Signal Processing 128 (2016) 28–39

Contents lists available at ScienceDirect

Signal Processing

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

Efficient Fisher Discrimination Dictionary Learning

Rui Jiang [a], Hong Qiao [a,b,*], Bo Zhang [c]

[a] State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
[b] CAS Center for Excellence in Brain Science and Intelligence Technology (CEBSIT), Chinese Academy of Sciences, Shanghai 200031, China
[c] LSEC and Institute of Applied Mathematics, AMSS, Chinese Academy of Sciences, Beijing 100190, China

ARTICLE INFO

Article history:
Received 17 July 2015
Received in revised form
15 December 2015
Accepted 22 March 2016
Available online 25 March 2016

Keywords:
Fisher discrimination dictionary learning
Nesterov's accelerated gradient method
Face recognition
Domain adaptation

ABSTRACT

Fisher Determination Dictionary Learning (FDDL) has shown to be effective in image classification. However, the Original FDDL (O-FDDL) method is time-consuming. To address this issue, a fast Simplified FDDL (S-FDDL) method was proposed. But S-FDDL ignores the role of collaborative reconstruction, thus having an unstable performance in classification tasks with unbalanced changes in different classes. This paper focuses on developing an Efficient FDDL (E-FDDL) method, which is more suitable for such classification problems. Precisely, instead of solving the original Fisher Discrimination based Sparse Representation (FDSR) problem, we propose to solve an Approximate FDSR (A-FDSR) problem whose objective function is an upper bound of that of FDSR. A-FDSR considers the role of both the discriminative reconstruction and the collaborative reconstruction. This makes E-FDDL stable when dealing with classification tasks with unbalanced changes in different classes. Furthermore, fast optimization strategies are applicable to A-FDSR, thus leading to the high efficiency of E-FDDL which can be explained by analysis on convergence rate and computational complexity. We also use E-FDDL to accelerate the Shared Domain-adapted Dictionary Learning (SDDL) algorithm which is a FDDL based new method for domain adaptation. Experimental results on face and object recognition demonstrate the stable and fast performance of E-FDDL.

1. Introduction

Fisher Discrimination Dictionary Learning (FDDL), which is an interesting variant of Sparse Representation Classifier (SRC) [1,2], was proposed recently in [3,4]. The success of FDDL can be ascribed to three key ideas. The first one is *discriminative reconstruction*. Different from those Dictionary Learning (DL) methods for learning a shared dictionary (see, e.g., [5–14]), FDDL learns a dictionary composed of class-specific sub-dictionaries. Each sub-dictionary is encouraged to well reconstruct the corresponding training examples, but poorly reconstruct the others. Thus, the class-wise reconstruction errors can be used for classification. The second one is *collaborative reconstruction*, which means that, the reconstruction of each training example should be performed collaboratively over the whole dictionary. This idea distinguishes FDDL from those DL methods for learning a dictionary for each class independently (see, e.g., [15–20]). And the third one is *discriminative representation*. This idea implies that the representation coefficients of the training examples should have a small within-class variance and a large between-class scatter. Thus, the representation coefficients can be exploited in classification. FDDL is an essentially supervised DL method. In this category, many methods have been proposed. Here, we review some state-of-the-art methods. Discriminative KSVD (D-KSVD) [12] is a supervised DL method designed for face recognition. Label Consistent KSVD (LC-KSVD) [13,14] is an extension of D-KSVD. The Shared Domain-adapted DL (SDDL) method [21,22] is a FDDL based supervised DL method for domain adaptation. For more information of this class of DL methods, see a recent survey [23].

The tradeoff among the three ideas indeed leads to a good performance of FDDL. However, the original FDDL model is complicated, and the derived DL method, i.e., O-FDDL, is often time-consuming. To address this issue, a simplified FDDL model was presented in [4], which is obtained from the original FDDL model under the assumption that each training example can only be reconstructed by columns in its corresponding sub-dictionary. The simplified model has much fewer variables, so the S-FDDL method is much faster than the O-FDDL method. Nevertheless, S-FDDL is a class-by-class DL method, which means that, in the learning process of S-FDDL, only the discriminative reconstruction and the discriminative representation are considered, but the idea of collaborative reconstruction is ignored. In [4], the equivalence of S-FDDL and O-FDDL is empirically investigated in various image classification problems. Based on the experimental results, it was

[*] Corresponding author at: State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China.
E-mail addresses: jiangrui627@163.com (R. Jiang), hong.qiao@ia.ac.cn (H. Qiao), b.zhang@amt.ac.cn (B. Zhang).

http://dx.doi.org/10.1016/j.sigpro.2016.03.013
0165-1684/© 2016 Elsevier B.V. All rights reserved.

concluded in [4] that, for the image classification problems in which data have similar variations in different classes, the classification accuracy of S-FDDL is often close to that of O-FDDL, so S-FDDL can serve as an efficient FDDL in these tasks; however, for the classification tasks involving data with unbalanced variations in different classes, such as some face recognition tasks in which the changes in illumination, accessory, expression, pose or view are often non-uniform, the classification accuracy of S-FDDL is always worse than that of O-FDDL due to the ignorance of the collaborative construction. For more details about this, see Section 6.1.1 of [4].

In this paper, we develop an Efficient FDDL (E-FDDL) method, which is particularly suitable for the classification tasks involving data with unbalanced variations in different classes. To do this, we first notice that the O-FDDL method is an iterative optimization process to alternatively solve two problems until convergence: the Fisher Discrimination based Sparse Representation (FDSR) problem and the Dictionary Update (DU) problem. The DU strategy of O-FDDL used in [3,4] is very fast, but the optimization procedure of FDSR suffers from a great deal of execution time. Our E-FDDL addresses this issue by solving an Approximate FDSR (A-FDSR) problem whose objective function is an upper bound of that of the original FDSR problem in the O-FDDL method. A-FDSR has two advantages. Firstly, A-FDSR considers all the three key ideas of FDDL. In image classification tasks involving data with unbalanced variations in different classes, this property ensures that the E-FDDL method has a better and more stable performance compared with the S-FDDL method. Secondly, A-FDSR can be split into several subproblems, and the dual problem of each subproblem is smooth, strongly convex and has fewer variables than the primal problem. This makes it possible to apply fast optimization strategies, such as Nesterov's accelerated gradient method [24], to the dual problems of these subproblems, thus effectively accelerating O-FDDL. To explain this more clearly, we analyze and compare the convergence rates and computational complexities of the key steps in solving FDSR and A-FDSR, respectively. In the experimental section, the stability and efficiency of E-FDDL are verified in face recognition tasks on two popular databases.

In addition, we evaluate the performance of the E-FDDL method in domain adaptation applications. The SDDL method, which is a FDDL based discriminative DL method, was proposed recently in [21,22] and has been proved to be effective in object recognition tasks involving data from multiple visual domains. We use our E-FDDL algorithm to replace the O-FDDL algorithm in the original SDDL method, thus obtaining a more efficient version of SDDL which is called Efficient SDDL (E-SDDL) in this paper. Object recognition experiments on two real-world databases involving four different domains were conducted to show that E-SDDL keeps the good recognition accuracy of SDDL but is much faster than SDDL. Obviously, this superiority owes much to the stability and efficiency of the proposed E-FDDL.

The remaining part of this paper is organized as follows. We give a brief review of the FDDL and FDDL-based SDDL methods in Section 2. In Section 3, we present the details of the proposed E-FDDL and E-SDDL methods. The experimental results on face and object recognition are presented in Sections 4.1 and 4.2, respectively. The final section concludes this paper.

2. A review of Fisher Discrimination Dictionary Learning

In this section, we first briefly review the original and simplified models of FDDL, and their corresponding DL methods. Then we give a review on the recently proposed domain-adaptive discriminative DL method called SDDL, which can be viewed as a modification of FDDL for domain adaptation applications.

2.1. The O-FDDL model and the O-FDDL method

Given the training examples $\boldsymbol{Y} = [\boldsymbol{Y}_1, \boldsymbol{Y}_2, \dots, \boldsymbol{Y}_C] \in \mathbb{R}^{n \times N}$, where C is the number of classes (known, fixed), n is the dimensionality of these N training examples, $\boldsymbol{Y}_j \in \mathbb{R}^{n \times N_j}$ is a matrix composed of N_j training examples with class label j. Let the desired over-complete dictionary be $\boldsymbol{D} = [\boldsymbol{D}_1, \boldsymbol{D}_2, \dots, \boldsymbol{D}_C] \in \mathbb{R}^{n \times K}$ with $n < K$, where K is the number of columns in the whole dictionary, $\boldsymbol{D}_j \in \mathbb{R}^{n \times K_j}$ is the sub-dictionary associated with class j and K_j is the number of columns in this sub-dictionary. We denote the sparse representation matrix of $\boldsymbol{Y}$ over $\boldsymbol{D}$ by $\boldsymbol{X} = [\boldsymbol{X}_1, \boldsymbol{X}_2, \dots, \boldsymbol{X}_C] \in \mathbb{R}^{K \times N}$, where each $\boldsymbol{X}_j \in \mathbb{R}^{K \times N_j}$ can be written as $\boldsymbol{X}_j = [\boldsymbol{X}_j^1; \boldsymbol{X}_j^2; \cdots; \boldsymbol{X}_j^C]$ to satisfy $\boldsymbol{D}\boldsymbol{X}_j = \sum_{k=1}^{C} \boldsymbol{D}_k \boldsymbol{X}_j^k$. Hereafter, by convention, the concatenation of two matrices (including vectors) will be written as $[\boldsymbol{A}_1, \boldsymbol{A}_2] \doteq [\boldsymbol{A}_1\ \boldsymbol{A}_2]$ and $[\boldsymbol{A}_1; \boldsymbol{A}_2] \doteq \left[\begin{smallmatrix}\boldsymbol{A}_1\\ \boldsymbol{A}_2\end{smallmatrix}\right]$. The Original FDDL (O-FDDL) model is

$$\begin{aligned} &\min_{\boldsymbol{X},\boldsymbol{D}} \quad R_{\mathrm{O}}(\boldsymbol{Y}, \boldsymbol{D}, \boldsymbol{X}) + \lambda_2 f(\boldsymbol{X}) + \lambda_1 \| \boldsymbol{X} \|_1 \\ &\text{s. t.} \quad \| \boldsymbol{d}_l \|_2 = 1 \quad (l = 1, \dots, K), \end{aligned} \tag{1}$$

where

$$R_{\mathrm{O}}(\boldsymbol{Y}, \boldsymbol{D}, \boldsymbol{X}) = \sum_{j=1}^{C} \left(\| \boldsymbol{Y}_j - \boldsymbol{D}\boldsymbol{X}_j \|_{\mathrm{F}}^2 + \| \boldsymbol{Y}_j - \boldsymbol{D}_j \boldsymbol{X}_j^j \|_{\mathrm{F}}^2 + \sum_{k \neq j} \| \boldsymbol{D}_k \boldsymbol{X}_j^k \|_{\mathrm{F}}^2 \right)$$

and $f(\boldsymbol{X}) = \mathrm{Tr}(S_{\mathrm{W}}(\boldsymbol{X}) - S_{\mathrm{B}}(\boldsymbol{X})) + \eta \| \boldsymbol{X} \|_{\mathrm{F}}^2$. Here $S_{\mathrm{W}}(\boldsymbol{X})$ is the within-class scatter and $S_{\mathrm{B}}(\boldsymbol{X})$ is the between-class scatter, λ_1 and λ_2 are the sparsity regularization parameter and the regularization parameter associated with f, respectively.

In (1), minimizing the term $\| \boldsymbol{Y}_j - \boldsymbol{D}_j \boldsymbol{X}_j^j \|_{\mathrm{F}}^2 + \sum_{k \neq j} \| \boldsymbol{D}_k \boldsymbol{X}_j^k \|_{\mathrm{F}}^2$ emphasizes the principle that the learned dictionary $\boldsymbol{D}$, which is a concatenation of class-specific sub-dictionaries $\boldsymbol{D}_j$ with $j = 1, 2, \dots, C$, should represent $\boldsymbol{Y}_j$ discriminatively. And minimizing the term $\| \boldsymbol{Y}_j - \boldsymbol{D}\boldsymbol{X}_j \|_{\mathrm{F}}^2$ accurately reflects the idea that the whole dictionary $\boldsymbol{D}$ should also represent $\boldsymbol{Y}_j$ collaboratively. Besides, the Fisher Discrimination Criterion (FDC) is applied to strength the discriminativeness of the representation coefficients $\boldsymbol{X}$. Specifically, minimizing the trace difference form of FDC, i.e., $\mathrm{Tr}(S_{\mathrm{W}}(\boldsymbol{X}) - S_{\mathrm{B}}(\boldsymbol{X}))$, is adopted. And adding the term $\eta \| \boldsymbol{X} \|_{\mathrm{F}}^2$ can make $f(\boldsymbol{X})$ convex.

Algorithm 1. The O-FDDL Method.

```
Input: Training set $\boldsymbol{Y} \in \mathbb{R}^{n \times N} (n < N)$, initial over-complete dictionary $\boldsymbol{D}^{(0)} \in \mathbb{R}^{n \times K} (n < K)$, initial sparse representation matrix $\boldsymbol{X}^{(0)} \in \mathbb{R}^{K \times N}$, $\lambda_1 > 0$, $\lambda_2 > 0$, $\eta > 0$, the threshold value $\varepsilon > 0$ for solving problem (2), the number of iterations $T$ for O-FDDL.
1: Initialization: $t$:=0, $\boldsymbol{X}$:=$\boldsymbol{X}^{(0)}$, $\boldsymbol{D}$:=$\boldsymbol{D}^{(0)}$.
2: Repeat
3:    Update $\boldsymbol{X}$: Letting $\boldsymbol{D} = \boldsymbol{D}^{(t)}$ and computing $\boldsymbol{X}^{(t+1)}$ by solving the Fisher Discrimination based Sparse Representation (FDSR) problem:
      $\min_{\boldsymbol{X}} R_{\mathrm{O}}(\boldsymbol{Y}, \boldsymbol{D}^{(t)}, \boldsymbol{X}) + \lambda_2 f(\boldsymbol{X}) + \lambda_1 \| \boldsymbol{X} \|_1.$   (2)
4:    Update $\boldsymbol{D}$: Fixing $\boldsymbol{X} = \boldsymbol{X}^{(t+1)}$ and computing $\boldsymbol{D}^{(t+1)}$ by solving the Dictionary Update (DU) problem:
      $\min_{\boldsymbol{D}} R_{\mathrm{O}}(\boldsymbol{Y}, \boldsymbol{D}, \boldsymbol{X}^{(t+1)}) \quad \text{s. t.} \quad \| \boldsymbol{d}_l \|_2 = 1 \ (l = 1, \dots, K).$   (3)
5:    $t$:=$t$ + 1.
```

```
6: Quit If $t=T$.
Output: $\boldsymbol{D}^{(t)}$ and $\boldsymbol{X}^{(t)}$.
```

The O-FDDL model (1) is non-convex. Thus, the O-FDDL method, proposed in [3,4], is an alternating optimization procedure, as shown in Algorithm 1, to split model (1) into the two problems (2) and (3). Though the DU problem (3) is non-convex, an efficient DU strategy has been presented in [3,4]. In this paper, we focus on the optimization issue of the FDSR problem (2), and meanwhile, use the same DU strategy as in [3,4].

2.2. Current optimization strategies for the FDSR problem

In the O-FDDL method, the FDSR problem (2) is solved class by class at each iteration, that is, each $\boldsymbol{X}_j$ is updated individually while keeping $\overline{\boldsymbol{X}}_j = \boldsymbol{X} \backslash \boldsymbol{X}_j$ fixed. Thus, the FDSR problem (2) can be split into C subproblems:

$$\min_{\boldsymbol{X}_j} R_{\mathrm{O}}(\boldsymbol{Y}_j, \boldsymbol{D}^{(t)}, \boldsymbol{X}_j) + \lambda_2 f(\boldsymbol{X}_j) + \lambda_1 \parallel \boldsymbol{X}_j \parallel_1, \tag{4}$$

where

$$R_{\mathrm{O}}(\boldsymbol{Y}_j, \boldsymbol{D}^{(t)}, \boldsymbol{X}_j) = \parallel \boldsymbol{Y}_j - \boldsymbol{D}^{(t)}\boldsymbol{X}_j \parallel_{\mathrm{F}}^2 + \parallel \boldsymbol{Y}_j - \boldsymbol{D}_j^{(t)}\boldsymbol{X}_j^j \parallel_{\mathrm{F}}^2 + \sum_{k \neq j} \parallel \boldsymbol{D}_k^{(t)}\boldsymbol{X}_j^k \parallel_{\mathrm{F}}^2,$$

and

$$f(\boldsymbol{X}_j) = [\eta - 1 + (N_j/N)] \parallel \boldsymbol{X}_j \parallel_{\mathrm{F}}^2 + [2 - (N_j/N)]\mathrm{Tr}(\boldsymbol{X}_j(\boldsymbol{I} - \boldsymbol{W}_j)\boldsymbol{X}_j^{\top}) + 2\mathrm{Tr}(\boldsymbol{X}_j\boldsymbol{B}_j\overline{\boldsymbol{X}}_j^{\top})$$

with $\boldsymbol{W}_j = (1/N_j)\mathbf{1}_{N_j}\mathbf{1}_{N_j}^{\top}$ and $\boldsymbol{B}_j = (1/N)\mathbf{1}_{N_j}\mathbf{1}_{N-N_j}^{\top}$. Here, we denote by $\mathbf{1}_d$ the d-dimensional column vector of all ones.

Let $\widetilde{\boldsymbol{D}}^{(t)}$ be a block diagonal matrix with main diagonal square matrices $\boldsymbol{D}_1^{(t)}, \boldsymbol{D}_2^{(t)}, \ldots, \boldsymbol{D}_C^{(t)}$, i.e.,

$$\widetilde{\boldsymbol{D}}^{(t)} = \mathrm{diag}(\boldsymbol{D}_1^{(t)}, \boldsymbol{D}_2^{(t)}, \ldots, \boldsymbol{D}_C^{(t)}).$$

Then the objective function of the subproblem (4) is always strongly convex with the parameter

$$\mu = 2\lambda_{\min}(\widetilde{\boldsymbol{D}}^{(t)\top}\widetilde{\boldsymbol{D}}^{(t)}) + 2\lambda_2[\eta - 1 + (N_j/N)] \tag{5}$$

in the case when $\lambda_2 > 0$ and $\eta > 1 - (N_j/N)$, and the gradient of $R_{\mathrm{O}} + \lambda_2 f$ is Lipschitz continuous with the Lipschitz constant

$$L = 2\lambda_{\max}(\boldsymbol{D}^{(t)\top}\boldsymbol{D}^{(t)} + \widetilde{\boldsymbol{D}}^{(t)\top}\widetilde{\boldsymbol{D}}^{(t)}) + 2\lambda_2(\eta + 1). \tag{6}$$

Here, the minimum eigenvalue and the maximum eigenvalue of a square matrix $\boldsymbol{A}$ are denoted by $\lambda_{\min}(\boldsymbol{A})$ and $\lambda_{\max}(\boldsymbol{A})$, respectively. In the O-FDDL method, the Iterative Projection Method (IPM) [25] is applied to solve the subproblem (4). Specifically, the derivatives of R_{O} and f w.r.t. $\boldsymbol{X}_j$ can be derived as

$$\nabla R_{\mathrm{O}}(\boldsymbol{X}_j) = 2(\boldsymbol{D}^{(t)\top}\boldsymbol{D}^{(t)} + \widetilde{\boldsymbol{D}}^{(t)\top}\widetilde{\boldsymbol{D}}^{(t)})\boldsymbol{X}_j - 2\boldsymbol{D}^{(t)\top}\boldsymbol{Y}_j - 2[\mathbf{0}; \boldsymbol{D}_j^{(t)\top}\boldsymbol{Y}_j; \mathbf{0}]$$

and

$$\nabla f(\boldsymbol{X}_j) = 2(\eta + 1)\boldsymbol{X}_j - 4\boldsymbol{m}_j\mathbf{1}_{N_j}^{\top} + 2\boldsymbol{m}\mathbf{1}_{N_j}^{\top},$$

where $\boldsymbol{m}_j$ is the mean vector of $\boldsymbol{X}_j$, and $\boldsymbol{m}$ is the mean vector of $[\boldsymbol{X}_j, \overline{\boldsymbol{X}}_j]$. Let h be the iteration counter. IPM updates $\boldsymbol{X}_j$ by

$$\boldsymbol{X}_j^{(h+1)} = \mathcal{S}_{\lambda_1/\sigma}\left(\boldsymbol{X}_j^{(h)} - \frac{1}{\sigma}\left[\nabla R_{\mathrm{O}}(\boldsymbol{X}_j^{(h)}) + \lambda_2 \nabla f(\boldsymbol{X}_j^{(h)})\right]\right)$$

with $\sigma = (L + \mu)/2$. Hereafter, we use $\mathcal{S}_\nu$ to denote the component-wise soft thresholding operator with threshold ν, i.e., $\mathcal{S}_\nu(a) = \mathrm{sign}(a)\max\{|a| - \nu, 0\}$.

In [4], it was also recommended to use the Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) [26] to solve the subproblem (4). Let $t^{(0)} = 1$ and $\boldsymbol{Z}_j^{(0)} = \boldsymbol{X}_j^{(0)}$. Then FISTA performs the update rules as follows:

$$t^{(h)} = \left(1 + \sqrt{1 + 4t^{(h-1)^2}}\right)/2,$$

$$\boldsymbol{Z}_j^{(h)} = \boldsymbol{X}_j^{(h)} + \frac{t^{(h-1)} - 1}{t^{(h)}}(\boldsymbol{X}_j^{(h)} - \boldsymbol{X}_j^{(h-1)}),$$

$$\boldsymbol{X}_j^{(h+1)} = \mathcal{S}_{\lambda_1/L}\left(\boldsymbol{Z}_j^{(h)} - \frac{1}{L}[\nabla R_{\mathrm{O}}(\boldsymbol{Z}_j^{(h)}) + \lambda_2 \nabla f(\boldsymbol{Z}_j^{(h)})]\right).$$

In this paper, We also use Nesterov's Accelerated Proximal Gradient (NAPG) method [24] to solve (4). Let $\boldsymbol{Z}_j^{(0)} = \boldsymbol{X}_j^{(0)}$. Then the update steps are

$$\boldsymbol{Z}_j^{(h)} = \boldsymbol{X}_j^{(h)} + \frac{\sqrt{L} - \sqrt{\mu}}{\sqrt{L} + \sqrt{\mu}}(\boldsymbol{X}_j^{(h)} - \boldsymbol{X}_j^{(h-1)}),$$

$$\boldsymbol{X}_j^{(h+1)} = \mathcal{S}_{\lambda_1/L}\left(\boldsymbol{Z}_j^{(h)} - \frac{1}{L}[\nabla R_{\mathrm{O}}(\boldsymbol{Z}_j^{(h)}) + \lambda_2 \nabla f(\boldsymbol{Z}_j^{(h)})]\right).$$

When $\lambda_2 > 0$ and $\eta = 1$, the objective function of (4) is strongly convex. Then the subproblem (4) can be solved by IPM with a linear convergence rate $\mathcal{O}(\exp((-4\mu h)/(L + \mu)))$ [24]. NAPG has been proved to further accelerate IPM with a linear convergence rate $\mathcal{O}(\exp((-\sqrt{\mu}h)/\sqrt{L}))$ [24]. From the update rules of FISTA, we can see that FISTA does not rely on the strongly convex parameter μ and its convergence rate is $\mathcal{O}(L/[2(h + 1)^2])$ [26]. The complexity of computing L and μ are $\mathcal{O}(K^2 n) + \mathcal{O}(K^3)$ and $\sum_{j=1}^{C}(\mathcal{O}(K_j^2 n) + \mathcal{O}(K_j^3))$, respectively, and the complexity of each update step of IPM, FISTA and NAPG for (4) is $\mathcal{O}(K^2 N_j)$.

2.3. The S-FDDL model and the S-FDDL method

FDSR has NK variables, so its optimization procedure is time-consuming. By assuming that each training example can only be reconstructed by columns in its corresponding sub-dictionary, i.e., $\boldsymbol{X}_j^k = \mathbf{0}$ when $k \neq j$, the O-FDDL model (1) can be reduced into a much simpler problem:

$$\begin{aligned} \min_{\boldsymbol{X},\boldsymbol{D}} \quad & R_{\mathrm{S}}(\boldsymbol{Y}, \boldsymbol{D}, \boldsymbol{X}) + \lambda_2 f_{\mathrm{S}}(\boldsymbol{X}) + \lambda_1 \parallel \boldsymbol{X} \parallel_1 \\ \text{s. t.} \quad & \parallel \boldsymbol{d}_l \parallel_2 = 1 \quad (l = 1, \ldots, K), \end{aligned} \tag{7}$$

where

$$R_{\mathrm{S}}(\boldsymbol{Y}, \boldsymbol{D}, \boldsymbol{X}) = \sum_{j=1}^{C} 2 \parallel \boldsymbol{Y}_j - \boldsymbol{D}_j\boldsymbol{X}_j^j \parallel_{\mathrm{F}}^2$$

and

$$f_{\mathrm{S}}(\boldsymbol{X}) = \sum_{j=1}^{C} [2 - (N_j/N)]\mathrm{Tr}(\boldsymbol{X}_j^j(\boldsymbol{I} - \boldsymbol{W}_j)\boldsymbol{X}_j^{j\top}).$$

Obviously, this assumption helps reduce the number of variables of the FDSR problem (2) from NK to $\sum_{j=1}^{C} N_j K_j$, which makes the S-FDDL method very fast. However, from the formulations of $R_{\mathrm{S}}(\boldsymbol{Y}, \boldsymbol{D}, \boldsymbol{X})$ and $f_{\mathrm{S}}(\boldsymbol{X})$, it is seen that the S-FDDL model (8) can be further reduced to C independent class-specific sub-dictionary learning models, which means that, only the discriminative reconstruction and the discriminative representation are considered in the S-FDDL method, but the collaborative reconstruction is ignored. In [4], the experimental results for various image classification tasks show that this ignorance will affect the classification accuracy of S-FDDL for tasks involving data with non-uniform variations in different classes.

2.4. The SDDL model and the SDDL method

In [21,22], the O-FDDL model (1) has been modified to solve domain shift problem in which the test examples from the target domain have a different distribution with most of the training examples from the source domain. The derived DL model is referred to as the Shared Domain-adapted Dictionary Leaning (SDDL) model. Here, we only consider a simple case. In this case, only one source domain is considered, that is, we have the training data from two domains, $\bar{\boldsymbol{Y}}_S \in \mathbb{R}^{n_S \times N_S}$ from the source domain and $\bar{\boldsymbol{Y}}_T \in \mathbb{R}^{n_T \times N_T}$ from the target domain. The SDDL model aims at jointly learning two projectors $\boldsymbol{P}_S \in \mathbb{R}^{n \times n_S}$, $\boldsymbol{P}_T \in \mathbb{R}^{n \times n_T}$ and a common discriminative dictionary $\boldsymbol{D} = [\boldsymbol{D}_1, \boldsymbol{D}_2, \dots, \boldsymbol{D}_C] = [\boldsymbol{d}_1, \boldsymbol{d}_2, \dots, \boldsymbol{d}_K] \in \mathbb{R}^{n \times K}$. Let $\tilde{\boldsymbol{P}} = [\boldsymbol{P}_S, \boldsymbol{P}_T]$, $\tilde{\boldsymbol{Y}} = \operatorname{diag}(\bar{\boldsymbol{Y}}_S, \bar{\boldsymbol{Y}}_T) = [\tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2, \dots, \tilde{\boldsymbol{y}}_{(N_S+N_T)}]$ and the representation matrix $\tilde{\boldsymbol{X}} = [\boldsymbol{X}_S, \boldsymbol{X}_T]$. The SDDL model can be cast as the following optimization problem:

$$\min_{\boldsymbol{D}, \tilde{\boldsymbol{P}}, \tilde{\boldsymbol{X}}} \quad \tilde{R}_O(\boldsymbol{D}, \tilde{\boldsymbol{P}}, \tilde{\boldsymbol{X}}) + \nu \tilde{r}(\tilde{\boldsymbol{P}}) + \lambda_1 \| \boldsymbol{X} \|_1$$

$$\text{s. t.} \quad \boldsymbol{P}_S \boldsymbol{P}_S^\top = \boldsymbol{I}, \quad \boldsymbol{P}_T \boldsymbol{P}_T^\top = \boldsymbol{I}, \quad \| \boldsymbol{d}_l \|_2 = 1 \ \ (l = 1, \dots, K), \tag{8}$$

where

$$\tilde{R}_O(\boldsymbol{D}, \tilde{\boldsymbol{P}}, \tilde{\boldsymbol{X}}) = \| \tilde{\boldsymbol{P}}\tilde{\boldsymbol{Y}} - \boldsymbol{D}\tilde{\boldsymbol{X}} \|_F^2 + \tau_1 \| \tilde{\boldsymbol{P}}\tilde{\boldsymbol{Y}} - \boldsymbol{D}\tilde{\boldsymbol{X}}_{\text{in}} \|_F^2 + \tau_2 \| \tilde{\boldsymbol{P}}\tilde{\boldsymbol{Y}} - \boldsymbol{D}\tilde{\boldsymbol{X}}_{\text{out}} \|_F^2 \tag{9}$$

with

$$\tilde{\boldsymbol{X}}_{\text{in}}[i, j] = \begin{cases} \tilde{\boldsymbol{X}}[i, j] & \boldsymbol{d}_i, \tilde{\boldsymbol{y}}_j \text{ are in the same class} \\ 0 & \text{otherwise} \end{cases}$$

and

$$\tilde{\boldsymbol{X}}_{\text{out}}[i, j] = \begin{cases} \tilde{\boldsymbol{X}}[i, j] & \boldsymbol{d}_i, \tilde{\boldsymbol{y}}_j \text{ are in different classes} \\ 0 & \text{otherwise,} \end{cases}$$

and the second term $\tilde{r}(\tilde{\boldsymbol{P}}) = -\operatorname{Tr}((\tilde{\boldsymbol{P}}\tilde{\boldsymbol{Y}})(\tilde{\boldsymbol{P}}\tilde{\boldsymbol{Y}})^\top)$ is a regularization term on $\tilde{\boldsymbol{P}}$ to preserve the information from the original domains. From the formulations of $\tilde{R}_O$ and $\tilde{r}$, it can be observed that, when $\tilde{\boldsymbol{P}}$ is fixed, after ordering $\tilde{\boldsymbol{P}}\tilde{\boldsymbol{Y}}$ and $\tilde{\boldsymbol{X}}$ as $\boldsymbol{Y} = [\boldsymbol{Y}_1, \boldsymbol{Y}_2, \dots, \boldsymbol{Y}_C]$ and $\boldsymbol{X} = [\boldsymbol{X}_1, \boldsymbol{X}_2, \dots, \boldsymbol{X}_C] = [\boldsymbol{X}_j^1; \boldsymbol{X}_j^2; \cdots; \boldsymbol{X}_j^C]$, $\tilde{R}_O$ actually becomes similar with the R_O term of the O-FDDL model, $\tilde{r}$ becomes a constant that can be ignored, so we have obtained the modified O-FDDL model

$$\min_{\boldsymbol{D}, \boldsymbol{X}} \quad \sum_{j=1}^{C} \left(\| \boldsymbol{Y}_j - \boldsymbol{D}\boldsymbol{X}_j \|_F^2 + \tau_1 \| \boldsymbol{Y}_j - \boldsymbol{D}_j \boldsymbol{X}_j^j \|_F^2 + \tau_2 \sum_{k \neq j} \| \boldsymbol{D}_k \boldsymbol{X}_j^k \|_F^2 \right) + \lambda_1 \| \boldsymbol{X} \|_1$$

$$\text{s. t.} \quad \| \boldsymbol{d}_l \|_2 = 1 \quad (l = 1, \dots, K). \tag{10}$$

Based on this observation, in [21,22], the SDDL method is designed as an optimization procedure with following two alternating steps:

- first fix $\boldsymbol{D}$ and $\boldsymbol{X}$ and update the projectors $\tilde{\boldsymbol{P}}$ by using the Stiefel manifold optimization technique [27],
- then fix $\tilde{\boldsymbol{P}}$ and update the dictionary $\boldsymbol{D}$ and the representation matrix $\boldsymbol{X}$ by using the O-FDDL method.

The kernelized version of SDDL was also proposed in [21,22]. In each version of SDDL, the O-FDDL method serves as an important component. This means that a stable and fast version of O-FDDL can definitively improve the computational efficiency of the SDDL method.

3. Efficient Fisher Discrimination Dictionary Learning

In this section, we propose an Efficient FDDL (E-FDDL) method which is much faster than O-FDDL, and meanwhile, gives more accurate and stable classification results than S-FDDL for data with non-uniform variations in different classes. In our E-FDDL method, instead of solving the original FDSR problem (2), we consider an approximate FDSR problem which can be solved by an efficient optimization strategy. Further, by replacing O-FDDL in the original SDDL procedure with E-FDDL, we obtain an efficient SDDL method called E-SDDL.

3.1. The approximate problem of FDSR (A-FDSR)

In the objective function of the original FDSR problem (2), $R_O(\boldsymbol{Y}, \boldsymbol{D}^{(t)}, \boldsymbol{X})$ accurately reflects the principle of collaborative and discriminative reconstructions. Noting that

$$\| \boldsymbol{D}_k^{(t)} \boldsymbol{X}_j^k \|_F^2 \leq \| \boldsymbol{D}_k^{(t)} \|_2^2 \| \boldsymbol{X}_j^k \|_F^2, \tag{11}$$

where $k = 1, \dots, C$, we see that an upper bound of $R_O(\boldsymbol{Y}, \boldsymbol{D}^{(t)}, \boldsymbol{X})$ is

$$R_E(\boldsymbol{Y}, \boldsymbol{D}^{(t)}, \boldsymbol{X}) = \sum_{j=1}^{C} \left(\| \boldsymbol{Y}_j - \boldsymbol{D}^{(t)}\boldsymbol{X}_j \|_F^2 + \| \boldsymbol{Y}_j \|_F^2 - 2\operatorname{Tr}(\boldsymbol{Y}_j^\top \boldsymbol{D}_j^{(t)} \boldsymbol{X}_j^j) + \| \boldsymbol{D}_j^{(t)} \|_2^2 \| \boldsymbol{X}_j^j \|_F^2 + \sum_{k \neq j} \| \boldsymbol{D}_k^{(t)} \|_2^2 \| \boldsymbol{X}_j^k \|_F^2 \right),$$

which also reflects the principle of collaborative and discriminative reconstructions. We now rewrite R_E as

$$R_E(\boldsymbol{Y}, \boldsymbol{D}^{(t)}, \boldsymbol{X}) = \sum_{i=1}^{N} \left(\| \boldsymbol{y}_i - \boldsymbol{D}^{(t)}\boldsymbol{x}_i \|_2^2 + \| \boldsymbol{y}_i \|_2^2 - 2\boldsymbol{y}_i^\top \boldsymbol{D}_{c_i}^{(t)} \boldsymbol{x}_i^{c_i} + \| \boldsymbol{D}_{c_i}^{(t)} \|_2^2 \| \boldsymbol{x}_i^{c_i} \|_2^2 + \sum_{j \neq c_i} \| \boldsymbol{D}_j^{(t)} \|_2^2 \| \boldsymbol{x}_i^j \|_2^2 \right), \tag{12}$$

where $\boldsymbol{y}_i$ is a training example with class label c_i, and $\boldsymbol{x}_i$ is its representation coefficients. From the minimization of (12) w.r.t. $\boldsymbol{X}$, it can be seen that minimizing the term $\| \boldsymbol{y}_i - \boldsymbol{D}^{(t)}\boldsymbol{x}_i \|_2^2$ reflects the idea of collaborative reconstruction, and $\| \boldsymbol{y}_i \|_2^2$ is a constant that can be ignored. Minimizing the third term of the right-hand side of (12) means that $\boldsymbol{D}_{c_i}^{(t)} \boldsymbol{x}_i^{c_i}$ is enforced to be as close to $\boldsymbol{y}_i$ as possible and minimizing the fourth term $\| \boldsymbol{D}_{c_i}^{(t)} \|_2^2 \| \boldsymbol{x}_i^{c_i} \|_2^2$ is used to bound the increment of its length. Similarly, minimizing the fifth term $\| \boldsymbol{D}_j^{(t)} \|_2^2 \| \boldsymbol{x}_i^j \|_2^2$ is used to restrict the ability of each $\boldsymbol{D}_j^{(t)} (j \neq c_i)$ in reconstructing $\boldsymbol{y}_i$. Thus R_E also takes the discriminative reconstruction into consideration. This ensures the E-FDDL method to perform well in classification tasks with unbalanced variations in different classes.

Using R_E to replace R_O in the objective function of the original FDSR problem (2) leads to the following problem:

$$\min_{\boldsymbol{X}} R_E(\boldsymbol{Y}, \boldsymbol{D}^{(t)}, \boldsymbol{X}) + \lambda_2 f(\boldsymbol{X}) + \lambda_1 \| \boldsymbol{X} \|_1. \tag{13}$$

This problem is referred to as the Approximate FDSR (A-FDSR) problem. Our E-FDDL method, presented in Algorithm 2, will be based on solving (13) instead of (2).

Algorithm 2. The E-FDDL Method.

```
Input: Training set $\boldsymbol{Y} \in \mathbb{R}^{n\times N} (n < N)$, initial over-complete dictionary $\boldsymbol{D}^{(0)} \in \mathbb{R}^{n\times K} (n < K)$, initial sparse representation matrix $\boldsymbol{X}^{(0)} \in \mathbb{R}^{K\times N}$, $\lambda_1 > 0$, $\lambda_2 > 0$, $\eta > 0$, the threshold value $\varepsilon > 0$ for solving problem (13), the number of iterations $T$ for E-FDDL.
Initialization: $t$:=0, $\boldsymbol{X}$:=$\boldsymbol{X}^{(0)}$, $\boldsymbol{D}$:=$\boldsymbol{D}^{(0)}$.
Repeat
    Update $\boldsymbol{X}$: Letting $\boldsymbol{D} = \boldsymbol{D}^{(t)}$ and computing $\boldsymbol{X}^{(t+1)}$ by solving the A-FDSR problem (13).
    Update $\boldsymbol{D}$: Fixing $\boldsymbol{X} = \boldsymbol{X}^{(t+1)}$ and computing $\boldsymbol{D}^{(t+1)}$ by solving the DU problem (3).
    $t$:=$t + 1$.
Quit If $t$=$T$.
Output $\boldsymbol{D}^{(t)}$ and $\boldsymbol{X}^{(t)}$.
```

3.2. An efficient optimization strategy for A-FDSR

The A-FDSR problem (13) can be split into N subproblems, that is, updating each $\boldsymbol{x}_i$ individually while holding $\boldsymbol{x}_k$ $(k \neq i)$ fixed. Let $\gamma_i = \lambda_2[(4/N_{c_i})\sum_{k\neq i, c_k \neq c_i}\boldsymbol{x}_k - (2/N)\sum_{k\neq i}\boldsymbol{x}_k]$. We have the formulation of each subproblem

$$\min_{\boldsymbol{x}_i} \ \| \boldsymbol{y}_i - \boldsymbol{D}^{(t)}\boldsymbol{x}_i \|_2^2 + \sum_{j=1}^{C} (\alpha_i^j \| \boldsymbol{x}_i^j \|_2^2 - \boldsymbol{\beta}_i^{j\top}\boldsymbol{x}_i^j + \lambda_1 \| \boldsymbol{x}_i^j \|_1), \tag{14}$$

where

$$\alpha_i^j = \| \boldsymbol{D}_j^{(t)} \|_2^2 + \lambda_2[\eta + 1 - (2/N_j) + (1/N)],$$

$\boldsymbol{\beta}_i^{c_i} = \gamma_i^{c_i} + 2\boldsymbol{D}_{c_i}^{(t)\top}\boldsymbol{y}_i$ and $\boldsymbol{\beta}_i^j = \gamma_i^j$ for $j \neq c_i$. To facilitate the analysis, we drop the subscript i and obtain that

$$\min_{\boldsymbol{x}} \ \| \boldsymbol{y} - \boldsymbol{D}^{(t)}\boldsymbol{x} \|_2^2 + \sum_{j=1}^{C} (\alpha^j \| \boldsymbol{x}^j \|_2^2 - \boldsymbol{\beta}^{j\top}\boldsymbol{x}^j + \lambda_1 \| \boldsymbol{x}^j \|_1). \tag{15}$$

We now derive a dual problem of the subproblem (15). To this end, let $R(\boldsymbol{z}) = \| \boldsymbol{y} - \boldsymbol{z} \|_2^2$ and $r_j(\boldsymbol{x}^j) = \alpha^j \| \boldsymbol{x}^j \|_2^2 - \boldsymbol{\beta}^{j\top}\boldsymbol{x}^j + \lambda_1 \| \boldsymbol{x}^j \|_1$. Then the dual problem of (15) is given as

$$\min_{\boldsymbol{\mu}} F_{\rm D}(\boldsymbol{\mu}) = R^*(-\boldsymbol{\mu}) + \sum_{j=1}^{C} r_j^*(\boldsymbol{u}^j), \tag{16}$$

where $R^*(-\boldsymbol{\mu}) = (1/4)\boldsymbol{\mu}^\top\boldsymbol{\mu} - \boldsymbol{y}^\top\boldsymbol{\mu}$ is the conjugate function of R, $\boldsymbol{u} = \boldsymbol{D}^{(t)\top}\boldsymbol{\mu}$, and

$$r_j^*(\boldsymbol{u}^j) = [1/(4\alpha^j)] \sum_{k=1}^{K_j} [\max\{|u_k^j + \beta_k^j| - \lambda_1, 0\}]^2$$

is the conjugate function of r_j. When $\alpha^j > 0$, r_j is strongly convex [28]. This implies the differentiability of r_j^* and that $\nabla r_j^*(\boldsymbol{u}^j) = [1/(2\alpha^j)]\mathcal{S}_{\lambda_1}(\boldsymbol{u}^j + \boldsymbol{\beta}^j)$ is Lipschitz continuous with the Lipschitz constant $L(\nabla r_j^*) = 1/(2\alpha^j)$. The derivation method in [28] can be easily generalized to derive the formulations of $R^*(-\boldsymbol{\mu})$, $r_j^*(\boldsymbol{u}^j)$ and $\nabla r_j^*(\boldsymbol{u}^j)$.

The dual problem (16) has some good properties. First, the number of variables is n which is less than K, the number of the primal variables. Second, $F_{\rm D}$ is strongly convex with the constant $\mu(F_{\rm D}) = 0.5$. Third, the gradient of $F_{\rm D}$, i.e.,

$$\nabla F_{\rm D}(\boldsymbol{\mu}) = 0.5\boldsymbol{\mu} - \boldsymbol{y} + \boldsymbol{D}[\nabla r_1^*(\boldsymbol{u}^1); \nabla r_2^*(\boldsymbol{u}^2); \cdots; \nabla r_C^*(\boldsymbol{u}^C)]$$

is Lipschitz continuous with the Lipschitz constant

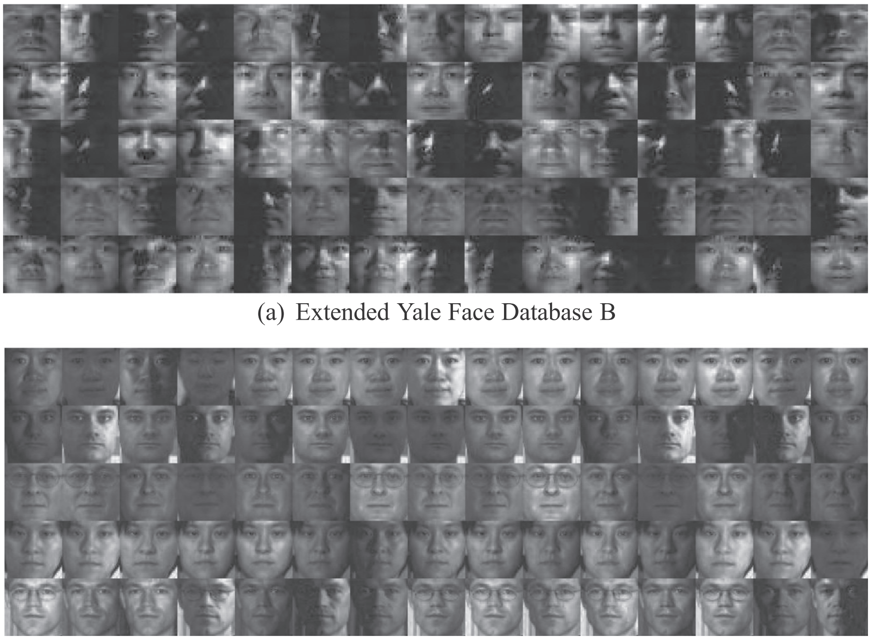
$$L(\nabla F_{\rm D}) = 0.5 + \max\{L(\nabla r_1^*(\boldsymbol{u}^1)), \ldots, L(\nabla r_C^*(\boldsymbol{u}^C))\} \| \boldsymbol{D}^{(t)} \|_2^2.$$

In view of these properties, we can employ Nesterov's Accelerated Gradient method to solve the problem (16). This strategy starts at $\boldsymbol{\zeta}^{(0)} = \boldsymbol{\mu}^{(0)}$ and iterates as follows:

$$\boldsymbol{\zeta}^{(k)} = \boldsymbol{\mu}^{(k-1)} - \frac{1}{L(\nabla F_{\rm D})}\nabla F_{\rm D}(\boldsymbol{\mu}^{(k-1)}),$$

$$\boldsymbol{\mu}^{(k)} = \boldsymbol{\zeta}^{(k)} + \frac{\sqrt{L(\nabla F_{\rm D})} - \sqrt{\mu(F_{\rm D})}}{\sqrt{L(\nabla F_{\rm D})} + \sqrt{\mu(F_{\rm D})}}(\boldsymbol{\zeta}^{(k)} - \boldsymbol{\zeta}^{(k-1)}).$$

This strategy is called NADGA in our previous work [28]. Actually, when $\lambda_2 > 0$, NADGA can also be used to solve the FDSR problem



(a) Extended Yale Face Database B

(b) CMU PIE Face Database

Fig. 1. Part of training examples from (a) and (b).

(2). For the reason why we did not do this, see Appendix A.

3.3. Convergence rate and computational complexity

The subproblem (16) can be solved by NADGA with a linear convergence rate $O(\exp(-\sqrt{\mu(F_{\mathrm{D}})}(h-1)/\sqrt{L(\nabla F_{\mathrm{D}})}))$ which is better than that of NAPG for solving (4), i.e., $O(\exp(-\sqrt{\mu}(h-1)/\sqrt{L}))$. This follows easily from Proposition 3.1 which is proved in Appendix B.

Propostion 3.1. *L/μ is always larger than $L(\nabla F_{\mathrm{D}})/\mu(F_{\mathrm{D}})$.*

Further, the complexity of computing $L(\nabla F_{\mathrm{D}})$ is $O(Kn^2)+O(n^3)+\sum_j(O(K_j^2 n)+O(K_j^3))$, and the complexity of each update step of NADGA for (16) is $O(Kn)$. Due to the fact that $K>n$, compared with the computational complexity of the key steps in NAPG for (4) (see Section 2.2), the complexity of the key steps in NADGA for (16) is much lower. This demonstrates that E-FDDL is more efficient than O-FDDL, as also illustrated in the experimental results.

3.4. The E-SDDL method

Since E-FDDL is much more efficient than O-FDDL, it is natural to use E-FDDL to replace O-FDDL in the SDDL algorithm. This leads to a more efficient SDDL method called E-SDDL. From the modified O-FDDL model (10), it is seen that E-SDDL involves repeatedly solving the modified A-FDSR problem

$$\min_{\boldsymbol{X}} \tilde{R}_{\mathrm{E}}(\boldsymbol{Y},\boldsymbol{D}^{(t)},\boldsymbol{X})+\lambda_1\|\boldsymbol{X}\|_1, \tag{17}$$

where

$$\tilde{R}_{\mathrm{E}}(\boldsymbol{Y},\boldsymbol{D}^{(t)},\boldsymbol{X})=\sum_{i=1}^{N}\Bigg(\|\boldsymbol{y}_i-\boldsymbol{D}^{(t)}\boldsymbol{x}_i\|_2^2+\tau_1\|\boldsymbol{y}_i\|_2^2-2\tau_1\boldsymbol{y}_i^{\mathrm{T}}\boldsymbol{D}_{c_i}^{(t)}\boldsymbol{x}_i^{c_i}+\tau_1\|\boldsymbol{D}_{c_i}^{(t)}\|_2^2\|\boldsymbol{x}_i^{c_i}\|_2^2 \\ +\tau_2\sum_{j\neq c_i}\|\boldsymbol{D}_j^{(t)}\|_2^2\|\boldsymbol{x}_i^{j}\|_2^2\Bigg). \tag{18}$$

Accordingly, in the subproblem (14), we have $\alpha_i^{c_i}=\tau_1\|\boldsymbol{D}_{c_i}^{(t)}\|_2^2$, $\boldsymbol{\beta}_i^{c_i}=2\tau_1\boldsymbol{D}_{c_i}^{(t)\mathrm{T}}\boldsymbol{y}_i$, and for $j\neq c_i$, $\alpha_i^j=\tau_2\|\boldsymbol{D}_j^{(t)}\|_2^2$, $\boldsymbol{\beta}_i^j=\boldsymbol{0}$. Thus, we have a special case of the subproblem (14), so NADGA is applicable.

Table 1
Recognition results on Extended Yale Face Database B (The best accuracy is marked in boldface.)

DL Method (Opt. strategy for FDSR or A-FDSR)	$K_j=N_j=15$ for O-FDDL and E-FDDL		$K_j=N_j=20$ for O-FDDL and E-FDDL	
	Accuracy (%)	CPU time (s)	Accuracy (%)	CPU time (s)
O-FDDL (TwIST)	94.60 ± 0.60	805	95.87 ± 0.97	1513
O-FDDL (FISTA)	94.63 ± 0.75	660	95.90 ± 1.04	1232
O-FDDL (NAPG)	94.84 ± 0.79	435	95.82 ± 0.94	853
S-FDDL (IPM)	90.88 ± 1.49	6	92.11 ± 1.53	6
E-FDDL (NADGA)	**95.44 ± 0.71**	221	**96.71 ± 0.57**	519
D-KSVD	88.60 ± 1.14	453	90.76 ± 0.78	581
LC-KSVD	89.05 ± 0.93	18	91.03 ± 0.61	24

Table 2
Recognition results on CMU PIE Face Database (Pose c27) (The best accuracy is marked in boldface.)

DL method (Opt. strategy for FDSR or A-FDSR)	$K_j=N_j=10$ for O-FDDL and E-FDDL		$K_j=N_j=15$ for O-FDDL and E-FDDL	
	Accuracy (%)	CPU time (s)	Accuracy (%)	CPU time (s)
O-FDDL (TwIST)	96.12 ± 0.42	1882	94.33 ± 0.79	4607
O-FDDL (FISTA)	95.88 ± 0.66	1839	94.31 ± 0.58	5869
O-FDDL (NAPG)	96.29 ± 0.38	1295	94.94 ± 0.39	4064
S-FDDL (IPM)	94.56 ± 0.63	6	95.02 ± 0.48	7
E-FDDL (NADGA)	**97.15 ± 0.42**	389	**97.69 ± 0.39**	1414
D-KSVD	95.77 ± 0.76	528	96.43 ± 0.39	701
LC-KSVD	95.53 ± 0.80	22	96.49 ± 0.61	22

4. Experiments

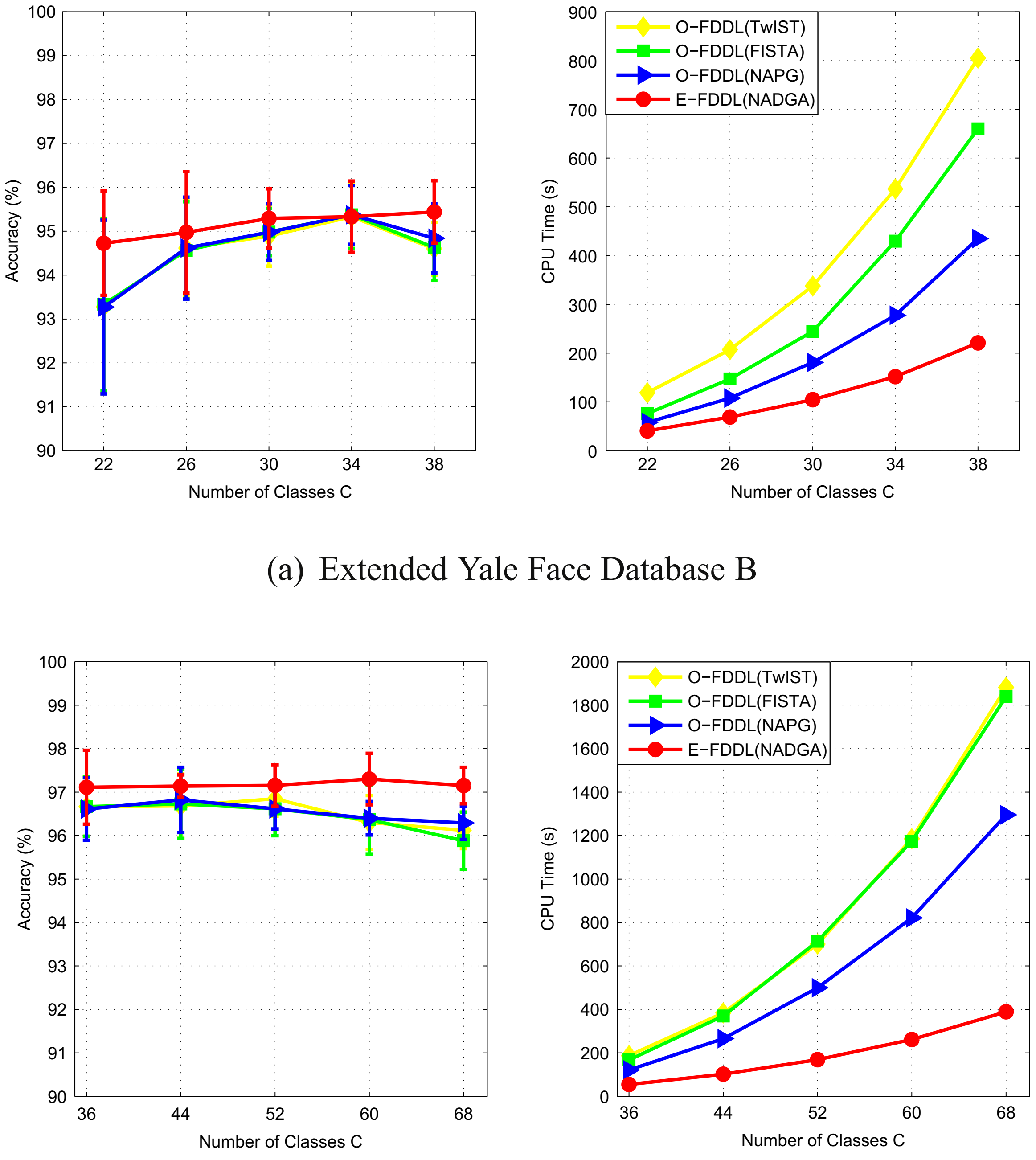
We now conduct face and object recognition experiments to verify the performance of the proposed E-FDDL method. All experiments are performed in Matlab R2013b on a Lenovo Windows 7 PC with Intel Core i3-2120 CPU (3.30 GHz) and 12 GB RAM.

4.1. Face recognition

We first present experimental results on face recognition to evaluate the effectiveness of E-FDDL for classification tasks with unbalanced variations in different classes.

Table 3
Recognition results on CMU PIE Face Database (Pose c27) with N_j=20 (The best accuracy is marked in boldface.)

DL method (Opt. strategy for FDSR or A-FDSR)	$K_j=N_j=20$ for O-FDDL and E-FDDL		K_j=15 for O-FDDL and E-FDDL	
	Accuracy (%)	CPU time (s)	Accuracy (%)	CPU time (s)
O-FDDL (TwIST)	88.90 ± 0.54	9114	95.37 ± 0.62	6712
O-FDDL (FISTA)	91.87 ± 0.63	14,830	95.18 ± 0.63	6528
O-FDDL (NAPG)	93.12 ± 1.21	9586	95.49 ± 0.60	5810
S-FDDL (IPM)	95.62 ± 0.47	8	–	–
E-FDDL (NADGA)	**97.66 ± 0.42**	3583	**97.69 ± 0.41**	2292
D-KSVD	97.19 ± 0.15	1031	–	–
LC-KSVD	97.27 ± 0.43	45	–	–



(a) Extended Yale Face Database B

(b) CMU PIE Face Database

Fig. 2. The performance of O-FDDL and E-FDDL on (a) and (b) versus the number of classes C. Left: Recognition accuracy versus C. Right: CPU time versus C.

4.1.1. *Data sets*

The following two face databases are used in the experiments.

- The first data set is Extended Yale Face Database B [29] with $C=38$ objects and 64 near frontal images for each object. We use the cropped images and resize them into 32 × 32. All the 64 images are taken under different illuminations.
- The second data set is CMU PIE Face Database [30] with $C=68$ classes. There are 49 near frontal images of size 64 × 64 for each object. In addition to illumination changes, the expression variations and the accessory variations make the task more challenging.

For the two datasets, we randomly select N_j images from each class for training and other different N_j images for test. In this way, the variations in the C classes are very likely to be different. Part of training examples used are shown in Fig. 1 from which we can see the unbalanced changes in different classes. Note that the lighting and expression variations in the five classes shown in Fig. 1(a) are not very similar, and in Fig. 1(b), not only the non-uniform lighting and expression variations but also the unbalanced accessory (e.g., glasses) variations can be observed. In all experiments, we use the Eigenface feature [31] with dimension 300, i.e., $n=300$.

4.1.2. *Compared DL methods*

We compare E-FDDL with O-FDDL and S-FDDL. For O-FDDL, three optimization strategies are used for FDSR, i.e., TwIST[1] [32], FISTA [26] and NAPG [24]. We apply IPM [25] to minimize FDSR in

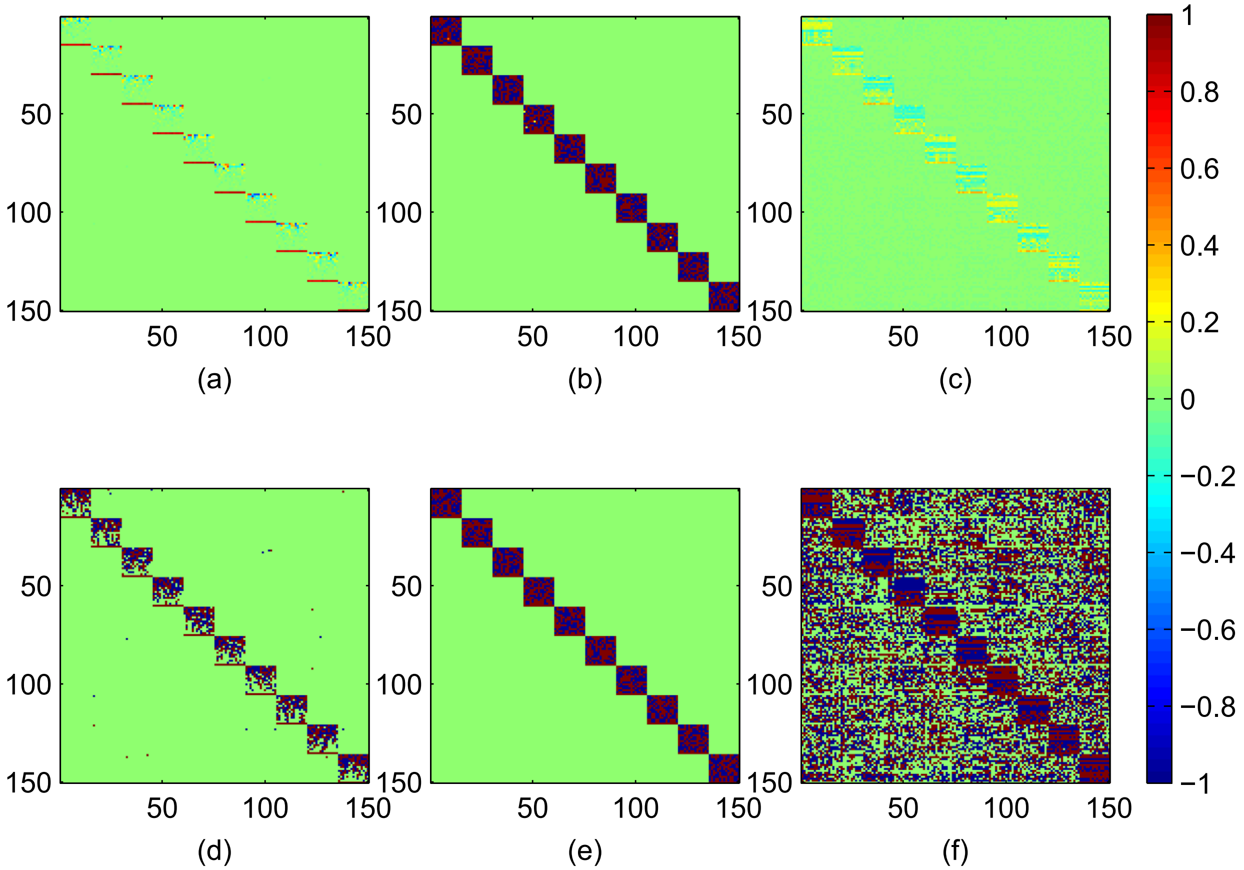


Fig. 3. Visualization of sparse representation matrices ***X*** learned by (a) O-FDDL, (b) S-FDDL and (c) E-FDDL on Extended Yale Face Database B. (d)–(f) are the corresponding sign matrices. For space limitations, here we only visualize sub-matrices extracted from the first 150 rows and the first 150 columns of the original matrices.

S-FDDL and employ NADGA [28] to optimize A-FDSR in E-FDDL. The DU strategy presented in [3,4] is used in all the three DL methods. We set the maximum number of iteration as 200 for the optimization of each subproblem, i.e., (4) and (16), and $T=15$ for each DL method. We use the relative change in the objective function as the stopping criterion for IPM, TwIST, FISTA and NAPG, and choose the relative dual gap as the stopping criterion for NADGA. For all stopping criteria, we set the threshold value ε to be 10^{-6}. In each DL stage, there are three parameters to be tuned. They are the number K_j of the sub-dictionary columns, the sparsity regularization parameter λ_1 and the FDC regularization parameter λ_2. For E-FDDL and O-FDDL, we set K_j as the number of the training examples in class j, i.e., $K_j=N_j$, unless otherwise stated. For S-FDDL, K_j is selected by cross validation with the search range $\{1, 2, \ldots, N_j\}$. For all the three DL methods, we use cross validation to choose λ_1 and λ_2, and set the search range to be {0.001, 0.005, 0.01, 0.05, 0.1}. And η is fixed to 1. After the dictionary ***D*** and the sparse representation matrix ***X*** are learned, in the classification stage, we employ the global classifier presented in [3,4]. There are two parameters γ and ω to be tuned. We set $\gamma = \lambda_1$ and $\omega = 0.5$.

We also compare E-FDDL with D-KSVD [12] and LC-KSVD [14]. D-KSVD formulates DL and classifier learning as a unified framework, and employs the KSVD algorithm to solve the optimization problem. LC-KSVD introduces an explicit correspondence between the dictionary columns and the label information, thus formulating a discriminative sparse code error term which is incorporated into the objective function of D-KSVD as a regularization term. We use the D-KSVD and LC-KSVD codes provided by their authors and select the parameter K by cross validation. The search range of K is $38 \times \{3, 4, 5, 6, 7, 8\}$ for Extended Yale Face Database B, and $68 \times \{3, 4, 5, 6, 7, 8\}$ for CMU PIE Face Database. For the D-KSVD model, we set $\gamma=255$ and $\beta=1$, while, for the LC-KSVD model, we use $\alpha=16$ and $\beta=4$. For these two methods, the sparsity parameter T is set as 16 and the iteration number is set as 100. In the classification stage, we employ the corresponding classifiers presented in [12,14].

4.1.3. *Face recognition results*

In the first and second experiments, we use Extended Yale Face Database B. In the first experiment, we set $N_j=15$, while, in the second experiment, we increase the sample size N_j to 20. Accordingly, the size of the test set from each class is also increasing. With this setting, we compare these DL methods in terms of accuracy, efficiency and stability. For E-FDDL and O-FDDL, the result of cross-validation is $\lambda_1 = 0.005$ and $\lambda_2 = 0.001$, while, for S-FDDL, $K_j=7$, $\lambda_1 = 0.005$ and $\lambda_2 = 0.05$. For D-KSVD and LC-KSVD, $K=38 \times 7=266$ in the first experiment, and $K=38 \times 8=304$ in the second experiment. In each experiment, 10 tests are conducted. The average performance, evaluated by the accuracy and the CPU time, is reported in Table 1. From Table 1, we can see that the results support our analysis. First, FISTA indeed accelerate O-FDDL with TwIST in this case, and NAPG can make O-FDDL faster. The optimization strategy for FDSR has little effect on the recognition accuracy of O-FDDL. Second, S-FDDL is extremely fast, but its recognition accuracy is obviously lower than that of O-FDDL and E-FDDL. This suggests that, in this case, the role of the collaborative reconstruction is indispensable. Thirdly, E-FDDL is much faster than O-FDDL. It is also observed that, in the two experiments, E-FDDL is much better than D-KSVD in both accuracy and

[1] In the demo of FDDL provided by the authors, they employ TwIST, instead of IPM, as the optimization strategy for FDSR.

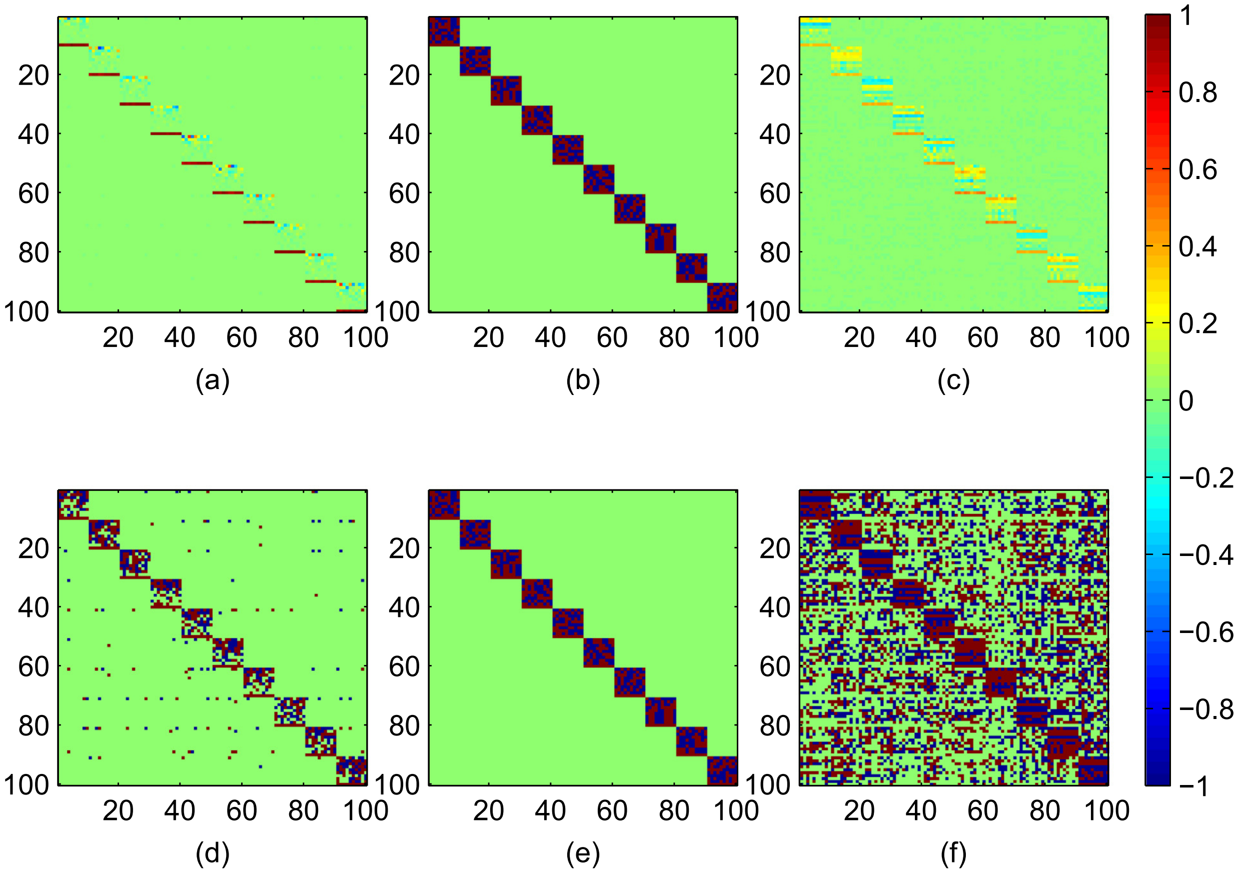


Fig. 4. Visualization of sparse representation matrices $\boldsymbol{X}$ learned by (a) O-FDDL, (b) S-FDDL and (c) E-FDDL on CMU PIE Face Database. (d)–(f) are the corresponding sign matrices. For space limitations, here we only visualize sub-matrices extracted from the first 100 rows and the first 100 columns of the original matrices.

efficiency. LC-KSVD is faster than E-FDDL, but its recognition accuracy is much lower than that of E-FDDL. When N_j increases, the accuracy of all the comparing methods are increasing, but E-FDDL has the best performance.

In the third and fourth experiments, we use CMU PIE Face Database. For E-FDDL and O-FDDL, the value of the parameters selected by cross validation are $\lambda_1 = 0.005$ and $\lambda_2 = 0.005$. For S-FDDL, $K_j = 3$, $\lambda_1 = 0.005$ and $\lambda_2 = 0.1$. For D-KSVD and LC-KSVD, K is set to be $68 \times 4 = 272$ and $68 \times 3 = 204$, respectively. We choose $N_j = 10$ in the third experiment and then increase the number N_j to 15 in the fourth experiment. 10 tests are conducted, and the average performance of each DL method is reported in Table 2. We can see that, in both experiments, E-FDDL still outperforms S-FDDL in accuracy and is faster than O-FDDL. In addition, in the two experiments, the accuracy performance of E-FDDL is still better than that of D-KSVD and LC-KSVD, though its efficiency performance is not the best. From Table 2 it is seen that, when N_j increases, O-FDDL using FISTA becomes the slowest algorithm, and the accuracy of the O-FDDL methods is decreasing. It is further seen that the accuracy of S-FDDL method is increasing and getting similar with that of the O-FDDL methods; however, the recognition performances of other comparing methods is better and more stable, and in particular, E-FDDL gives the most stable recognition performance.

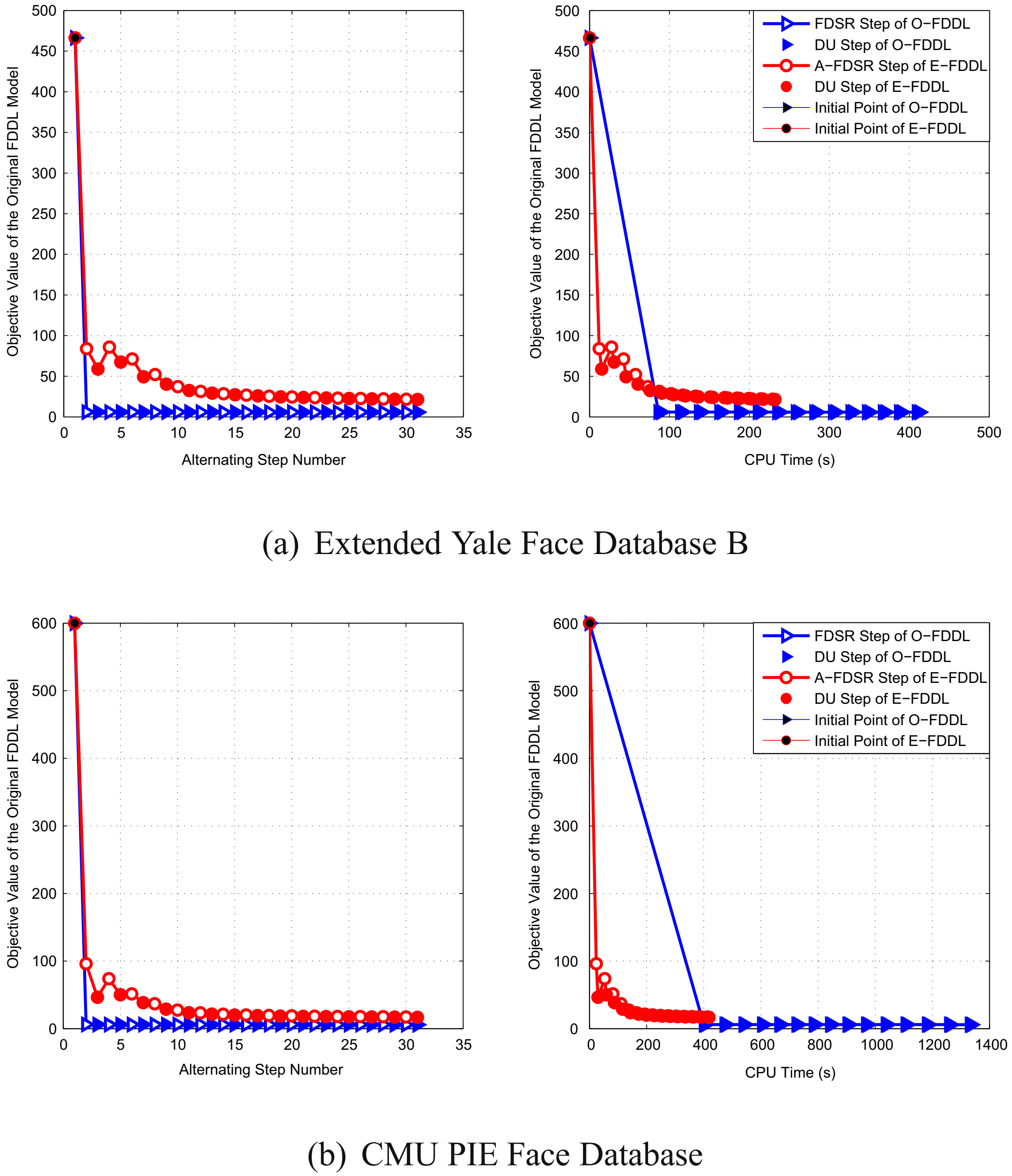
We further conduct the fifth and sixth experiments on CMU PIE Face Database. In these two experiments, we increase N_j to 20, and select $K_j = 20$ and $K_j = 15$, respectively. For D-KSVD and LC-KSVD, K is set as $68 \times 5 = 340$. For other experimental settings, we follow that used in the third and fourth experiments. 10 tests are conducted, and the average performance of each DL method is recorded in Table 3. From Table 3 it is noted that, under the setting $K_j = N_j$, compared with other DL methods, the O-FDDL methods give worse result in term of accuracy and CPU time. It is founded that the setting $K_j = 15$ is more appropriate for the case with a larger size of examples. Specifically, as N_j increases, the unbalanced variations are getting reduced, so, under the setting $K_j = 15$, the recognition performance of O-FDDL and S-FDDL is almost the same. In both experiments, E-FDDL performs stably. Though E-FDDL is much slower than D-KSVD and LC-KSVD, but its recognition accuracy is better than that of the two state-of-the-art algorithms.

We now investigate how O-FDDL and E-FDDL behave as we vary the number C of classes in the first and the third experiments. For the first experiment, we set the values of C as 22, 26, 30, 34, 38. And for the second experiment, the values of C are set to be 36, 44, 52, 60, 68. For each value of C, 10 tests are conducted. And all the average results are shown in Fig. 2. From Fig. 2 it is observed that E-FDDL stably outperforms O-FDDL in both accuracy and CPU time, and meanwhile, its stability performance is also good.

4.1.4. The role of collaborative reconstruction

From all the experimental results, we can empirically observe that, for these face recognition tasks involving data with unbalanced variations in different classes, the recognition accuracy of E-FDDL is much better than that of S-FDDL, and is also better even than that of O-FDDL. We now investigate the reason for this.

Figs. 3 and 4 show the sparse representation matrices $\boldsymbol{X}$ leaned by O-FDDL, S-FDDL and E-FDDL on the two databases and their sign matrices. From visualization of $\boldsymbol{X}$ generated by O-FDDL, it is clear that $\boldsymbol{X}$ learned by O-FDDL is an almost block diagonal matrix, the sign matrix of which reveals the idea of discriminative reconstruction and collaborative reconstruction, that is, each



(a) Extended Yale Face Database B

(b) CMU PIE Face Database

Fig. 5. Convergence curves of O-FDDL using NAPG and E-FDDL. Left: Objective value of the original O-FDDL model versus alternating step number. Right: Objective value of the original O-FDDL versus CPU time.

training example is mainly reconstructed by columns from its corresponding sub-dictionary, and a few columns from other sub-dictionaries with coefficients very close to zero are also helpful in the reconstruction. In contrast to O-FDDL, due to the assumption that each training example can only be reconstructed by columns in its corresponding sub-dictionary, $\boldsymbol{X}$ learned by S-FDDL is an exact block diagonal matrix, which implies the ignorance of the collaborative reconstruction. For E-FDDL, the learned “sparse” representation matrix $\boldsymbol{X}$ in which coefficients with large absolute values sparsely distributed is also an almost block diagonal matrix. Different from that of O-FDDL, the sign matrix shows that much more columns from other sub-dictionaries with very small coefficients are used in the reconstruction of each training example, which can lead to much more complementary information. This indicates that E-FDDL considers both the discriminative reconstruction and collaborative reconstruction, and meanwhile, emphasizes the role of the latter, thus giving more accurate and stable performance in such image classification tasks, where non-uniform variations exist in different classes in the data.

4.1.5. Convergence study

In this part, we show numerically that the E-FDDL method is convergent. We further investigate how fast E-FDDL can converge. Fig. 5 displays the convergence curves of both O-FDDL and E-FDDL on the two face databases. For the two figures shown on the left, the y-axis is the objective value of the original FDDL model, and the x-axis is the alternative step number. We can see that, compared with O-FDDL, E-FDDL always converges to a point

Table 4
Recognition results on single source four domains benchmark (The best results are marked in boldface.)

Source	Target	Accuracy (%)		CPU time (s)	
		SDDL	E-SDDL	SDDL	E-SDDL
Amazon	Caltech-256	32.96 ± 2.59	**33.56 ± 2.44**	63	**30**
Amazon	DSLR	85.36 ± 3.36	**86.91 ± 3.22**	61	**31**
Amazon	Webcam	80.55 ± 3.73	**81.79 ± 3.36**	62	**31**
DSLR	Amazon	55.64 ± 2.49	**56.30 ± 2.30**	37	**14**
DSLR	Caltech-256	32.22 ± 2.75	**32.65 ± 2.53**	37	**14**
DSLR	Webcam	82.25 ± 3.85	**82.34 ± 4.24**	36	**14**
Webcam	Amazon	**56.35 ± 2.57**	55.60 ± 2.80	36	**13**
Webcam	Caltech-256	32.58 ± 1.58	**32.63 ± 1.51**	37	**14**
Webcam	DSLR	**84.95 ± 4.69**	82.89 ± 4.21	36	**14**

with a larger objective value. This is due to the fact that the objective function of A-FDSR problem in the E-FDDL method is an upper bound of that of the FDSR problem in the O-FDDL method. However, this has no negative effect on the classification accuracy of E-FDDL in our experiments. Furthermore, the two figures shown on the right, whose x-axis is the CPU time, suggest that E-FDDL always consumes much less CPU time than O-FDDL using the NAPG method for solving the FDSR problem.

4.2. Object recognition

In this subsection, we present object recognition experiments on two real-world databases involving four different visual domains to evaluate the performance of the E-FDDL based E-SDDL method in domain adaptation applications.

4.2.1. Data sets

The following two real-world object databases are used in the experiments.

- The first one is the benchmark database for domain adaptation contributed by [33]. This database contains three domains: images downloaded from Amazon, high quality images captured by a Digital Single-Lens Reflex (DSLR) camera and low quality images from a webcam. For simplicity, we use Amazon, DSLR and Webcam to represent the three domains, respectively.
- The second one is the Caltech-256 database [34] which is used as the fourth domain but only as one of the target domains.

In all the four datasets, there are 10 common classes: Backpack, Bike, Calculator, Headphone, Keyboard, Laptop Computer, Monitor, Mouse, Mug and Projector. We restrict to the 10 classes. We use Amazon, DSLR or Webcam as the source domain and then choose one of the remaining two datasets and the Caltech-256 dataset as the target domain, so we have 9 source-target pairs. We randomly select 20 training examples per class from Amazon, 8 training examples per class from DSLR and Webcam when used as source, while 6 training examples from all the datasets when used as target. All the remaining images from the target domain are set as examples for test. We use the 800 dimensional SURF [33] features in all the experiments. For more details of the feature extraction, see experimental sections of [21,22].

4.2.2. Compared DL methods

We compare the O-FDDL based SDDL method and the E-FDDL based E-SDDL method. For these two DL methods, we use their kernelized versions, and the used kernel is the non-parametric histogram intersection kernel. We set $\tau_1 = 4$, $\tau_2 = 30$, $\lambda_1 = 0.043$, $K_i{=}8$, $n{=}60$. For O-FDDL and E-FDDL, the number of iterations is set as 10. We apply TwIST to solve the FDSR problem in O-FDDL and NADGA to solve the A-FDSR problem in E-FDDL. For the setting of other relevant parameters, see Section 4.1.2. For SDDL and E-SDDL, the number of iterations is also set as 10. After the SDDL and E-SDDL procedures are ended completely, classification scheme proposed in [21,22] is used to do recognition. For the involved OMP algorithm [35], T_0 is set as 15.

4.2.3. Object recognition results

For each experiment, 10 tests are conducted, and the average results evaluated by the accuracy and the CPU time are shown in Table 4. From these results, we can see that the proposed E-SDDL method gives a similar recognition accuracy with the original SDDL method. However, in almost every case, the CPU time of E-SDDL is less than or equal to half of that of SDDL based on O-FDDL. These observations indirectly support the stable and efficient performance of E-FDDL.

5. Conclusion

In this paper, we proposed an efficient FDDL algorithm called E-FDDL. In addition to the role of the discriminative representation, E-FDDL considers the role of the discriminative reconstruction and collaborative reconstruction. This makes E-FDDL more effective than S-FDDL when dealing with classification problems involving data with unbalanced variations in different classes. Furthermore, based on the analysis on the convergence rate and the computational complexity, we conclude that E-FDDL is much faster than O-FDDL. We also employed E-FDDL to accelerate the SDDL method for domain adaptation applications. Face and object recognition experiments on real-world databases have been conducted to verify the effectiveness of E-FDDL.

Acknowledgments

The authors thank the handling editor and the referees for their constructive comments. The authors thank Yuren Zhang and Yinlin Li for their helpful suggestions. This work was partly supported by the National Natural Science Foundation of China under Grants 61210009, 61033011, 61379093 and 11131006.

Appendix A

In this appendix, we explain the reason why NADGA is not suggested to use in solving the FDSR problem (2). By fixing $\boldsymbol{x}_k (k \neq i)$ first and then updating each $\boldsymbol{x}_i$ individually, the FDSR problem (2) can be split into N problems

$$\min_{\boldsymbol{x}_i} \quad \| \boldsymbol{y}_i - \boldsymbol{D}^{(t)}\boldsymbol{x}_i \|_2^2 + \| \boldsymbol{y}_i - \boldsymbol{D}_{c_i}^{(t)}\boldsymbol{x}_i^{c_i} \|_2^2 + \sum_{j \neq c_i} \| \boldsymbol{D}_j^{(t)}\boldsymbol{x}_i^j \|_2^2 + \sum_{j=1}^{C} (\alpha_i^j \| \boldsymbol{x}_i^j \|_2^2 - \boldsymbol{\beta}_i^{j\top}\boldsymbol{x}_i^j + \lambda_1 \| \boldsymbol{x}_i^j \|_1), \quad (i = 1, \ldots, N), \tag{A.1}$$

where $\alpha_i^j = \lambda_2[\eta + 1 - (2/N_j) + (1/N)]$, $\boldsymbol{\beta}_i^j = \boldsymbol{\gamma}_i^j$ and $\boldsymbol{\gamma}_i$ is defined in Section 3.2. For simplicity, dropping the subscript i in (A.1) and obtain that

$$\min_{\boldsymbol{x}} \quad \| \boldsymbol{y} - \boldsymbol{D}^{(t)}\boldsymbol{x} \|_2^2 + \| \boldsymbol{y} - \boldsymbol{D}_{c_i}^{(t)}\boldsymbol{x}^{c_i} \|_2^2 + \sum_{j \neq c_i} \| \boldsymbol{D}_j^{(t)}\boldsymbol{x}^j \|_2^2 + \sum_{j=1}^{C} (\alpha^j \| \boldsymbol{x}^j \|_2^2 - \boldsymbol{\beta}^{j\top}\boldsymbol{x}^j + \lambda_1 \| \boldsymbol{x}^j \|_1). \tag{A.2}$$

Our NADGA [28] aims at deriving a dual problem of the original problem, say (A.2) here, and then using Nesterov's Accelerated

Gradient (NAG) method to solve this dual problem. When $\lambda_2 > 0$, i.e., $\alpha^j > 0$, let $R(\boldsymbol{z}) = \| \boldsymbol{y} - \boldsymbol{z} \|_2^2$, $R_{c_i}(\boldsymbol{z}_{c_i}) = \| \boldsymbol{y} - \boldsymbol{z}_{c_i} \|_2^2$, $R_j(\boldsymbol{z}_j) = \| \boldsymbol{z}_j \|_2^2$ for $j \neq c_i$ and $r_j(\boldsymbol{x}^j) = \alpha^j \| \boldsymbol{x}^j \|_2^2 - \boldsymbol{\beta}^{j\top}\boldsymbol{x}^j + \lambda_1 \| \boldsymbol{x}^j \|_1$. Then a dual problem of (A.2) is given as

$$\min_{\boldsymbol{\mu},\boldsymbol{\mu}_1,\ldots,\boldsymbol{\mu}_C} \quad F_{\rm D}(\boldsymbol{\mu}, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_C) = R^*(-\boldsymbol{\mu}) + R_{c_i}^*(-\boldsymbol{\mu}_{c_i}) + \sum_{j \neq c_i} R_j^*(-\boldsymbol{\mu}_j) + \sum_{j=1}^{C} r_j^*(\boldsymbol{u}^j), \tag{A.3}$$

where $R^*(-\boldsymbol{\mu}) = (1/4)\boldsymbol{\mu}^\top\boldsymbol{\mu} - \boldsymbol{y}^\top\boldsymbol{\mu}$, $R_{c_i}^*(-\boldsymbol{\mu}_{c_i}) = (1/4)\boldsymbol{\mu}_{c_i}^\top\boldsymbol{\mu}_{c_i} - \boldsymbol{y}^\top\boldsymbol{\mu}_{c_i}$, $R_j^*(-\boldsymbol{\mu}_j) = (1/4)\boldsymbol{\mu}_j^\top\boldsymbol{\mu}_j$ are the conjugate functions of R, R_{c_i} and $R_j(j \neq c_i)$, respectively, $\boldsymbol{u}^j = \boldsymbol{D}_j^{(t)\top}(\boldsymbol{\mu} + \boldsymbol{\mu}_j)$, and

$$r_j^*(\boldsymbol{u}^j) = [1/(4\alpha^j)] \sum_{k=1}^{K_j} [\max\{|u_k^j + \beta_k^j| - \lambda_1, 0\}]^2$$

is the conjugate function of r_j. For the case $\lambda_2 > 0$, the NAG method can also be applicable to the problem (A.3) whose objective function is strongly convex and smooth. However, this problem has $n \times (C+1)$ unknown variables, where C is the number of classes and n is the dimension of the training examples. This means that, if the number of classes C is big then the number of unknown variables of the problem (A.3) is very large. So, compared with using NAG method to solve problem (16) which has only n variables, using NAG method to solve problem (A.3) will lead to a higher computational complexity.

When $\lambda_2 = 0$, i.e., $\alpha^j = 0$, the objective function of the problem (A.2) may not be strongly convex. In this case, we cannot derive a dual problem of the problem (A.2) to which NAG method is applicable. Thus, in this case, NADGA cannot be used to solve the problem (A.2).

Appendix B. Proof of Proposition 3.1

In this appendix, we prove Proposition 3.1 in Section 3.3. From the formulations of L and μ, i.e., (6) and (5), we obtain that

$$L/\mu = \frac{\lambda_{\max}(\boldsymbol{D}^{(t)\top}\boldsymbol{D}^{(t)} + \widetilde{\boldsymbol{D}}^{(t)\top}\widetilde{\boldsymbol{D}}^{(t)}) + \lambda_2(\eta + 1)}{\lambda_{\min}(\widetilde{\boldsymbol{D}}^{(t)\top}\widetilde{\boldsymbol{D}}^{(t)}) + \lambda_2[\eta - 1 + (N_j/N)]}.$$

By Weyl's Theorem on eigenvalues [36], we have

$$\lambda_{\max}(\boldsymbol{D}^{(t)\top}\boldsymbol{D}^{(t)} + \widetilde{\boldsymbol{D}}^{(t)\top}\widetilde{\boldsymbol{D}}^{(t)}) \geq \| \boldsymbol{D}^{(t)} \|_2^2 + \lambda_{\min}(\widetilde{\boldsymbol{D}}^{(t)\top}\widetilde{\boldsymbol{D}}^{(t)}).$$

This implies that

$$L/\mu \geq 1 + \frac{\| \boldsymbol{D}^{(t)} \|_2^2 + \lambda_2[2 - (N_j/N)]}{\lambda_{\min}(\widetilde{\boldsymbol{D}}^{(t)\top}\widetilde{\boldsymbol{D}}^{(t)}) + \lambda_2[\eta - 1 + (N_j/N)]}.$$

Since $\lambda_2 \geq 0$, we have

$$\| \boldsymbol{D}^{(t)} \|_2^2 + \lambda_2[2 - (N_j/N)] \geq \| \boldsymbol{D}^{(t)} \|_2^2,$$

$$\lambda_{\min}(\widetilde{\boldsymbol{D}}^{(t)\top}\widetilde{\boldsymbol{D}}^{(t)}) + \lambda_2[\eta - 1 + (N_j/N)] < \min\{\alpha^j\},$$

This implies that $L/\mu > 1 + \| \boldsymbol{D}^{(t)} \|_2^2/\min\{\alpha^j\} = L(\nabla F_{\rm D})/\mu(F_{\rm D})$. The proof is complete.

References

[1] J. Wright, A. Yang, A. Ganesh, S. Sastry, Y. Ma, Robust face recognition via sparse representation, IEEE Trans. Pattern Anal. Mach. Intell. 31 (2) (2009) 210–227.

[2] H. Cheng, Z. Liu, L. Yang, X. Chen, Sparse representation and learning in visual recognition: theory and applications, Signal Process. 93 (6) (2013) 1408–1425.

[3] M. Yang, L. Zhang, X. Feng, D. Zhang, Fisher discrimination dictionary learning for sparse representation, in: Proceedings of ICCV, 2011, pp. 543–550.

[4] M. Yang, L. Zhang, X. Feng, D. Zhang, Sparse representation based Fisher discrimination dictionary learning for image classification, Int. J. Comput. Vis. 109 (3) (2014) 209–232.

[5] J. Mairal, J. Ponce, G. Sapiro, A. Zisserman, F. R. Bach, Supervised dictionary learning, in: Proceedings of NIPS, 2009, pp. 1033–1040.

[6] F. Rodriguez, G. Sapiro, Sparse Representations for Image Classification: Learning Discriminative and Reconstructive Non-parametric Dictionaries, Technical report, DTIC Document, 2008.

[7] Y. Naderahmadian, M.A. Tinati, S. Beheshti, Generalized adaptive weighted recursive least squares dictionary learning, Signal Process. 118 (2016) 89–96.

[8] M. Aharon, M. Elad, A. Bruckstein, K-svd: an algorithm for designing overcomplete dictionaries for sparse representation, IEEE Trans. Signal Process. 54 (11) (2006) 4311–4322.

[9] D.-S. Pham, S. Venkatesh, Joint learning and dictionary construction for pattern recognition, in: Proceedings of CVPR, 2008, pp. 1–8.

[10] X.-C. Lian, Z. Li, B.-L. Lu, L. Zhang, Max-margin dictionary learning for multiclass image categorization, in: Proceedings of ECCV, 2010, pp. 157–170.

[11] Q. Qiu, Z. Jiang, R. Chellappa, Sparse dictionary-based representation and recognition of action attributes, in: Proceedings of ICCV, 2011, pp. 707–714.

[12] Q. Zhang, B. Li, Discriminative k-svd for dictionary learning in face recognition, in: Proceedings of ICCV, 2010, pp. 2691–2698.

[13] Z. Jiang, Z. Lin, L. Davis, Learning a discriminative dictionary for sparse coding via label consistent k-svd, in: Proceedings of CVPR, 2011, pp. 1697–1704.

[14] Z. Jiang, Z. Lin, L.S. Davis, Label consistent k-svd: learning a discriminative dictionary for recognition, IEEE Trans. Pattern Anal. Mach. Intell. 35 (11) (2013) 2651–2664.

[15] I. Ramirez, P. Sprechmann, G. Sapiro, Classification and clustering via dictionary learning with structured incoherence and shared features, in: Proceedings of CVPR, 2010, pp. 3501–3508.

[16] J. Mairal, F. Bach, J. Ponce, G. Sapiro, A. Zisserman, Discriminative learned dictionaries for local image analysis, in: Proceedings of CVPR, 2008, pp. 1–8.

[17] P. Sprechmann, G. Sapiro, Dictionary learning and sparse coding for unsupervised clustering, in: Proceedings of ICASSP, 2010, pp. 2042–2045.

[18] H. Wang, C. Yuan, W. Hu, C. Sun, Supervised class-specific dictionary learning for sparse modeling in action recognition, Pattern Recognit. 45 (11) (2012) 3902–3911.

[19] A. Castrodad, G. Sapiro, Sparse modeling of human actions from motion imagery, Int. J. Comput. Vis. 100 (1) (2012) 1–15.

[20] Y.N. Wu, Z. Si, H. Gong, S.-C. Zhu, Learning active basis model for object detection and recognition, Int. J. Comput. Vis. 90 (2) (2010) 198–235.

[21] S. Shekhar, V. M. Patel, H. Nguyen, R. Chellappa, Generalized domain-adaptive dictionaries, in: Proceedings of CVPR, 2013, pp. 361–368.

[22] S. Shekhar, V. Patel, H. Nguyen, R. Chellappa, Coupled projections for adaptation of dictionaries, IEEE Trans. Image Process. 24 (10) (2015) 2941–2954.

[23] Z. Zhang, Y. Xu, J. Yang, X. Li, D. Zhang, A survey of sparse representation: algorithms and applications, IEEE Access 3 (2015) 490–530.

[24] Y. Nesterov, Introductory Lectures on Convex Optimization: A Basic Course, Kluwer Academic Publishers, 2004.

[25] L. Rosasco, A. Verri, M. Santoro, S. Mosci, S. Villa, Iterative Projection Methods for Structured Sparsity Regularization, Technical Report MIT-CSAIL-TR-2009-050, CBCL-282, MIT, 2007.

[26] A. Beck, M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, SIAM J. Imag. Sci. 2 (1) (2009) 183–202.

[27] Z. Wen, W. Yin, A feasible method for optimization with orthogonality constraints, Math. Program. 142 (12) (2013) 397–434.

[28] R. Jiang, H. Qiao, B. Zhang, Speeding up graph regularized sparse coding by dual gradient ascent, IEEE Signal Process. Lett. 22 (3) (2015) 313–317.

[29] A. Georghiades, P. Belhumeur, D. Kriegman, From few to many: illumination cone models for face recognition under variable lighting and pose, IEEE Trans. Pattern Anal. Mach. Intell. 23 (6) (2001) 643–660.

[30] T. Sim, S. Baker, M. Bsat, The cmu pose, Illumination, and Expression (pie) Database of Human Faces, Technical Report CMU-RI-TR-01-02, Robotics Institute, 2001.

[31] M. Turk, A. Pentland, Eigenfaces for recognition, J. Cogn. Neurosci. 3 (1) (1991) 71–86.

[32] J. Bioucas-Dias, M. Figueiredo, A new twist: two-step iterative shrinkage/thresholding algorithms for image restoration, IEEE Trans. Image Process. 16 (12) (2007) 2992–3004.

[33] K. Saenko, B. Kulis, M. Fritz, T. Darrell, Adapting visual category models to new domains, in: Proceedings of ECCV, 2010, pp. 213–226.

[34] G. Griffin, A. Holub, P. Perona, Caltech-256 Object Category Dataset, Technical Report CNS-TR-2007-001, 2007.

[35] Y. C. Pati, R. Rezaiifar, P. Krishnaprasad, Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition, in: Conference Record of The Twenty-Seventh Asilomar Conference on Signals, Systems and Computers, 1993, pp. 40–44.

[36] H. Weyl, Das asymptotische verteilungsgesetz der eigenwerte linearer partieller differentialgleichungen, Math. Ann. 71 (4) (1912) 441–479.