ramanathan, Rama Chellappa, and Soma Biswas, ”Computational methods for modeling facial aging: A survey, J. Vis. Lang. Comput., vol. 20, no. 3, pp. 131-144, 2009.

[3] Yun Fu and Thomas S. Huang, ”Human age estimation with regression on discriminative aging manifold,” IEEE Transactions on Multimedia, vol. 10, no. 4, pp. 578-584, 2008.

[4] Xin Geng, Zhi-Hua Zhou, and Kate Smith-Miles, ”Automatic age estimation based on facial aging patterns,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 29, no. 12, pp. 2234-2240, 2007.

[5] Guodong Guo, Yun Fu, Charles R. Dyer, and Thomas S. Huang, ”Image-based human age estimation by manifold learning and locally adjusted robust regression,” IEEE Transactions on Image Processing, vol. 17, no. 7, pp. 1178-1188, 2008.

[6] Guodong Guo, Guowang Mu, Yun Fu, and Thomas S. Huang, ”Human age estimation using bio-inspired features,” in CVPR, 2009, pp. 112-119.

[7] Young H. Kwon and Niels Da Vitoria Lobo, ”Age classification from facial images,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition, 1999, pp. 762-767.

[8] A. Lanitis, C. Draganova, and C. Christodoulou, ”Comparing different classifiers for automatic age estimation,” IEEE Trans. Syst., Man, Cybern, vol. 34, no. 1, pp. 621-628, Feb. 2004.

[9] Albert Montillo and Haibin Ling, ”Age regression from faces using random forests,” in ICIP, 2009, pp. 2465-2468.

[10] Narayanan Ramanathan and Rama Chellappa, ”Face verification across age progression,” IEEE Transactions on Image Processing, vol. 15, no. 11, pp. 3349-3361, 2006.

[11] Junyan Wang, Yan Shang, Guangda Su, and Xinggang Lin, ”Age simulation for face recognition,” in ICPR (3), 2006, pp. 913-916.

[12] Shuicheng Yan, Huan Wang, Xiaoou Tang, and Thomas S. Huang, ”Learning auto-structured regressor from uncertain nonnegative labels,” in ICCV, 2007, pp. 1-8.

[13] Shaohua Kevin Zhou, Bogdan Georgescu, Xiang Sean Zhou, and Dorin Comaniciu, ”Image based regression using boosting method,” in ICCV, 2005, pp. 541-548.

[14] Andreas Lanitis, Christopher J. Taylor, and Timothy F. Cootes, ”Toward automatic simulation of aging effects on face images,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 24, no. 4, pp. 442-455, 2002.

[15] Jin-Li Suo, Song Chun Zhu, Shiguang Shan, and Xilin Chen, ”A compositional and dynamic model for face aging,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 32, no. 3, pp. 385-401, 2010.

[16] Jin-Li Suo, Xilin Chen, Shiguang Shan, and Wen Gao, ”Learning long term face aging patterns from partially dense aging databases,” in ICCV, 2009, pp. 622-629.

[17] Norimichi Tsumura, Nobutoshi Ojima, Kayoko Sato, Mitsuhiro Shiraishi, Hideto Shimizu, Hirohide Nabeshima, Syuuichi Akazaki, Kimihiko Hori, and Yoichi Miyake, ”Image based skin color and texture analysis/synthesis by extracting hemoglobin and melanin information in the skin,” ACM Trans. Graph., vol. 22, no. 3, pp. 770-779, 2003.

[18] Unsang Park, Yiying Tong, and Anil K. Jain, ”Age-invariant face recognition,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 32, no. 5, pp. 947-954, 2010.

[19] Haibin Ling, Stefano Soatto, Narayanan Ramanathan, and David W. Jacobs, ”Face verification across age progression using discriminative methods,” IEEE Transactions on Information Forensics and Security, vol. 5, no. 1, pp. 82-91, 2010.

[20] Zhifeng Li, Unsang Park, and Anil K. Jain, ”A discriminative model for age invariant face recognition,” IEEE Transactions on Information Forensics and Security, vol. 6, no. 3-2, pp. 1028-1037, 2011.

[21] Timo Ojala, Matti Pietik ainen, and Topi Maenp, ”Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 24, no. 7, pp. 971-987, 2002.

[22] Krystian Mikolajczyk and Cordelia Schmid, ”A performance evaluation of local descriptors,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 27, no. 10, pp. 1615-1630, 2005.

[23] D. Lowe, ”Distinctive image features from scale-invariant keypoints,” Int’l Journal of Computer Vision, vol. 60, no. 2, pp. 91-110, 2004.

[24] Cong Geng and Xudong Jiang, ”Face recognition using sift features,” in ICIP, 2009, pp. 3313-3316.

[25] Timo Ahonen, Abdenour Hadid, and Matti Pietik ainen, ”Face recognition with local binary patterns,” in ECCV (1), 2004, pp. 469-481.

[26] Timo Ahonen, Abdenour Hadid, and Matti Pietik ainen, ”Face description with local binary patterns: Application to face recognition,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 28, no. 12, pp. 2037-2041, 2006.

[27] Peter N. Belhumeur, Jo a o P. Hespanha, and David J. Kriegman, ”Eigenfaces vs. fisherfaces: Recognition using class specific linear projection,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 19, no. 7, pp. 711-720, 1997.

[28] Kai Zhu, Dihong Gong, Zhifeng Li, and Xiaoou Tang, ”Orthogonal Gaussian Process for Automatic Age Estimation”, in Proc. ACM MM, 2014.

[29] Xiaogang Wang and Xiaoou Tang, ”A unified framework for subspace face recognition,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 26, no. 9, pp. 1222-1228, 2004.

[30] Karl Ricanek Jr. and Tamirat Tesafaye, ”Morph: A longitudinal image database of normal adult age-progression,” in FG, 2006, pp. 341-345.

[31] FaceVACS Software Developer Kit, Cognitec Systems GbmH, http://www.cognitec-systems.de.

[32] Xiaogang Wang, Xiaoout Tang, ”Random sampling LDA for face recognition,” in CVPR, 2004, pp. 259-265.

[33] J. Du, C. Zhai, and Y. Ye, ”Face aging simulation and recognition based on NMF algorithm with sparseness constraints,” Neurocomputing, 2012.

[34] B. Klare and A. K. Jain, ”Face Recognition Across Time Lapse: On Learning Feature Subspaces”, IJCB, Washington, DC, Oct. 11-13, 2011.

[35] C. Otto, H. Han, and A. K. Jain, ”How Does Aging Affect Facial Components”, ECCV WIAF Workshop, Florence, Italy, Oct. 7-13, 2012.

[36] FG-NET Aging Database, http://www.fgnet.rsunit.com/.

[37] L. Zhen and P. Matti, and L. Stan, ”Learning Discriminant Face Descriptor,” PAMI 2013, pp. 289-302.

[38] W. Zhang, X. Wang and X. Tang, ”Coupled information-theoretic encoding for face photo-sketch recognition” CVPR 2011, pp. 513-520.

[39] Shan Caifeng and Gritti Tommaso, ”Learning Discriminative LBP-Histogram Bins for Facial Expression Recognition,” BMVC, 2008.

[40] Toews, M., Wells, W., SIFT-Rank: ”Ordinal description for invariant feature correspondence,” CVPR 2009, pp. 172-177.

[41] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In CVPR (1), pages 886-893, 2005. 4

[42] B. Klare, Z. Li and A. K. Jain, ”Matching forensic sketches to mugshot photos,” IEEE Transactions on Pattern Analysis and Machine Intelligence, 2010.

[43] D. Gong, Z. Li, D. Lin, J. Liu, X. Tang, ”Hidden Factor Analysis for Age Invariant Face Recognition,” ICCV 2013.

[44] B. Chen, C. Chen, and W. H. Hsu, ”Cross-Age Reference Coding for Age-Invariant Face Recognition and Retrieval,” ECCV 2014.

[45] Z. Cao, Q. Yin, X. Tang, J. Sun, ”Face Recognition with Learning-based Descriptor,” CVPR 2010.

[46] Y. Sun, X. Wang, X. Tang, ”Deeply Learned Face Representations are Sparse, Selective, and Robust”, arXiv:1412.1265,2014.

[47] Y. Sun, X. Wang, X. Tang, ”Deep Learning Face Representation from Predicting 10,000 Classes”, CVPR 2014.

[48] H. Dibeklioglu, F. Alnajar, A. Ali Salah, and T. Gevers, ”Combining Facial Dynamics With Appearance for Age Estimation”, IEEE Transactions on Image Processing, 2015.

[49] L. Du and H. Ling, ”Cross-Age Face Verification by Coordinating with Cross-Face Age Verification,” CVPR 2015.

[50] L. Wiskott, J. Fellous, N. Kuiger, and C. Malsberg, ”Face recognition by elastic bunch graph matching,” IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(7): 775-779, 1997.

[51] L. Wolf, T. Hassner, and Y. Taigman, ”Descriptor based methods in the wild,” in Workshop on Faces Real-Life Images at ECCV 2008.

[52] K. Simonyan, O. M. Parkhi, A. Vedaldi, and A. Zisserman, ”Fisher vector faces in the wild,” In Proc. BMVC 2013.

[53] Ling Shao, Li Liu, and Xuelong Li, ”Feature learning for image classification via multiobjective genetic programming,” IEEE Trans. Neural Netw. Learn. Syst. vol. 25, No. 7, pp. 1359-1371, 2014.

[54] Ling Shao, Di Wu, and Xuelong Li, ”Learning deep and wide: A spectral method for learning deep networks,” IEEE Trans. Neural Netw. Learn. Syst. vol. 25, No. 12, pp. 2303-2308, 2014.

[55] Yuelong Li, Li Meng, Jufu Feng, and Jigang Wu, ”Downsampling sparse representation and discriminant information aided occluded face recognition,” SCIENCE CHINA Information Sciences, 2014.

[56] Zhifeng Li, Dihong Gong, Xuelong Li, and Dacheng Tao, ”Learning Compact Feature Descriptor and Adaptive Matching Framework for Face Recognition,” IEEE Transactions on Image Processing (TIP), 2015.

[57] Zhifeng Li, Dihong Gong, Yu Qiao, and Dacheng Tao, ”Common Feature Discriminant Analysis for Matching Infrared Face Images to Optical Face Images,” IEEE Transactions on Image Processing (TIP), 2014.

[58] D. Chen, X. Cao, F. Wen, and J. Sun, ”Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification,” in Proc. CVPR 2013.

[59] X.Qi, R. Xiao, G. Li, Yu Qiao, J. Guo, X. Tang, ”Pairwise Rotation Invariant Co-occurrence Local Binary Pattern ”, IEEE Trans. on Pattern Analysis and Machine Intelligence(T-PAMI), Vol. 36, No. 11, pp. 2199 - 2213, Nov. 2014

[60] Yongqiang Gao, Weilin Huang, and Yu Qiao, ”Local Multi-Grouped Binary Descriptor with Ring-based Pooling Configuration and Optimization”, IEEE Trans. on Image Processing (T-IP) , Vol. 24, No. 12, pp. 4820-4833, 2015

[61] M. Kan, S. shan, H. Zhang, S. Lao, and X. Chen, ”Multi-View Discriminant Analysis,” IEEE Trans. on Pattern Analysis and Machine Intelligence, 2016.

[62] X. Cao, D. Wipf, F. Wen, G. Duan, and J. Sun, ”A practical transfer learning algorithm for face verification,” in Proc. ICCV, 2013.

[63] D. Chen, X. Cao, L. Wang, F. Wen, and J. Sun, ”Bayesian face revisited: A joint formulation,” in Proc. ECCV, 2012.

[64] Zhifeng Li and Xiaoou Tang, ”Using Support Vector Machines to Enhance the Performance of Bayesian Face Recognition,” IEEE Trans. Information Forensics and Security, 2007.

[65] S. Chopra, R. Hadsell, and Y. LeCun, ”Learning a similarity metric discriminatively, with application to face verification,” in Proc. CVPR, 2005.

[66] A. Wagner, J. Wright, A. Ganesh, Z. Zhou, H. Mobahi, and Y. Ma, ”Toward a practical face recognition system: Robust alignment and illumination by sparse representation,” IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012.

[67] Zhifeng Li, Dahua Lin, and Xiaoou Tang, ”Nonparametric Discriminant Analysis for Face Recognition,” IEEE Trans. Pattern Analysis and Machine Intelligence, 2009.

[68] Xiaoou Tang and Zhifeng Li, ”Audio-guided Video Based Face Recognition,” IEEE Trans. Circuits and System for Video Technology. vol. 19, No. 7, pp. 955-964, 2009.

[69] J. Wright, A. Y. Yang, A. Ganesh, S. Sastry, and Y. Ma, ”Robust face recognition via sparse representation,” IEEE Transactions on Pattern Analysis and Machine Intelligence, 2009.

[70] Jiwen Lu, Gang Wang, Weihong Deng, Kui Jia, ”Reconstruction-Based Metric Learning for Unconstrained Face Verification,” IEEE Transactions on Information Forensics and Security, 2015.

[71] Y. Sun, X. Wang, and X. Tang, ”Hybrid Deep Learning for Face Verification,” IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016.