Integration of Multi-Feature Fusion and Dictionary Learning for Face Recognition

Donghui Wang*, Xikui Wang*, Shu Kong*

*College of Computer Science and Technology, Zhejiang University, Hangzhou, 310027, P.R. China

Abstract

Recent research emphasizes more on analyzing multiple features to improve face recognition (FR) performance. One popular scheme is to extend the sparse representation based classification framework with various sparse constraints. Although these methods jointly study multiple features through the constraints, they just process each feature individually such that they overlook the possible high-level relationship among different features. It is reasonable to assume that the low-level features of facial images, such as edge information and smoothed/low-frequency image, can be fused into a more compact and more discriminative representation based on the latent high-level relationship. FR on the fused features is anticipated to produce better performance than that on the original features, since they provide more favorable properties. Focusing on this, we propose two different strategies which start from fusing multiple features and then exploit the dictionary learning (DL) framework for better FR performance. The first strategy is a simple and efficient two-step model, which learns a fusion matrix from training face images to fuse multiple features and then learns class-specific dictionaries based on the fused features. The second one is a more effective model requiring more computational time that learns the fusion matrix and the class-specific dictionaries simultaneously within an iterative optimization procedure. Besides, the second model considers to separate the shared common components from class-specified dictionaries to enhance the discrimination power of the dictionaries. The proposed strategies, which integrate multi-feature fusion process and dictionary learning framework for FR, realize the following goals: (1) exploiting multiple features of face images for better FR performances; (2) learning a fusion matrix to merge the features into a more compact and more discriminative representation; (3) learning class-specific dictionaries with consideration of the common patterns for better classification performance. We perform a series of experiments on public available databases to evaluate our methods, and the experimental results demonstrate the effectiveness of the proposed models.

Keywords: Dictionary Learning, Face Recognition, Multiple Features Fusion, Tensor Decomposition, Sparse Coding

*Corresponding author. Fax: +086-571-87951916, Tel: +086-57187951916
Email addresses: dhwang@zju.edu.cn (Donghui Wang), xkwang@zju.edu.cn (Xikui Wang), aimerykong@zju.edu.cn (Shu Kong)

Preprint submitted to Image and Vision Computing August 1, 2013
1. **Introduction**

With the recent endeavor of computer vision researchers, lots of features have been designed to characterize various aspects of an object. Taking advantage of multiple features can provide more information for face recognition (FR), and the advantages of jointly analyzing multiple features are demonstrated in the literature [1, 2, 3, 4]. Although it is widely believed that recognition performance can benefit from multiple features, in front of the developed multi-feature approaches, it remains an exploratory task to design a more effective and more efficient method to exploit multiple features.

In recent years, several FR methods [5, 6, 7] have been developed based on the dictionary learning (DL) framework, and achieved very promising results. These DL-based FR methods are mainly developed in the following two tracks [8]:

1. directly making the dictionary discriminative, such as learning a class-specified sub-dictionary for each class;
2. making the sparse coefficients discriminative to propagate the discrimination power to the dictionary.

Even though DL-based recognition methods achieve very promising and even state-of-the-art performances, they only work on a single feature type, e.g. the original grayscale facial image or facial outline image, rather than multiple informative features. In other words, they cannot exploit multiple features of one face image and their possible semantic relationships to enhance FR performance.

Aware of the limitations of these DL-based methods that they cannot deal with multiple features, researchers have proposed several methods to tackle this problem [9, 10, 11]. Yuan and Yan propose a multi-task joint sparse representation based classification method (MTJSRC), which treats the recognition with multiple features as a multi-task problem, and each feature type is one task [9]. MTJSRC assumes that the coefficients share the same sparsity pattern among all the features. However, this assumption is too strict and is not held in practice. Therefore, Zhang et al. propose a joint dynamic sparse representation classification method (JDSRC) [10] to address this problem. They argue that the same sparsity pattern is shared among the coefficients at class-level, but not necessarily at atom-level. Yang et al. also address this problem by proposing a relaxed collaborative representation method (RCR), which assumes the sparse codes among different features should be similar in appearance [11].

All above three methods elaborately use multiple features and try to exploit the sparse pattern between the coefficients of different features. Although they produce improved performance, there are still some intrinsic problems:

1. Since the overall dictionary consists of all the features from all training images, when the training data increase in number, the dictionary will become too large that can lower the computational efficiency;
2. Simply taking all features into computation will raise the computational burden and will induce redundant information that does not conduce or even can degrade FR performance;

3. Although different features are connected through the coefficient constraints, these methods neglects the internal relationships among these features which may enhance the FR performance further;

4. The dictionary constituted by all the training data has common components that are shared by different classes, and these components can be interchangeably used for reconstructing the query images, in which way the performance can be compromised.

To address the above problems, we extend our previous work reported in [12, 13] with the proposed two different strategies to integrate multi-feature fusion process and dictionary learning framework. The first one is a two-step model, which first learns a fusion matrix from the training data to fuse different features and then learns class-specific dictionaries. The fusion process exploits the high-level relationship among different features, and fuse these features into a more compact and more discriminative representation. Over the fused features of one specific class, the corresponding dictionary is learned. The second strategy is to learn the fusion matrix and class-specific dictionaries simultaneously. This strategy takes more time but produces better performance. Moreover, in this scheme, we explicitly separate the common components from different classes in the dictionary to make the learned dictionary more compact and more discriminative. As demonstrated by the experimental results, our two strategies both achieve better performances than other closely related methods.

The rest of this paper is organized as follows. In Section 2, we briefly introduce the background and review several approaches that motivate ours. We elaborate the proposed two strategies in Section 3. Extensive experiments on three face recognition datasets are presented in Section 4. Finally, we conclude our paper in Section 5 with discussions.

2. Preliminary

2.1. Tensor Algebra

As we consider to generalize dictionary learning method to multiple features, we turn to the tensor algebra calculation framework. The notations and calculations are follow [16, 17]. High-order tensors are denoted by boldface Euler script letters, e.g. \( \mathbf{X} \). Specially, \( \mathbf{X}_{(n)} \) symbolizes the matrix corresponding to the flattened tensor \( \mathbf{X} \) along the \( n^{th} \) mode. Mathematically, tensor element \( \mathbf{X}_{i_1,i_2,...,i_k} \) of a \( K^{th} \)-order tensor \( \mathbf{X} \in \mathbb{R}^{h_1 \times h_2 \times \cdots \times h_k} \) maps to the element \( (i_k, j) \) of matrix \( \mathbf{X}_{(n)} \), where:

---

1In this work of multiple features fusion, we only focus on the 2D face image databases instead of 2.5D or 3D images [14, 15], as all the compared methods perform on the 2D image databases.
Figure 1: The upper panel shows the $K$ features of a query datum $X$ are approximated by $K$ dictionaries with $K$ sparse coefficients. Three existing methods impose different constraints on the coefficients among the $K$ coefficients: atom-level sparsity [9] shown in (a), group-level sparsity [10] in (b), and overall similarity [11] in (c).

$$j = 1 + \sum_{k=1}^{K} (i_k - 1)J_k \quad \text{with} \quad J_k = \prod_{m=1, m \neq k}^{K-1} I_m$$ (1)

$I_m$ denotes the $m \times m$ identity matrix with 1’s along the diagonal positions and 0’s elsewhere. Moreover, the $n^{th}$ element in a sequence is denoted by a superscript in parentheses, e.g. $X^{(n)}$ is the $n^{th}$ matrix in a sequence. $X^T$ is the transpose of matrix $X$. We assume there are $N$ observations, and each one has $K$ features. Particularly, $X^{(n)} = [x_1^{(n)}, \ldots, x_i^{(n)}, \ldots, x_K^{(n)}] \in \mathbb{R}^{p \times K}$ consists of all $K$ features of the $n^{th}$ sample\(^2\).

The $k$-mode product of a $K^{th}$-order tensor $X \in \mathbb{R}^{I_1 \times \cdots \times I_k \times \cdots I_K}$ by matrix $U \in \mathbb{R}^{J_k \times I_k}$ is expressed as $X \times_k U \in \mathbb{R}^{I_1 \times \cdots \times I_{k-1} \times J_k \times I_{k+1} \times \cdots I_K}$, which can be calculated as:

$$(X \times_k U)_{i_1 \ldots i_{k-1} j_k i_{k+1} \ldots i_K} = \sum_{i_{k} = 1}^{I_k} x_{i_1 i_2 \ldots i_{k-1} i_k u j_k i_{k+1} \ldots i_K}$$ (2)

where the element $x$ and $u$ in $X$ and $U$ are indexed by the corresponding subscripts.

2.2. Related Work

Inspired by the research on human vision, sparse coding based recognition methods have raised a lot of attention. Sparse representation-based classification (SRC) [18] is a far-reaching method under sparse coding theory. It achieves

\(^2\)Without losing generality, in this paper, we assume all the $K$ features have the same length $p$. 

|
very encouraging performances on FR with robustness to illumination changes and occlusions. SRC construct a predefined dictionary $D = [X_1, \ldots, X_c, \ldots, X_C] \in \mathbb{R}^{p \times N_c}$, where $X_c \in \mathbb{R}^{p \times N'}$ is the columnized $N_c$ face images for the $c^{th}$ individual. Given test image $x \in \mathbb{R}^p$, SRC first calculates the sparse code $a$ on the predefined dictionary via $\ell^1$-norm minimization $a = \arg\min_a \|x - Da\|_2^2 + \lambda\|a\|_1$ where $\lambda$ is the scalar controls the sparse degree of $a$, and then assign $x$ to the $c^{th}$ individual such that $c = \arg\min_i \|x - X_i\|_2^2$, where $\delta_i(\cdot)$ is a vector indicator function that extract the elements corresponding to the $i^{th}$ training set $X_i$. Although SRC achieves quite promising performance on FR, it lacks the ability to process multiple features. Moreover, SRC uses all training data as a predefined dictionary therefore the sparse coding procedure is very time-consuming especially as the number of training data increases.

2.2.1. Face recognition with multiple features

One intuitive way to take multiple features into consideration is to go “horizontal”: using one specific dictionary $D_k$ for each $k^{th}$ feature ($k = 1, \ldots, K$). The sub-dictionary $D_k = [X_k^1, \ldots, X_k^c, \ldots, X_k^C]$ contains the $k^{th}$ feature of all training data concatenated horizontally. We name this extension as separate SRC (S-SRC) [10], as demonstrated in the upper panel of Fig. 1. This method constructs each dictionary for each feature independently, and summarizes the reconstruction errors of all features in each class for classification. Another way is to go “vertical”: concatenating all the features vertically into one huge vector as a whole and calculate the reconstruction error of each class for classification. We name this extension as holistic SRC (H-SRC) [18]. Although these two methods use all features for recognition, different features are still computed independently and the relationships between different features are not touched.

To address this problem, researchers propose several models to regularize the sparse codes to reveal the relationships between different features. Yuan and Yan claim the multi-task joint sparse representation classification method (MTJSRC) [9]. MTJSRC treats $K$ features as $K$ different tasks under the multi-task learning framework, and it assumes that these features share the same sparsity pattern at atom-level as demonstrated by Fig. 1 (a). Another method proposed by Zhang et al. is the joint dynamic sparse coding classification (JDSRC) [10]. JDSRC applies a joint dynamic sparsity regularization term to promote joint sparsity pattern shared at group-level while allows distinction at the atom-level, as illustrated by Fig. 1 (b). Yang et al. propose the relaxed collaborative sparse representation (RCR) for FR by using a $\ell^2$ term to reduce the variance of the sparse code $a_k$ as shown by Fig. 1 (c). Rather than atom-level identity in [9] and group-level identity in [10], RCR do not forcing the sparse coefficient to be distinct at atom-level or group-level, but it encourages the sparse coefficient of different features becoming similar. All the three methods take different coding strategies but use a same classification method.

To summarize, given a query image containing $k$ features $X = [x_1, \ldots, x_k, \ldots, x_K] \in \mathbb{R}^{p \times K}$, all the three methods solve the same objective function to get the sparse coefficient $a_k$, but with different sparse pattern regularization $\Phi(\cdot)$:
\[ [a_1, \ldots, a_K] = \arg\min_{a_1, \ldots, a_K} \sum_{k=1}^{K} \|x_k - D_k a_k\|_2^2 + \lambda \Phi(a_1, \ldots, a_K), \tag{3} \]

where \(D_k\) is the dictionary whose columns are the \(k^{th}\) feature vectors of all training images. After calculating the sparse coefficients, all the three methods identify the query image based on the following function:

\[ \text{identity}(X) = \arg\min_c \sum_{k=1}^{K} \|x_k - D_k^c a_k\|_2^2, \tag{4} \]

where \(D_k^c\) is the sub-dictionary consisting of the \(k^{th}\) feature vectors of \(c^{th}\) class face images.

2.2.2. Face recognition with dictionary learning

To circumvent the problem that SRC uses all training data as predefined dictionary which bring too much redundant information and computation burden, Yang et al. propose a Metaface learning method to learn a class-specified dictionary \(D_c [d_c^1, \ldots, d_{K_c}^c] \in \mathbb{R}^{p \times K_c}\) for the \(c^{th}\) individual [19]:

\[ [D_c, A_c] = \arg\min_{D_c, A_c} ||X_c - D_c A_c||_F^2 + \lambda ||A_c||_1, \tag{5} \]

\[ \text{s.t.} \|d_j^c\|_2 = 1, \forall j = 1, \ldots, K_c, \]

where \(\|A_c\|_1\) is defined as the summation of \(\ell^1\)-norm of all the columns of \(A_c = [a_1^c, \ldots, a_{N_c}^c] \in \mathbb{R}^{K_c \times N_c}\), i.e. \(\|A_c\|_1 = \sum_{j=1}^{N_c} \|a_j^c\|_1\). This method concatenates all the sub-dictionaries as an overall dictionary \(D = [D_1, \ldots, D_C]\) for classification which resembles the classification step in SRC.

As pointed out in the literature [7, 20], the learned class-specified dictionaries usually share some common patterns between different classes, which do not help the discrimination of the facial images but are essential for representation of them. This observation also applies to FR well, as the facial images usually share some common patterns, such as illuminations and poses. Considering this, the recent DL method DL-COPAR proposed by Kong and Wang makes a difference between the class-specified features and the common patterns [7]. It learns \(C\) class-specified dictionaries \(D_c\)'s for each of the \(C\) individuals and a common pattern pool \(D_{C+1} \in \mathbb{R}^{p \times K_{C+1}}\). The two parts constitute the overall dictionary \(D = [D_1, \ldots, D_C, D_{C+1}] \in \mathbb{R}^{p \times K}\) in which \(K = \sum_{c=1}^{C+1} K_c\). Separating the common component among class-specified dictionaries helps increase the discrimination power of the class-specified dictionaries and make the
The existing DL-based methods (including DL-COPAR) can be summarized as a general framework in dictionary learning stage:

$$
\min_{D, A} \left\{ f \equiv C(X, D, A) + \lambda \phi(A) + \eta Q(D) \right\},
$$

(6)

In Metaface learning, $C(X, D, A) = \sum_{c=1}^{C} \|X_c - D_c A_c\|_F^2$, $\phi(A) = \sum_{c=1}^{C} \|A_c\|_1$ and $\eta$ is set zero. As for DL-COPAR, the three terms are explained as below:

- \( C() \) measures the reconstruction error of the training data by the dictionary. It ensures the dictionary to have the power of representing any face images, and the images from a specific class (say class \( c \)) can be well approximated by the collaborative effort of the \( c \)th sub-dictionary \( D_c \) and the common pattern pool \( D_{C+1} \). Define a selection operator \( R_c = [r'_1, \ldots, r'_j, \ldots, r'_{K_c}] \in \mathbb{R}^{K_c \times K_c} \) in which:

$$
r'_j = \frac{\sum_{m=1}^{j-1} K_m}{K_c} [0, \ldots, 0, 1, 0, \ldots, 0, 0, \ldots, 0]^T.
$$

Therefore, we have \([D_c, D_{C+1}] = D[R_c, R_{C+1}]\), and \( R_c^T A_c \) selects the coefficient part corresponding to the \( c \)th sub-dictionary and the common pattern pool, i.e. \([D_c, D_{C+1}]\). By denoting \( Q_c = [R_c, R_{C+1}] \), then \( C() \) is defined as:

$$
C(X, D, A) \equiv \sum_{c=1}^{C} \begin{vmatrix}
\|X_c - DA_c\|_F^2 + \\
\|X_c - DQ_c^T A_c\|_F^2
\end{vmatrix}.
$$

(7)

- Denote \( Q_c = [R_1, \ldots, R_{c-1}, R_{c+1}, \ldots, R_C] \), then the term \( \phi(A) \) is defined as:

$$
\phi(A) \equiv \sum_{c=1}^{C} \{\|Q_c^T A_c\|_F^2 + \lambda \|A_c\|_1\}.
$$

It can be seen that \( \phi(A) \) pushes coefficients \( A \) to be sparse with the \( \ell^1 \)-norm penalty and regularizes \( A \) to guarantee that the irrelevant sub-dictionaries \( D_i \)'s (\( i \neq c \) and \( i \neq C + 1 \)) do not contribute to the representation.
of the images from class $c$.

- $Q(D)$ works on the dictionary to drive different sub-dictionaries as incoherent as possible:

$$Q(D) = \sum_{c=1}^{C-1} \sum_{j \neq c}^{C} \|D_j^{c} D_j^{c}\|_F^2.$$

DL-COPAR optimizes the objective function alternatively, and adopts a SRC-like scheme [18] for classification.

3. Our methodology

As discussed in Section 2.2, these SRC-based multiple-feature FR methods focus too much on imposing constraints on coefficients and they ignore the semantic relationship among different features. As discussed in Section 1, there are some drawbacks among these methods. Concentrating on these concerns, we propose two different DL based multiple-feature fusion strategies to improve the FR performance. One is an efficient and simple method, where a core dictionary is learned based on the fused more compact features. The other one separates the common components from class-specifed dictionaries and updates the dictionaries and fusion matrix simultaneously for FR better performance.

3.1. Multiple Features Fusion

Multiple features contain much valuable information which can boost FR performances [2, 3, 4]. However, directly taking face images as dictionary and DL in original feature space will bring redundant information and lower the recognition efficiency. Thus, we assume there is a core dictionary which contains more discriminative information and the performance of FR will not be degraded as training data grow in number. We denote the query image by $X \in \mathbb{R}^{p \times K}$, where $X = [x_1, \ldots, x_k]$ and $x_k$ is the $k^{th}$ feature vectors of the query face image. Given query face image $X$ and a learned dictionary $D \in \mathbb{R}^{p \times D \times K}$ from feature space, we assume there is a transform matrix $W \in \mathbb{R}^{K \times M}$, such that $D = B \times_3 W$, where $B \in \mathbb{R}^{p \times D \times M}$. This means we transform $B$ into $D$ along the third mode through the transformation matrix $W$, as illustrated by Fig. 2. Therefore, with the core dictionary $B$ and the transformation/fusion matrix $W$, we rewrite Eq. 3 to derive the new objective function as below:

$$\{a^1, \ldots, a^K\} = \arg\min_{a^1, \ldots, a^K} \sum_{k=1}^{K} \|x_k - D^k a^k\|_F^2 + \lambda \Phi(a^1, \ldots, a^K),$$

s.t. $D = B \times_3 W$, $D^k$ is the $k^{th}$ slice of $D$ along the third mode.
Here the Lagrange constraint $\Phi(\cdot)$ is imposed on the coefficients corresponding to the $K$ multi-feature dictionaries, such as $\ell^1$-norm penalty [18], group sparsity [10] and overall-similarity term [11].

However, if we base our FR method on Eq. 8, we still need to compute all the features, i.e. this core dictionary does not bring any computation benefit to our new model. In order to solve this problem and explore the feature correlation explicitly, we take an alternative solution. Instead of transforming $K$-feature dictionary into $M$-feature core dictionary, we employ the fusion matrix $W$ directly on query face image $X$. This gives us a compact representation $Y \in \mathbb{R}^{p \times M}$, such that $Y = XW$, where $X \in \mathbb{R}^{p \times K}$. Therefore, we have the following objective function as below:

$$\{a_1^i, \ldots, a_M^i\} = \arg\min_{a_1^i, \ldots, a_M^i} \sum_{m=1}^{M} ||y_m - B^m a^m||_F^2 + \lambda \Phi(a_1^i, \ldots, a_M^i),$$

(9)

where $y_m$ is the $m^{th}$ feature vector of fused query data $Y$, and $B^m \in \mathbb{R}^{p \times D}$ is the $m^{th}$ sub-dictionary of the core dictionary $B \in \mathbb{R}^{p \times D \times M}$. As $M < K$ or $M \ll K$, solving the alternative objective function Eq. 9 is more computationally efficient than solving Eq. 8, and the core dictionary $B$ can be fully exploited.

As previous mentioned [10] uses group information and [11] exploits sparse coefficient similarity, we can use extra information to enhance the recognition performance. However, different from these methods manipulating the sparse coefficients, we choose to apply the label information on the features directly and fuse these features into a new form to obtain better result. We expect the fused features can be more discriminative between classes, therefore maximizing the Fisher criterion [21] is a good choice. We derive the fusion matrix $W$ as below:

$$W = \arg\max_w \frac{\sum_{c=1}^{C} \sum_{i \in J_c} (X_{ti} - \bar{X}_c)W||_F^2}{\sum_{c=1}^{C} \sum_{i \in J_c} (X_{ti} - \bar{X}_c)W||_F^2}$$

(10)

$$= \arg\max_w \left\{ \frac{\text{tr}(W^T \{ \sum_{c=1}^{C} \sum_{i \in J_c} (X_{ti} - \bar{X}_c)^\top (X_{ti} - \bar{X}_c) \} W)}{\text{tr}(W^T \{ \sum_{c=1}^{C} \sum_{i \in J_c} (X_{ti} - \bar{X}_c)^\top (X_{ti} - \bar{X}_c) \} W)} \right\}$$

where $J_c$ is the index set of images from class $c$, $X_{ci} = \frac{1}{N_c} \sum_{i \in J_c} X_i$ is the mean of the $c^{th}$ class, and similarly $X = \frac{1}{N} \sum_{i=1}^{N} X_i$ is the global mean. Let $S_b = \sum_{c=1}^{C} N_c (X_{ci} - \bar{X})^\top (X_{ci} - \bar{X})$ and $S_w = \sum_{i \in J_c} (X_{ti} - \bar{X}_c)^\top (X_{ti} - \bar{X}_c)$.
Thus, solving Eq. 10 to derive $W$ is equivalent to calculating the generalized eigenvalue problem [22]: $S_bw = \lambda S_ww$, for $\lambda \neq 0$. In detail, we have $W = [w_1, \ldots, w_m, \ldots, w_M]$, where $w_m$ is the eigenvector corresponding to the $m^{th}$ largest eigenvalue of $S_b^{-1}S_w$. Note that Eq. 10 is not the same as the classical linear discriminant analysis (LDA) [22], but is a special case of two-dimensional LDA [23] or multilinear discriminant analysis [24] that only deals with the relationship of multi-feature information along the 2nd mode. Moreover, the tensorial application, which is brought in to resolve the multi-feature learning, can alleviate overfitting problem to some extent, especially when the training sample number is limited [25].

Given above definitions, we can learn the core dictionary $B$ and fusion matrix $W$ from the training data directly. Assume we arrange $N_c$ training face images for the $c^{th}$ individual as $X_c = [X_1; \ldots; X_k; \ldots; X_K]$, where $X_k \in \mathbb{R}^{p \times N_c}$. Thus, the compact face images denoted as $Y_c \in \mathbb{R}^{p \times N_c \times M}$ can be obtained from $Y_c = X_c \times_3 W^T$. Based on these fused features, learning core dictionary for the $c^{th}$ individual $B_c$ becomes a typical dictionary learning problem. We can obtain the core dictionary through solving following problem:

$$\min_{B_c = [B_{1c}; \ldots; B_{Mc}]} \sum_{m=1}^{M} \|b_{mj}^w - a_{mj}^w\|^2_F + \lambda \|a_{mj}\|_1 \quad \text{s.t.} \quad \|b_{mj}\|_2 = 1, \forall j = 1, \ldots, K$$

(11)

Fusion matrix $W$ can be obtained directly through solving Eq. 10, and we use K-SVD [27] to learn the core dictionary $B_c$ based on the fused features. We name this method Ours$_{fus}$.  

3.2. Dictionary separation with adaptive fusion

The method proposed in Section 3.1 is very effective and efficient, while we can still make more improvement for better FR performance. By considering the fact that the fusion matrix $W$ only initialized with Fisher discriminant but keep unchanged during the DL process, we can adaptively refine the fusion matrix during the DL process for more discrimination power compatible with the core dictionary. For this purpose, we choose to learn the fusion matrix $W$ with the core dictionary at the same time to refine the representation of face images. It takes more iterations, but higher accuracy is expected. Moreover, as we previously discussed in Section 2.2.2, separating common patterns from class-specified dictionaries can enhance the discriminative representation ability of class-specified dictionaries. Thus, we propose to adaptively learn the fusion matrix with the DL-COPAR framework to achieve higher FR accuracy.

To derive our adaptive fusion method, we first extend the DL-COPAR framework to multiple features. Given the face images of $C$ individuals, we extract $K$ features from each image. Suppose we have learned the class-specified dictionaries for each feature $k$ individually, and a common pattern pool contains the shared components among the
dictionaries, we can easily rewrite the first term Eq. 7 in Eq. 6 to incorporate multiple features:

\[
C(X, D, A) \equiv \sum_{c=1}^{C} \sum_{i \in I_c} \sum_{k=1}^{K} \left\{ \|x^k_i - D^k a^k_i\|_F^2 + \|x^k_i - D^k Q_c^T a^k_i\|_F^2 \right\}.
\]  

(12)

Similarly, in order to increase the incoherency of different class-specified dictionary, we can extend the penalty term as below:

\[
\phi^*(A) \equiv \sum_{c=1}^{C} \sum_{i \in I_c} \|Q_c^T A^k_c\|_F^2 + \lambda \|A^k_c\|_1,
\]

\[
Q(D) \equiv \sum_{c=1}^{C+1} \sum_{j \neq c} \sum_{k=1}^{K} \|D^k_c D^k_j\|_F^2.
\]  

(13)

Now we further add the multiple-feature fusion process to this multiple features DL-COPAR model. As we point out in Eq. 9, we employ the fusion matrix on face image features directly for calculation efficiency. We rewrite Eq. 12:

\[
C^*(X, B, A) \equiv \sum_{c=1}^{C} \sum_{i \in I_c} \sum_{m=1}^{M} \left\{ \|X_m w^m_c - B^m_c a^m_c\|_F^2 + \|X_m w^m_c - B^m_c Q^T_c a^m_c\|_F^2 \right\},
\]  

(14)

where the fusion matrix \(W\) is initialized the same as which defined in Eq. 10. Accordingly, the regularization on the core dictionary \(B\) and sparse coefficient \(A\) are re-defined as:

\[
\phi^*(A) \equiv \sum_{c=1}^{C} \sum_{m=1}^{M} \|Q_c^T A^m_c\|_F^2 + \lambda \|A^m_c\|_1,
\]

\[
Q^*(B) \equiv \sum_{c=1}^{C+1} \sum_{j \neq c} \sum_{m=1}^{M} \|B^{mT}_c B^m_j\|_F^2.
\]  

(15)

In order to learn the fusion matrix \(W\) adaptively, we choose to initialize the \(W\) with Fisher discriminant analysis which define in Eq. 10, and update it under max margin criterion (MMC) [26] to further refine the fusion matrix when learning the core dictionary:
To summarize, we have our final objective function for this model:

\[
f \equiv C'(\mathbf{X}, \mathbf{B}, \mathbf{A}) + \lambda \phi'(\mathbf{A}) + \eta Q'(\mathbf{B}) + \omega \psi'(\mathbf{W}),
\]

where \( C'(\mathbf{X}, \mathbf{B}, \mathbf{A}) \), \( \phi'(\mathbf{A}) \) and \( Q'(\mathbf{B}) \) are defined in Eq. 14, and Eq. 15, respectively. \( \lambda \), \( \eta \) and \( \omega \) are constant scalars to balance the contribution of each term. Besides, after the initialization of \( \mathbf{W} \), we have the fused representation of all the facial images, then we use the K-SVD [27] to initialize the \( M \) dictionaries of the core dictionary \( \mathbf{B} \) and the corresponding coefficients \( \mathbf{A} \). This objective function can be easily optimized alternatively over each parameter. For the updating of \( \mathbf{A} \) and \( \mathbf{B} \), we can split them into \( M \) slices, and solve them with the method provided in DL-COPAR [7]. The updating scheme of \( \mathbf{W} \) can be obtained from the following steps:

1. We denote \( \mathbf{S} = \mathbf{S}_u - \mathbf{S}_b \) and define \( \mathbf{G}_{c}^{1} = \mathbf{B}^{u} \mathbf{a}_{i}^{u} \), \( \mathbf{G}_{c}^{2} = \mathbf{B}^{u} \mathbf{Q} \mathbf{Q}^\top \mathbf{a}_{i}^{u} \), where \( i \in I_c \) for the \( c \)th individual, then we have \( \mathbf{G}_{1}^{c} = [\mathbf{G}_{i1}^{c}; ..., \mathbf{G}_{iM}^{c}] \), and \( \mathbf{G}_{2}^{c} = [\mathbf{G}_{i1}^{c}; ..., \mathbf{G}_{iM}^{c}] \). We select the terms that contain \( \mathbf{W} \) from Eq. 17, and rewrite it as a function of \( \mathbf{W} \):

\[
f(\mathbf{W}) = \sum_{c=1}^{C} \| \mathbf{W}^\top \mathbf{x}_{c(3)} - \mathbf{G}_{1}^{c} \|^2_F + \| \mathbf{W}^\top \mathbf{x}_{c(3)} - \mathbf{G}_{2}^{c} \|^2_F + \omega \text{tr}(\mathbf{W}^\top \mathbf{S} \mathbf{W}).
\]

2. We denote \( \mathbf{\Omega} = [\mathbf{X}_{1(3)}; ..., \mathbf{X}_{C(3)}; ..., \mathbf{X}_{C(3)}] \) and \( \Phi = [\mathbf{G}_{1}^{1(3)}; ..., \mathbf{G}_{1}^{C(3)}; \mathbf{G}_{2}^{1(3)}; ..., \mathbf{G}_{2}^{C(3)}] \), then the Eq. 18 can be further simplified as follow:

\[
f(\mathbf{W}) = \| \mathbf{W}^\top \mathbf{\Omega} - \Phi \|^2_F + \omega \text{tr}(\mathbf{W}^\top \mathbf{S} \mathbf{W}).
\]

3. The Eq. 19 is a quadratic function of \( \mathbf{W} \). The updating scheme can be obtained from the derivative of Eq. 19 respect to \( \mathbf{W} \):

\[
\mathbf{W} = (\mathbf{\Omega} \mathbf{\Omega}^\top + \omega \mathbf{S})^{-1} \mathbf{\Omega} \Phi^\top.
\]

This method not only exploits the relationship between different features, but further separates the common components of the class-specified dictionaries. Compared with the method in Section 3.1, this one obtains more discrim-
inative representation of face images and better recognition performances, with few more iterations. We name this method Ours$_{cop}$.

### 3.3. Classification Scheme

Recently developed methods, e.g. MTJSRC [9], JDSRC [10] and RCR [11], adopt a sparse representation-based classification (SRC) scheme with different regularizations on the coefficients. In detail, given a query image with $K$ features $X \in \mathbb{R}^{p \times K}$, we can first fuse the features with the learned $W \in \mathbb{R}^{K \times M}$ into $Y = XW$, and then encode $Y$ over the core dictionary $B$:

$$A = \arg \min_A \sum_{m=1}^M \|y^m - B^ma^m\|^2_F + \lambda \phi(A),$$

where $\phi(A)$ is a regularization (different from the DL process) on the coefficients $A = [a^1, \ldots, a^M]$. It is $\phi(A)$ that makes these methods different from each other. Concretely, MTJSRC utilizes an $\ell_{2,1}$-norm penalty over the coefficients [9], JDSRC uses group sparsity [10], whereas RCR adopts an overall-similarity term [11]. Note that the order of atoms among the learned dictionaries is crucial for MTJSRC and RCR, as the sparsity pattern relies on the atom order across the dictionaries. Therefore, only the group-sparsity regularization of JDSRC can be directly used in our framework.

In this paper, we also employ a group-level sparse regularizer under the SRC scheme, but using a restricted one by allowing only one class-specified sub-dictionary and the common pattern pool to represent the query image. This constraint enables us to adopt a local sparse coding method for classification [7], i.e. using the individual sub-dictionary and the common pool to reconstruct the facial image:

1. Calculate the reconstruction error over the core dictionary $B$ w.r.t the $c^{th}$ individual for $c = 1, \ldots, C$. Since we have two different strategies, the reconstruction error calculation methods are not the same. For Ours$_{fus}$, the reconstruction error is obtained from:

$$e_c = \min_{a^m} \sum_{m=1}^M \|y^m - B^m a^m\|_2^2 + \lambda_a |a^m|_2^2,$$

as for Ours$_{cop}$, the reconstruction error is calculated by:

$$e_c = \min_{a^m} \sum_{m=1}^M \|y^m - B^m Q_c Q^*_c a^m\|_2^2 + \lambda_a |a^m|_2^2,$$
Figure 3: The ten features of four persons. The left panel shows the features of four persons, on each row, and the right one displays the features of one same person under four different illumination conditions.

Table 1: Different feature extraction parameter values and dictionary sizes for each database.

<table>
<thead>
<tr>
<th>Parameter Value</th>
<th>$\sigma_1$</th>
<th>$\sigma_2$</th>
<th>$\sigma_3$</th>
<th>$\sigma_4$</th>
<th>$\sigma_5$</th>
<th>$\sigma_6$</th>
<th>$\sigma_7$</th>
<th>$\sigma_8$</th>
<th>$\sigma_9$</th>
<th>$D_{\text{fus}}$</th>
<th>$D_{\text{class}}$</th>
<th>$D_{\text{common}}$</th>
<th>$D_{\text{cop}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extended Yale B</td>
<td>0.1</td>
<td>2.0</td>
<td>2.1</td>
<td>2.3</td>
<td>2.4</td>
<td>2.6</td>
<td>2.8</td>
<td>3.0</td>
<td>3.2</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>390</td>
</tr>
<tr>
<td>CMU-PIE</td>
<td>0.1</td>
<td>0.7</td>
<td>1.1</td>
<td>1.5</td>
<td>1.8</td>
<td>2.1</td>
<td>2.5</td>
<td>3.2</td>
<td>4.3</td>
<td>20</td>
<td>12</td>
<td>14</td>
<td>830</td>
</tr>
<tr>
<td>LFW</td>
<td>0.1</td>
<td>0.6</td>
<td>1.3</td>
<td>1.6</td>
<td>2.0</td>
<td>2.3</td>
<td>2.6</td>
<td>3.0</td>
<td>3.2</td>
<td>12</td>
<td>13</td>
<td>15</td>
<td>262</td>
</tr>
</tbody>
</table>

2. Identify the query image to the individual that produces the smallest reconstruction error: $\text{label}(X) = \arg\min_c e_c$.

This classification scheme is just a least square problem [28] with a Tikhonov regularization, thus it is much easier and more efficient to solve than solving the ad hoc sparse problems [9, 10, 11].

4. Experiments

In this section, we evaluate our two methods through a series experiments on three public available datasets: Extended Yale B [29], CMU-PIE [30] and LFW [31]. To fairly demonstrate the effectiveness of our method, we choose some closely related approaches for comparison. These methods include holistic SRC (H-SRC) [18], separate SRC (S-SRC) [10], MTJSRC [9], JDSRC [10] and RCR [11]. H-SRC and S-SRC act as baseline methods, in which H-SRC concatenates all the features into a huge vector, while S-SRC, as an intuitive extension of SRC, separately reconstructs multiple features and then summarizes the reconstruction error of each feature for classification.

4.1. Features and parameters

We extract ten types of features in each image, one is the original gray-scale image, and the other nine are low-level visual features generated from the original image, as illustrated in Fig. 3. We use the LoG filter [32] to extract various edge images as multiple features. The edge images are extracted with different $\sigma_i$, where $i = [1, 2, \ldots, 9]$. According to the Marr-Hildreth theory, intensity edge images at different resolutions are located by convolving the image with series of LoG filters having different spatial frequency parameters [33]. We set the parameter $\sigma_i$ with increasing values to retain the edge information at different level. The filter size is set to $5 \times 5$, and the different $\sigma_i$ for each dataset are summarized in Table 1. The dictionary size tuned to achieve the best performance are also reported in Table 1, where $D_{\text{fus}}$ stands for the number of dictionary atoms for each individual in Ours$fus$, $D_{\text{cop}}$ represents the...
In this subsection, we compare our methods with H-SRC, S-SRC, MTJSRC, JDSRC and RCR in face recognition application on three face databases: Extended Yale B, CMU-PIE and LFW. The different settings of each database are described as below:

- **Extended Yale B [29]**: This database contains 2,414 frontal face images of 38 persons under different illuminations, and about 68 images for each individual. All images are manually aligned, cropped, and resize to $32 \times 32$ in our experiments.

- **CMU-PIE [30]**: The CMU-PIE dataset contains 41,368 face images of 68 individuals. The face images of each person are taken under 13 different poses, 43 different illuminations and 4 different expressions. Here we select 5 frontal poses (C05, C07, C09, C27, C29) of all individuals for experiments, which leaves 170 images for each person [34].

### Table 2: Recognition accuracies varying fusing features number $K$. We fused $K$ features into 3 features for our method.

<table>
<thead>
<tr>
<th>Accuracy</th>
<th>Extended Yale B</th>
<th>CMU-PIE</th>
<th>LFW</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$K = 4$</td>
<td>$K = 10$</td>
<td></td>
</tr>
<tr>
<td>H-SRC</td>
<td>97.06 ± 0.43</td>
<td>97.43 ± 0.17</td>
<td>94.42 ± 0.31</td>
</tr>
<tr>
<td>S-SRC</td>
<td>94.15 ± 0.22</td>
<td>94.84 ± 0.46</td>
<td>92.13 ± 0.21</td>
</tr>
<tr>
<td>MTJSRC</td>
<td>98.27 ± 0.10</td>
<td>98.56 ± 0.42</td>
<td>95.65 ± 0.44</td>
</tr>
<tr>
<td>JDSRC</td>
<td>98.43 ± 0.17</td>
<td>99.05 ± 0.27</td>
<td>96.56 ± 0.28</td>
</tr>
<tr>
<td>RCR</td>
<td>98.13 ± 0.27</td>
<td>98.44 ± 0.30</td>
<td>95.87 ± 0.15</td>
</tr>
<tr>
<td>Ours$_{SOP}$</td>
<td>98.58 ± 0.26</td>
<td>98.87 ± 0.18</td>
<td>97.26 ± 0.54</td>
</tr>
<tr>
<td>Ours$_{cop}$</td>
<td>98.80 ± 0.23</td>
<td>99.65 ± 0.12</td>
<td>97.68 ± 0.14</td>
</tr>
</tbody>
</table>

Overall dictionary size in Ours$_{cop}$, and $D_{class}$ and $D_{common}$ denote the number of atoms that constitute each individual-specific dictionary and the common feature pool ($D_{cop} = \#individual \times D_{class} + D_{common}$). In our experiment, we do not make any differences on the features, but setting equal weight to them. Note that, MTJSRC only uses two types of features in [9], whereas it performs better on the ten features in this paper than that in [9]; JDSRC manually selects the face regions for multi-region FR in [10], yet we run it on the ten features to report the results; RCR also considers sophisticatedly segmenting the face images for multi-region FR, but we use RCR for multi-feature FR to report its performance.

In the objective function Eq. 17 of Ours$_{SOP}$, we have three parameters to balance the influence of each term. For $\lambda$, we set it to make approximate 10% elements of the coefficients nonzero. $\eta$ decides the incoherence of the individual-specific dictionaries, and we set it to be 1, the same as that in [7]. We set $\omega = 0.1$ to keep the discrimination between the fused features.

### 4.2. Face Recognition

In this subsection, we compare our methods with H-SRC, S-SRC, MTJSRC, JDSRC and RCR in face recognition
Figure 4: Recognition performance of varying fusing feature number on different dataset. (a) Extended Yale B; (b) CMU-PIE; (c) LFW.

- **LFW [31]**: This dataset is a large scale dataset designed for unconstrained face recognition under different poses, illumination, expressions and alignment. Here we choose the alignment version LFW-a [35]. As many individuals have very few images, we gather those who has more than 40 images to form a subset containing 19 individuals. Each image is cropped to 150 × 150 and resized to 30 × 30 pixel-resolution.

For each dataset, we randomly select half face images of each individual for training dictionary and the rest for testing, that is about 32 in Extended Yale B, 85 in CMU-PIE and at least 20 in LFW training face images for each person.

In our FR experiments, we randomly select $K$ features from the ten features to fuse into $M$ more compact and more discriminative features for our two methods. In Ours$_{fas}$, we learn $D_{fas}$-atom dictionary for every class $c$. In Ours$_{cop}$ we learn $D_{class}$-atom dictionary for each class and $D_{common}$-atom for common pattern representation. For comparison, we range $K$ from 4 to 10, with fused feature number $M = 3$. The dictionary size $D_{fas}$, $D_{class}$, $D_{common}$, and $D_{cop}$ will affect the recognition performance. We will discuss its effect later. The optimized dictionary size for each dataset is listed in Table 1. For H-SRC, we concatenate all features into a single huge vector, and run FR in the SRC fashion. For S-SRC, we calculate the reconstruction error of each feature individually, and summarize them up for FR. For each $K$, we run each method 10 times on each dataset and report the mean accuracies with standard deviations of each method given $K = 4$ and $K = 10$ on each dataset in Table 2. The corresponding figures of all results on each database are shown in Fig. 4(a), Fig 4(b) and Fig 4(c) for Extended Yale B, CMU-PIE and LFW respectively.

As showed by the results, H-SRC achieves good performance, while S-SRC derives no better accuracy. This is because H-SRC and S-SRC blindly use multiple features without utilizing the relationships among them. MTJSRC, JDSRC and RCR clearly improve the results over H-SRC and S-SRC, owing to their reasonable structured constraints on the sparse coefficients, which bridge the multiple features to enhance recognition performance. Both of our propose methods achieve better performances than the other methods. The fusion model we combined in Ours$_{fas}$ has better performances than those non-fusion methods since it take advantage of the more compact and discriminative data.
The adaptive model we propose in Ours\textsubscript{cop} achieves further improvement. It not only takes advantage of the fusion model, but also adaptively refines the fusion matrix with the core dictionary learning. This procedure spends more learning time than Ours\textsubscript{fus}, but gets higher recognition accuracy, as shown in Table 2.

In Fig. 5, we display the three fused features of the five individuals that are previously showed in Fig. 3. The first row illustrates the original facial images, and the following three rows correspond to the three fused features. From this figure, especially from the second and third row, we can see the fused features are robust to illumination changes. Even though there are shadows caused by various illumination angles, these shadows are alleviated in the fused features that provide the resilience to illumination changes. Additionally, from the second row, i.e. the first fused feature, we can see the trivial pixels that bring no discrimination power in the face are smoothed, and the important patterns, such as eyes and nose, are highlighted. We think it is the pixel-level spatial correspondence of the features represented by tensor attributes to this phenomenon, thus intuitively demonstrating why the fused features can produce more decent FR performances.

### 4.3. FR under different fused dimension

In this section, we inspect the effect of the fused feature number $M$ on FR performance. We select half of the images from each individual for training and the other half for testing, and vary $M$ from 1 to 6. Other parameters are set as described in Section 4.2. The results are listed in Table 3.

![Figure 5: The three fused features of different persons. The top row shows the original facial images of five people that are previously illustrated in Fig. 3. The left part shows the fused features of four individuals on each column, and the right one displays that of the same individual under four different illumination conditions.](image-url)
As we can see from the result, the recognition performance is relatively lower when $M$ is small. This is because the new features are too compact to be more discriminative than the originals, when fusing the multiple features into too few dimensions. As the fused feature number $M$ increasing, the performances of both methods are growing. This means the fusion procedure indeed compacts the features and makes them become more discriminative. But when $M$ reaches some thresholds, the FR performance will saturate. It is easy to explain that larger $M$ means more discriminative fused features are generated. As the regularization term and initialization of the fusion matrix $W$ are motivated by Fisher criterion [22] and MMC [26], this phenomenon resembles Fisher LDA that the classification becomes more accurate when the reduced dimension increases in a reasonable range.

### 4.4. FR under different training number

In this section, we discuss how the number of training samples affect FR performance. For comparison, we design three different training sample number settings $S_1, S_2$ and $S_3$ for each dataset. For Extended Yale B, we set $[S_1, S_2, S_3] = [10, 20, 30]$. For CMU-PIE, we set $[S_1, S_2, S_3] = [60, 90, 110]$. For LFW, we use $[S_1, S_2, S_3] = [5, 15, 20]$. The other parameters are set to same values as in section 4.2. The recognition result is listed in Table 4. The recognition performance grows as the training number increases, since there are more discriminative information combined in the dictionary. Moreover, when the training sample number is small, the performance of our method is much better than the others, which verifies the effectiveness of multilinear learning to alleviate small sample size problem, as illustrated in [25].

### 4.5. Efficiency comparison

As previously discussed in section 3, our methods are more efficient on face recognition compared with other methods. Here we provide the face recognition experiment on CMU-PIE dataset to illustrate this. We choose half of the face images of each individual to constitute the training set and the rest for the testing set. Each set contains 5777 face images. For our methods, we take 20 dictionary atoms for each individual in Our$_{fus}$, and use 12 dictionary atoms
for each individual, 14 common dictionary atoms in Ours$^{cop}$, where the parameter setting is the same as that in section 3.4.1. The visualized result is shown in Fig. 6.

As we can see, both of our methods are much more efficient than other compared methods. Since we use the simplest sparse coding method, and decrease the dictionary size with dictionary learning, the testing procedure only requires very small time expense. However, the MTJSRC, RCR and JDSRC spend much more time, since they include all training samples into computation and the constraints are much more complex than ours. The H-SRC and S-SRC methods have medium time cost, since they use simple constraints but compute on all training samples.

4.6. FR under different dictionary size

In the DL-COPAR framework, more class-specified dictionary atoms contain more discriminative information, and larger common pattern pool captures more shared components between class-specified dictionaries. However, the FR performance will not continually grow as the both sizes keep increasing. So, it is necessary to find an optimized balance between class-specified dictionary and common pattern pool size for Ours$^{cop}$. In this section, we carry out the experiment on dataset Extended Yale B for discussion. We range class-specified dictionary size $D_{\text{class}}$ and the common pattern pool size $D_{\text{common}}$ from 5 to 15. The results of both methods are shown in Fig. 7.

As we can see from Fig. 7, the FR accuracy increases as the class-specified dictionary size $D_{\text{class}}$ grows, when the common pattern pool size $D_{\text{common}}$ fixed. However, the accuracy becomes stable as the $D_{\text{class}}$ exceeds certain value, since the class-specified dictionary is sufficient for the discriminative data representation. Moreover, $D_{\text{class}}$ has much more effect on recognition performance than $D_{\text{common}}$. The reason is that class-specified dictionary contains the most discriminative information which is utilized for recognition, however the common pattern pool is designed to separate the shared components in the class-specified dictionaries. When the class-specified dictionaries contain insufficient discriminative information, the recognition performance will be deteriorated, even with large common pattern pool.

Meanwhile, the recognition performance will decrease as $D_{\text{common}}$ keeps increasing. This is because the common pattern pool is too large to contain only the common patterns shared by different classes, but also the discriminative ones owned by their corresponding class-specified dictionaries.
5. Conclusion and Future Work

In this paper, we discuss how to exploit multiple features for better FR performance. We demonstrate that popular sparse coding based methods only put effort on how to constraint sparse coefficients to connect different features, and the SRC based sparse coding scheme is time-consuming, when facing large-scale situations. To address these problems, we propose two different strategies. The first one is to learn a fusion matrix based on Fisher criterion from the training data to fuse the different features to a more discriminative and compact representation, and then use DL framework to learn a core dictionary on the new features for FR. The second strategy is to learn the fusion matrix and the core dictionary simultaneously, and use MMC to refine the fusion matrix \( W \). Moreover, we extend the DL-COPAR framework to deal with multiple features and derive more discriminative class-specified dictionaries through separating the common components among them. Compared with the first strategy, the second one takes more iterations but generate better FR performance. To evaluate the effectiveness of our methods, we compare them with several sparse coding based approaches on three popular datasets. The results demonstrate that the proposed methods achieve better performances than others. Although these two methods have very promising performances, we still have some problems to overcome in the future. In our methods, we only consider the global features and assume these different features have the same length, in which way, other features cannot be exploited in our models. Therefore, in our future work we consider to extend our method to other effective features with different length for better FR performance, \textit{i.e.} LBP [36], SIFT [37].

Acknowledgment

This work is supported by Natural Science Foundations of China (No.61071218) and 973 Program (Project No.2010CB327904).

References


[34] D. Cai, X. He, Y. Hu, J. Han, T. Huang, Learning a spatially smooth subspace for face recognition, in: CVPR, 2007.

