Optimal Couple Projections for Domain Adaptive Sparse Representation-Based Classification

Guoqing Zhang, Huaijiang Sun, Fatih Porikli, *Fellow, IEEE*, Yazhou Liu, and Quansen Sun

***Abstract*— In recent years, sparse representation-based classification (SRC) is one of the most successful methods and has been shown impressive performance in various classification tasks. However, when the training data have a different distribution than the testing data, the learned sparse representation may not be optimal, and the performance of SRC will be degraded significantly. To address this problem, in this paper, we propose an optimal couple projections for domain-adaptive SRC (OCPD-SRC) method, in which the discriminative features of data in the two domains are simultaneously learned with the dictionary that can succinctly represent the training and testing data in the projected space. OCPD-SRC is designed based on the decision rule of SRC, with the objective to learn coupled projection matrices and a common discriminative dictionary such that the between-class sparse reconstruction residuals of data from both domains are maximized, and the within-class sparse reconstruction residuals of data are minimized in the projected low-dimensional space. Thus, the resulting representations can well fit SRC and simultaneously have a better discriminant ability. In addition, our method can be easily extended to multiple domains and can be kernelized to deal with the nonlinear structure of data. The optimal solution for the proposed method can be efficiently obtained following the alternative optimization method. Extensive experimental results on a series of benchmark databases show that our method is better or comparable to many state-of-the-art methods.**

***Index Terms*— Dictionary learning, sparse representation, domain adaptation, joint projection and dictionary learning.**

I. Introduction

OVER the past few years, sparse representation has been successfully used in a wide variety of computer vision problems such as face recognition [1], image restoration [2], [3], image denoising [4] and image classification [5], [6]. To represent data efficiently and provide resilience against noise, sparse coding seeks a description of the signal as a linear combination of a few atoms in a dictionary that is usually over-completed. In sparse representation, the dictionary plays an important role as it is expected to faithfully and discriminatively represent the query image. Wright et al. [1] proposed a sparse representation based classification (SRC) method that employed the entire set training samples as the dictionary. Earlier methods take off-the-shelf bases (e.g., wavelets) as the dictionary [7], yet these dictionaries may not be optimal choices in certain recognition tasks. Instead, learning the best dictionary directly from training samples can lead to better performance [8]–[11]. There have been many state-of-the-art dictionary learning (DL) algorithms proposed in the literature, such as [12]–[16].

Fig. 1. Sample images from two different domains, namely Amazon and DLSR datasets.

When the training data used to train a classifier has a different distribution than the testing data which the classifier is applied, the learned dictionary may be inefficient, and the classification performance will degrade significantly at test time. In many practical applications, we often confront with these situations, for example, Fig. 1 shows some images of the bike class from two different datasets. Although these images have the same object category label, they are dissimilar visually. These variations often result in poor cross-dataset generalization. Domain adaptation (DA), also known as domain transfer learning, attempts to tackle this problem, where the training and testing data have the same categories, but the domain shift is unknown [17].

Domain adaptation has received substantial attention and has been extensively studied in many areas, including speech and language processing [18], machine learning [19], [20], and more recently computer vision [17], [21]–[23], [30]. Domain adaptation for visual recognition was first investigated by Saenko et al. [23] in a semi-supervised setting. This idea was extended by Brian et al. [17] to handle asymmetric domain transformation. Gopalan et al. [22] addressed the

Manuscript received October 21, 2016; revised July 18, 2017; accepted August 17, 2017. Date of publication August 29, 2017; date of current version September 21, 2017. This work was supported by the National Science Foundation of China under Grant 61772272, Grant 61273251, Grant 61401209, Grant 61673220, and Grant 61672286. The work of F. Porikli was supported by the Australian Research Council's Discovery Projects Funding Scheme under Project DP150104645. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Volkan Cevher. *(Corresponding author: Huaijiang Sun.)*

G. Zhang is with the School of Computer Science and Technology, Nanjing University of Science and Technology, Nanjing 210094, China, and also with the Research School of Engineering, Australian National University, Canberra, ACT 2601, Australia (e-mail: xiayang14551@163.com).

H. Sun, Y. Liu, and Q. Sun are with the School of Computer Science and Technology, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: sunhuaijiang@njust.edu.cn; yazhouliu@njust.edu.cn; sunquansen@njust.edu.cn).

F. Porikli is with the Research School of Engineering, Australian National University, Canberra, ACT 2601, Australia (e-mail: fatih.porikli@anu.edu.au).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2017.2745684

1057-7149 © 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

problem of unsupervised domain adaptation, by using an incremental approach based on Grassmann manifolds. This was further extended to learning a kernel distance between domains in [24]. A feature augmentation method was proposed in [28] and an information-theoretic method for unsupervised domain adaptation was proposed by Shi and Sha [29]. Various subspace based methods have also been proposed to tackle domain adaptation problem [25]–[27]. The idea behind these methods aims to find a latent space that is domain invariant, and then project the data from different domains onto this space where classification is performed. Another class of domain adaptation algorithms is based on parameter adaptation, such as domain transfer SVM [32], max-margin domain transfer [33] and domain adaptive multiple kernel learning [34]. We refer readers to [35] for a comprehensive survey on domain adaptation.

Sparse representation and dictionary based methods for domain adaptation are also gaining significant attention [36]–[39]. Zhang et al. [36] proposed a domain adaptive sparse representation based classification method (DASRC) that learns projections of data in a space where the sparsity of data is maintained. Ni et al. [27] proposed an unsupervised domain-adaptive dictionary learning framework by generating a set of intermediate dictionaries that bridge the domain shift. Shekhar et al. [38], [39] proposed a generalized domain adaptive dictionary learning framework for jointly learning coupled projections of data in the source and target domains. Nonetheless, a major drawback of these adapting dictionary learning approaches is that they have no direct connection to the decision rule of SRC. Thus, the learned spares representation may not be optimal for domain adaptation recognition problem.

Recently, deep learning has been widely applied in cross-domain classification and made notable improvement [53]–[57]. This is due in part to the fact that deep networks are able to learn extremely powerful hierarchical nonlinear representations of the inputs [58]–[60], making them suitable for domain adaptation. Sun and Saenko [58] addressed the case when the target domain is unlabeled, by extending correlation alignment objective to learn a nonlinear transformation that aligns correlations of layer activations in deep neural networks. Tzeng et al. [59] proposed a CNN architecture which introduces an adaptation layer and an additional domain confusion loss, to learn a representation that is both semantically meaningful and domain invariant. Although its success, deep learning based methods still have limitations. For example, they require a large number of additional training images, which is not available in many practical applications and superior computational platforms. As an alternative, our paper focuses on a method that does not use the large number of additional training samples.

Based on the above motivations, in order to enhance the recognition performance of SRC for domain adaptation, here we propose a novel dictionary learning method termed optimal couple projections for domain adaptive sparse representation-based classification (OCPD-SRC), which jointly learns the transformations of data in different domains and a discriminative dictionary in a common space. Since SRC predicts the class label of a given testing image based on the representation residual, OCPD-SRC learns coupled projection matrices and a common dictionary such that in the projected low dimensional subspace the between-class sparse reconstruction residuals of data in different domains are maximized and the within-class sparse reconstruction residuals of data are minimized. Thus, the resulting representations can well fit SRC and simultaneously have a better discriminant capacity. Moreover, the jointly learned projections can preserve the sparse structure of data, and the learned dictionary can represent the projected data from both domains well, and further enhances the discriminant capability of the coding vectors in the transformed subspace. Therefore, SRC can achieve optimal performance in the reduced space. Furthermore, our method can easily extend to handle multiple domains and is able to handle sparsity in nonlinear models by using kernel method. We adopt an iterative learning framework to alternatively derive the coupled projections and class-wise dictionary. An overview of the proposed method is shown in Fig. 2.

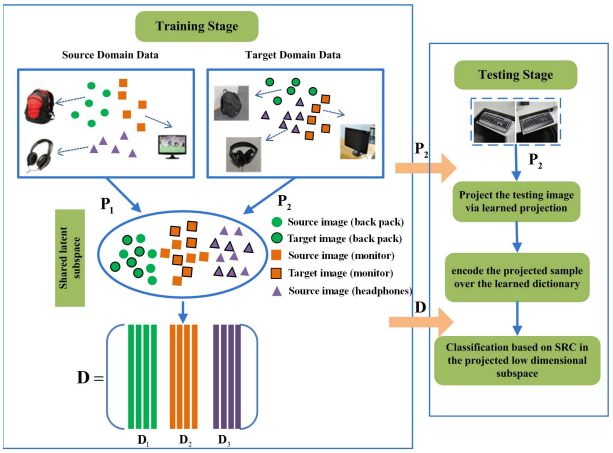


Fig. 2. Overview of the proposed OCPD-SRC method.

The remainder of paper is organized as follows. Section 2 reviews some of the related works. The proposed algorithm is introduced in Section 3, and the optimization technique is described in Section 4. Section 5 presents the classification scheme. Comparative experiments for the proposed algorithm and the related algorithms are shown in Section 6. Section 7 summarizes our conclusions.

II. Related Work

A. Sparse Coding and Dictionary Learning

Given an over-complete dictionary $\mathbf{D}$ and a signal $\mathbf{y}$, finding a sparse representation of $\mathbf{y}$ in $\mathbf{D}$ entails attaining the following optimization objective [1], [40]

$$\hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_0 \quad s.t.\ \mathbf{y} = \mathbf{D}\boldsymbol{\alpha}, \tag{1}$$

where $\|\cdot\|_0$ denotes the ℓ_0-norm, which counts the numbers of nonzero entries in the vector $\boldsymbol{\alpha}$. Solving for the objective in Eq.(1) is NP-hard and extremely time-consuming. Hence, an approximate solution is usually sought by the ℓ_1 optimization formulation

$$\hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} \|\mathbf{y} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1, \tag{2}$$

where λ is the regularization parameter to trade off between reconstruction error and sparsity.

The dictionary learning problem allows obtaining an appropriate dictionary that has a small reconstruction error over training data while preserving the sparse penalty. Let $\mathbf{Y}$ be a set of N input signals in a m-dimensional feature space $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_N]$. A dictionary with a fixed number of K atoms can be derived by the following minimization

$$\begin{aligned} &\arg\min_{\mathbf{D},\mathbf{X}} \ \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_2^2 + \lambda\|\mathbf{X}\|_1 \\ &s.t. \ \|\mathbf{d}_i\|_2 = 1, \quad \forall i \end{aligned} \tag{3}$$

where $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \cdots, \mathbf{d}_K] \in R^{m\times K}$ is the sought dictionary, $\mathbf{X} = [\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \cdots, \boldsymbol{\alpha}_N]$, $\boldsymbol{\alpha}_i \in R^K$ is the sparse code of input signal $\mathbf{y}_i$ over $\mathbf{D}$. Each dictionary atom $\boldsymbol{d}_i$ is ℓ_2-normalized, and a common approach for solving this problem is to update $\mathbf{X}$ and $\mathbf{D}$ alternately. When the dictionary $\mathbf{D}$ is fixed, optimizing the coefficient matrix $\mathbf{X}$ is exactly the sparse coding problem. In reverse, to update dictionary with $\mathbf{X}$ fixed. The dictionary can be learned class by class [41], [42].

B. Shared Domain-Adapted Dictionary Learning (SDDL)

We briefly explain the principle of shared domain-adapted dictionary learning (SDDL) [38], [39] in this section. Suppose we have the source and target domain data $\mathbf{Y_1} \in R^{m_1\times N_1}$ and $\mathbf{Y_2} \in R^{m_2\times N_2}$, respectively. SDDL aims to learn a shared K-atoms dictionary $\mathbf{D} \in R^{m_f\times K}$ and mapping $\mathbf{P_1} \in R^{m_f\times m_1}$ and $\mathbf{P_2} \in R^{m_f\times m_2}$ onto a common low-dimensional space that minimizes the reconstruction error in the projected space. The cost function can be expressed as follows

$$C_1(\mathbf{D}, \mathbf{P_1}, \mathbf{P_2}, \mathbf{X_1}, \mathbf{X_2}) = \|\mathbf{P_1}\mathbf{Y_1} - \mathbf{D}\mathbf{X_1}\|_F^2 + \|\mathbf{P_2}\mathbf{Y_2} - \mathbf{D}\mathbf{X_2}\|_F^2, \tag{4}$$

where $\mathbf{X_1} = [\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \cdots, \boldsymbol{\alpha}_{N_1}] \in R^{K\times N_1}$ and $\mathbf{X_2} = [\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \cdots, \boldsymbol{\alpha}_{N_2}] \in R^{K\times N_2}$ are the sparse representations of $\mathbf{Y_1}$ and $\mathbf{Y_2}$ over $\mathbf{D}$, respectively. $\mathbf{X_1}$ and $\mathbf{X_2}$ subject to sparsity constraints. $\mathbf{P_1}$ and $\mathbf{P_2}$ are orthogonal and normalized to unit-norm.

To enforce the projections do not lose too much information available in the original domains after the projection onto the latent space, a PCA-like regularization term is added, given as

$$C_2(\mathbf{P_1}, \mathbf{P_2}) = \|\mathbf{Y_1} - \mathbf{P_1^T}\mathbf{P_1}\mathbf{Y_1}\|_F^2 + \|\mathbf{Y_2} - \mathbf{P_2^T}\mathbf{P_1}\mathbf{Y_2}\|_F^2. \tag{5}$$

It is straightforward to show that the costs C_1 and C_2, after ignoring the constant terms in $\mathbf{Y}$ can be written as

$$C_1(\mathbf{D}, \widetilde{\mathbf{P}}, \widetilde{\mathbf{X}}) = \|\widetilde{\mathbf{P}}\widetilde{\mathbf{Y}} - \mathbf{D}\widetilde{\mathbf{X}}\|_F^2, \tag{6}$$

$$C_2(\widetilde{\mathbf{P}}) = -trace\left((\widetilde{\mathbf{P}}\widetilde{\mathbf{Y}})(\widetilde{\mathbf{P}}\widetilde{\mathbf{Y}})^T\right), \tag{7}$$

where $\widetilde{\mathbf{P}} = [\mathbf{P_1}, \mathbf{P_2}]$, $\widetilde{\mathbf{Y}} = \begin{bmatrix} \mathbf{Y_1} & \mathbf{0} \\ \mathbf{0} & \mathbf{Y_2} \end{bmatrix}$, and $\widetilde{\mathbf{X}} = [\mathbf{X_1}, \mathbf{X_2}]$. Hence, the overall optimization becomes

$$\begin{aligned} \left\{\mathbf{D}^*, \widetilde{\mathbf{P}}^*, \widetilde{\mathbf{X}}^*\right\} = \arg\min_{\mathbf{D},\widetilde{\mathbf{P}},\widetilde{\mathbf{X}}} \ & C_1(\mathbf{D}, \widetilde{\mathbf{P}}, \widetilde{\mathbf{X}}) + \lambda C_2(\widetilde{\mathbf{P}}) \\ s.t. \ & \mathbf{P_i}\mathbf{P_i^T} = \mathbf{I}, \quad \mathbf{i} = 1, 2 \ and \ \|\widetilde{\mathbf{x}_j}\|_0 \le T_0, \ \forall j, \end{aligned} \tag{8}$$

where λ is a positive constant and T_0 is the sparsity level. When the projection matrices and the dictionary have been learned, a novel test sample from the target domain can be projected onto the latent domain using $\mathbf{P_2}$ and classified it using the sparse embedding residual classifier proposed in [52].

Let the class-wise dictionary $\mathbf{D}$ be $\mathbf{D} = [\mathbf{D}_1, \cdots, \mathbf{D}_c]$, where c is the total number of classes. In order to improve the discriminative between different classes, SDDL encourages reconstruction samples of a given class by the dictionary of the corresponding class, and penalizes reconstruction by out-of-class dictionaries. Thus, the new cost function $C_1(\mathbf{D}, \widetilde{\mathbf{P}}, \widetilde{\mathbf{X}})$ can be defined as

$$\begin{aligned} C_1(\mathbf{D}, \widetilde{\mathbf{P}}, \widetilde{\mathbf{X}}) = \|\widetilde{\mathbf{P}}\widetilde{\mathbf{Y}} - \mathbf{D}\widetilde{\mathbf{X}}\|_F^2 + \mu\|\widetilde{\mathbf{P}}\widetilde{\mathbf{Y}} - \mathbf{D}\widetilde{\mathbf{X}}_{\mathbf{in}}\|_F^2 \\ + \nu\|\mathbf{D}\widetilde{\mathbf{X}}_{\mathbf{out}}\|_F^2, \end{aligned} \tag{9}$$

where μ and ν are the weights given to the discriminative terms, and matrices $\widetilde{\mathbf{X}}_{\mathbf{in}}$ and $\widetilde{\mathbf{X}}_{\mathbf{out}}$ are given as:

$$\widetilde{\mathbf{X}}_{\mathbf{in}}[i, j] = \begin{cases} \widetilde{\mathbf{X}}[i, j], & \mathbf{D_i}, \widetilde{\mathbf{Y}}_{\mathbf{j}} \in \text{same class} \\ 0, & \text{otherwise,} \end{cases}$$

$$\widetilde{\mathbf{X}}_{\mathbf{out}}[i, j] = \begin{cases} \widetilde{\mathbf{X}}[i, j], & \mathbf{D_i}, \widetilde{\mathbf{Y}}_{\mathbf{j}} \in \text{different class} \\ 0, & \text{otherwise.} \end{cases}$$

SDDL achieved appealing performance on face recognition and object recognition tasks. However, it has no direct connection to the classification rule of SRC, which predicts the class label of query sample using the reconstruction residual. SDDL considers only the within-class reconstruction residual while ignoring the between-class reconstruction information during the learning process. This makes the projections of data and dictionary lack of discriminative power in the reduced space, thus the learned sparse representation may not be optimal for domain adaptation classification.

III. Proposed Method

A. Formulation

In this section, we introduce a novel domain adaptation dictionary learning method (OCPD-SRC). Based on the decision rule of SRC, our method not only learns coupled projections of data from both domains but also obtains a latent discriminative dictionary simultaneously. It aims to maximize the between-class sparse reconstruction residuals of data in different domains and minimize the within-class sparse reconstruction residuals of data in the projected common space. Therefore, SRC can achieve the optimum performance in the shared common subspace.

For each training sample from the source domain where $\mathbf{y}^{\mathbf{1}}_{i,j}$ is the j-th sample of class i, we leave it out from the training set and use the remaining training samples to linearly represent it. By solving the ℓ_1 optimization problem, we find a representation coefficient vector $\boldsymbol{\alpha}^{\mathbf{1}}_{i,j}$. Then we define the within-class sparse reconstruction residual of source domain

in the projected space J_w^1 as follows

$$J_w^1 = tr\left(\sum_{i=1}^{c}\sum_{j=1}^{n_i^1}(\mathbf{P_1}\mathbf{y}_{i,j}^1 - \mathbf{D}\delta_i(\boldsymbol{\alpha}_{i,j}^1))(\mathbf{P_1}\mathbf{y}_{i,j}^1 - \mathbf{D}\delta_i(\boldsymbol{\alpha}_{i,j}^1))^T\right)$$
$$= tr\left((\mathbf{P_1}\mathbf{Y_1} - \mathbf{D}\boldsymbol{\Lambda}_w^1)(\mathbf{P_1}\mathbf{Y_1} - \mathbf{D}\boldsymbol{\Lambda}_w^1)^T\right)$$
$$= \|\mathbf{P_1}\mathbf{Y_1} - \mathbf{D}\boldsymbol{\Lambda}_w^1\|_F^2, \tag{10}$$

where $\boldsymbol{\Lambda}_w^1 = [\delta_1(\boldsymbol{\alpha}_{1,1}^1), \delta_1(\boldsymbol{\alpha}_{1,2}^1), \cdots, \delta_c(\boldsymbol{\alpha}_{c,n_i^1}^1)] \in R^{K\times N_1}$, n_i^1 is the number of training samples of class i from the source domain, $N_1 = \sum_{i=1}^{c} n_i^1$. $\delta_i(\cdot)$ is the characteristic function which selects the coefficients associated with class i.

Similarly, we define J_b^1 to evaluate the between-class sparse reconstruction residual of source domain in the projected space

$$J_b^1 = tr\left(\sum_{i=1}^{c}\sum_{j=1}^{n_i^1}\sum_{s\neq i}(\mathbf{P_1}\mathbf{y}_{i,j}^1 - \mathbf{D}\delta_s(\boldsymbol{\alpha}_{i,j}^1))(\mathbf{P_1}\mathbf{y}_{i,j}^1 - \mathbf{D}\delta_s(\boldsymbol{\alpha}_{i,j}^1))^T\right)$$
$$= \|\mathbf{P_1}\mathbf{Y_1} - \mathbf{D}\boldsymbol{\Lambda}_b^1\|_F^2, \tag{11}$$

where $\boldsymbol{\Lambda}_b^1 = [\delta_s(\boldsymbol{\alpha}_{1,1}^1), \delta_s(\boldsymbol{\alpha}_{1,2}^1), \cdots, \delta_s(\boldsymbol{\alpha}_{c,n_i^1}^1)] \in R^{K\times N_1}$, $\delta_s(\boldsymbol{\alpha}_{i,j}^1)$ is a vector whose only nonzero entries are the entries in $\boldsymbol{\alpha}_{i,j}^1$ associated with class s, $s \neq i$.

In the same manner, the within-class and between-class sparse reconstruction residuals of target domain in the projected low-dimensional space can be defined as

$$J_w^2 = tr\left(\sum_{i=1}^{c}\sum_{j=1}^{n_i^2}(\mathbf{P_2}\mathbf{y}_{i,j}^2 - \mathbf{D}\delta_i(\boldsymbol{\alpha}_{i,j}^2))(\mathbf{P_2}\mathbf{y}_{i,j}^2 - \mathbf{D}\delta_i(\boldsymbol{\alpha}_{i,j}^2))^T\right)$$
$$= \|\mathbf{P_2}\mathbf{Y_2} - \mathbf{D}\boldsymbol{\Lambda}_w^2\|_F^2, \tag{12}$$

and

$$J_b^2 = tr\left(\sum_{i=1}^{c}\sum_{j=1}^{n_i^2}\sum_{s\neq i}(\mathbf{P_2}\mathbf{y}_{i,j}^2 - \mathbf{D}\delta_s(\boldsymbol{\alpha}_{i,j}^2))(\mathbf{P_2}\mathbf{y}_{i,j}^2 - \mathbf{D}\delta_s(\boldsymbol{\alpha}_{i,j}^2))^T\right)$$
$$= \|\mathbf{P_2}\mathbf{Y_2} - \mathbf{D}\boldsymbol{\Lambda}_b^2\|_F^2, \tag{13}$$

where $\boldsymbol{\Lambda}_w^2 = [\delta_1(\boldsymbol{\alpha}_{1,1}^2), \delta_1(\boldsymbol{\alpha}_{1,2}^2), \cdots, \delta_c(\boldsymbol{\alpha}_{c,n_i^2}^2)] \in R^{K\times N_2}$ and $\boldsymbol{\Lambda}_b^2 = [\delta_s(\boldsymbol{\alpha}_{1,1}^2), \delta_s(\boldsymbol{\alpha}_{1,2}^2), \cdots, \delta_s(\boldsymbol{\alpha}_{c,n_i^2}^2)] \in R^{K\times N_2}$, n_i^2 is the number of training samples of class i from the target domain, $N_2 = \sum_{i=1}^{c} n_i^2$. $\delta_i(\boldsymbol{\alpha}_{i,j}^2)$ and $\delta_s(\boldsymbol{\alpha}_{i,j}^2)$ are the vectors whose only nonzero entries are the entries in $\boldsymbol{\alpha}_{i,j}^2$ associated with class i and s, $s \neq i$, respectively.

In our method, we aim to maximize the between-class sparse reconstruction residuals of data from both domains in the transformed space. Thus, we maximize the following cost function

$$\max J_b = \max\{J_b^1 + J_b^2\}$$
$$= \max\{\|\mathbf{P_1}\mathbf{Y_1} - \mathbf{D}\boldsymbol{\Lambda}_b^1\|_F^2 + \|\mathbf{P_2}\mathbf{Y_2} - \mathbf{D}\boldsymbol{\Lambda}_b^2\|_F^2\}$$
$$= \max \|\widetilde{\mathbf{P}}\widetilde{\mathbf{Y}} - \mathbf{D}\widetilde{\boldsymbol{\Lambda}}_b\|_F^2, \tag{14}$$

and simultaneous minimize the within-class sparse reconstruction residuals of data from both two domains in the transformed space

$$\min J_w = \min\{J_w^1 + J_w^2\}$$
$$= \min \|\widetilde{\mathbf{P}}\widetilde{\mathbf{Y}} - \mathbf{D}\widetilde{\boldsymbol{\Lambda}}_w\|_F^2, \tag{15}$$

where $\widetilde{\mathbf{P}} = [\mathbf{P_1}, \mathbf{P_2}]$, $\widetilde{\mathbf{Y}} = \begin{bmatrix}\mathbf{Y_1} & \mathbf{0}\\ \mathbf{0} & \mathbf{Y_2}\end{bmatrix}$, $\widetilde{\boldsymbol{\Lambda}}_b = [\boldsymbol{\Lambda}_b^1, \boldsymbol{\Lambda}_b^2]$ and $\widetilde{\boldsymbol{\Lambda}}_w = [\boldsymbol{\Lambda}_w^1, \boldsymbol{\Lambda}_w^2]$. Thus, we learn $\widetilde{\mathbf{P}}$ and $\mathbf{D}$ by maximizing the following objective function

$$J(\widetilde{\mathbf{P}}, \mathbf{D}) = \max_{\widetilde{\mathbf{P}},\mathbf{D}} \frac{tr\left((\widetilde{\mathbf{P}}\widetilde{\mathbf{Y}} - \mathbf{D}\widetilde{\boldsymbol{\Lambda}}_b)(\widetilde{\mathbf{P}}\widetilde{\mathbf{Y}} - \mathbf{D}\widetilde{\boldsymbol{\Lambda}}_b)^T\right)}{tr\left((\widetilde{\mathbf{P}}\widetilde{\mathbf{Y}} - \mathbf{D}\widetilde{\boldsymbol{\Lambda}}_w)(\widetilde{\mathbf{P}}\widetilde{\mathbf{Y}} - \mathbf{D}\widetilde{\boldsymbol{\Lambda}}_w)^T\right)}$$
$$s.t.\ \mathbf{P_1}\mathbf{P_1}^T = \mathbf{P_2}\mathbf{P_2}^T = \mathbf{I}. \tag{16}$$

Here, we require $\mathbf{P_1}$ and $\mathbf{P_2}$ are orthogonal. We show that this leads to an efficient scheme for optimization and makes the kernelization of the algorithm possible.

B. Multiple Domains

The above formulation can be extended so that is can handle multiple domains. For the $\mathbf{M}$ domains problem, we simply construct matrices $\widetilde{\mathbf{P}}$, $\widetilde{\mathbf{Y}}$, $\widetilde{\boldsymbol{\Lambda}}_b$, $\widetilde{\boldsymbol{\Lambda}}_w$ as $\widetilde{\mathbf{P}} = [\mathbf{P_1}, \cdots, \mathbf{P_M}]$, $\widetilde{\mathbf{Y}} = \begin{bmatrix}\mathbf{Y_1} & \cdots & \mathbf{0}\\ \vdots & \ddots & \vdots\\ \mathbf{0} & \cdots & \mathbf{Y_M}\end{bmatrix}$, $\widetilde{\boldsymbol{\Lambda}}_b = [\boldsymbol{\Lambda}_b^1, \cdots, \boldsymbol{\Lambda}_b^M]$, and $\widetilde{\boldsymbol{\Lambda}}_w = [\boldsymbol{\Lambda}_w^1, \cdots, \boldsymbol{\Lambda}_w^M]$.

With these definitions, Eq.(16) can be extended to multiple domains as follows

$$J(\widetilde{\mathbf{P}}, \mathbf{D}) = \max_{\widetilde{\mathbf{P}},\mathbf{D}} \frac{tr\left((\widetilde{\mathbf{P}}\widetilde{\mathbf{Y}} - \mathbf{D}\widetilde{\boldsymbol{\Lambda}}_b)(\widetilde{\mathbf{P}}\widetilde{\mathbf{Y}} - \mathbf{D}\widetilde{\boldsymbol{\Lambda}}_b)^T\right)}{tr\left((\widetilde{\mathbf{P}}\widetilde{\mathbf{Y}} - \mathbf{D}\widetilde{\boldsymbol{\Lambda}}_w)(\widetilde{\mathbf{P}}\widetilde{\mathbf{Y}} - \mathbf{D}\widetilde{\boldsymbol{\Lambda}}_w)^T\right)}$$
$$s.t.\ \mathbf{P_i}\mathbf{P_i}^T = \mathbf{I}, \quad \forall \mathbf{i} = \mathbf{1}, \mathbf{2}, \cdots, \mathbf{M}. \tag{17}$$

Similar to [38] and [39], we can prove the following proposition for the above optimization problem. The proof is given in the **Appendix A**.

Proposition 1: There exists an optimal solution $\mathbf{P_1^*}, \mathbf{P_2^*}, \cdots, \mathbf{P_M^*}, \mathbf{D}^*$ to Eq.(17), which has he following form:

$$\mathbf{P_i^*} = (\mathbf{Y_i}\mathbf{A_i})^T, \quad \forall \mathbf{i} = \mathbf{1}, \mathbf{2}, \cdots, \mathbf{M}, \tag{18}$$
$$\mathbf{D}^* = \widetilde{\mathbf{P}}^*\widetilde{\mathbf{Y}}\widetilde{\mathbf{B}}, \tag{19}$$

where $\widetilde{\mathbf{P}}^* = [\mathbf{P_1^*}, \mathbf{P_2^*}, \cdots, \mathbf{P_M^*}]$, for some $\mathbf{A_i} \in R^{N_i\times m_f}$ and some $\widetilde{\mathbf{B}} \in R^{\sum N_i\times K}$.

With this proposition, the objective function can be written as

$$J(\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}}) = \frac{tr(\widetilde{\mathbf{A}}^T\widetilde{\mathbf{K}}\widetilde{\mathbf{S}}_b\widetilde{\mathbf{K}}^T\widetilde{\mathbf{A}})}{tr(\widetilde{\mathbf{A}}^T\widetilde{\mathbf{K}}\widetilde{\mathbf{S}}_w\widetilde{\mathbf{K}}^T\widetilde{\mathbf{A}})}, \tag{20}$$

where $\widetilde{\mathbf{K}} = \widetilde{\mathbf{Y}}^T\widetilde{\mathbf{Y}}$ and $\widetilde{\mathbf{A}}^T = [\widetilde{\mathbf{A}}_\mathbf{1}^T, \cdots, \widetilde{\mathbf{A}}_\mathbf{M}^T]$. $\widetilde{\mathbf{S}}_b = (\mathbf{I} - \widetilde{\mathbf{B}}\widetilde{\boldsymbol{\Lambda}}_b)(\mathbf{I} - \widetilde{\mathbf{B}}\widetilde{\boldsymbol{\Lambda}}_b)^T$ is the between-class scatter matrix, $\widetilde{\mathbf{S}}_w = (\mathbf{I} - \widetilde{\mathbf{B}}\widetilde{\boldsymbol{\Lambda}}_w)(\mathbf{I} - \widetilde{\mathbf{B}}\widetilde{\boldsymbol{\Lambda}}_w)^T$ is the with-class scatter matrix.

Here, the equality constraint now becomes

$$\mathbf{P_i}\mathbf{P_i}^T = \mathbf{A_i}^T\mathbf{K_i}\mathbf{A_i} = \mathbf{I}, \quad \forall \mathbf{i} = \mathbf{1}, \mathbf{2}, \cdots, \mathbf{M} \tag{21}$$

where $\mathbf{K_i} = \mathbf{Y_i}\mathbf{Y_i}$. Then, the objective function could be expressed as

$$J(\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}}) = \max_{\widetilde{\mathbf{A}},\widetilde{\mathbf{B}}} \frac{tr(\widetilde{\mathbf{A}}^T\widetilde{\mathbf{K}}\widetilde{\mathbf{S}}_b\widetilde{\mathbf{K}}^T\widetilde{\mathbf{A}})}{tr(\widetilde{\mathbf{A}}^T\widetilde{\mathbf{K}}\widetilde{\mathbf{S}}_w\widetilde{\mathbf{K}}^T\widetilde{\mathbf{A}})}$$
$$s.t.\ \mathbf{A_i}^T\mathbf{K_i}\mathbf{A_i} = \mathbf{I}, \quad \forall \mathbf{i} = \mathbf{1}, \mathbf{2}, \cdots, \mathbf{M}. \tag{22}$$

IV. Optimization

We adopt a standard iterative learning framework to jointly learning the desired projection $\widetilde{\mathbf{P}}$ via $\widetilde{\mathbf{A}}$ and the discriminative dictionary $\mathbf{D}$ via $\widetilde{\mathbf{B}}$. We divide the objective function in Eq.(22) into two sub-problems: (1) keeping $\widetilde{\mathbf{B}}$ fixed followed by updating $\widetilde{\mathbf{A}}$; and (2) keeping $\widetilde{\mathbf{A}}$ fixed followed by updating $\widetilde{\mathbf{B}}$. Note that, the optimization problem is non-convex. Still, numerical simulations have shown that the algorithm usually converges to a local maximum in a few iterations. The proposed algorithm is shown in Algorithm 1.

Step 1 (Learn $\widetilde{A}$ With Fixed $\widetilde{B}$): When $\widetilde{\mathbf{B}}$ is fixed, the optimization can be solved by using trace ratio optimization method [6], [43], [44].

In order to avoid over-fitting for trace ratio maximization problem, we add a regularization term in the denominator to ensure that $\widetilde{\mathbf{K}}\widetilde{\mathbf{S}}_w\widetilde{\mathbf{K}}^T + \mu\mathbf{I}$ is of full rank.

We know that there is a maximum ρ^* that is reached for certain $\widetilde{\mathbf{A}}^*$. Then, for any $\widetilde{\mathbf{A}}$, we have

$$\frac{tr(\widetilde{\mathbf{A}}^T\widetilde{\mathbf{K}}\widetilde{\mathbf{S}}_b\widetilde{\mathbf{K}}^T\widetilde{\mathbf{A}})}{tr\left(\widetilde{\mathbf{A}}^T(\widetilde{\mathbf{K}}\widetilde{\mathbf{S}}_w\widetilde{\mathbf{K}}^T+\mu\mathbf{I})\widetilde{\mathbf{A}}\right)} \leq \rho^*, \tag{23}$$

and hence,

$$tr\left(\widetilde{\mathbf{A}}^T\widetilde{\mathbf{K}}\widetilde{\mathbf{S}}_b\widetilde{\mathbf{K}}^T\widetilde{\mathbf{A}}\right) - \rho^* tr\left(\widetilde{\mathbf{A}}^T(\widetilde{\mathbf{K}}\widetilde{\mathbf{S}}_w\widetilde{\mathbf{K}}^T+\mu\mathbf{I})\widetilde{\mathbf{A}}\right) \leq 0. \tag{24}$$

This means that for ρ^* we have

$$tr\left(\widetilde{\mathbf{A}}^T(\widetilde{\mathbf{K}}\widetilde{\mathbf{S}}_b\widetilde{\mathbf{K}}^T - \rho^*(\widetilde{\mathbf{K}}\widetilde{\mathbf{S}}_w\widetilde{\mathbf{K}}^T+\mu\mathbf{I})\widetilde{\mathbf{A}}\right) \leq 0.$$

To optimize our objective function, we define a function

$$\begin{aligned} f(\rho) &= \max_{\widetilde{\mathbf{A}}} G(\widetilde{\mathbf{A}},\rho) \\ &= \max_{\widetilde{\mathbf{A}}} tr\left(\widetilde{\mathbf{A}}^T(\widetilde{\mathbf{K}}\widetilde{\mathbf{S}}_b\widetilde{\mathbf{K}}^T - \rho^*\widetilde{\mathbf{K}}\widetilde{\mathbf{S}}_w\widetilde{\mathbf{K}}^T - \rho\mu\mathbf{I})\widetilde{\mathbf{A}}\right) \\ &= \max_{\widetilde{\mathbf{A}}} tr\left(\widetilde{\mathbf{A}}^T(\widetilde{\mathbf{K}}(\widetilde{\mathbf{S}}_b - \rho\widetilde{\mathbf{S}}_w)\widetilde{\mathbf{K}}^T - \rho\mu\mathbf{I})\widetilde{\mathbf{A}}\right). \end{aligned} \tag{25}$$

Here, $f(\rho)$ has the following properties, which are proved in the **Appendix B**.

Lemma 1: (i). $f(\rho)$ is a decreasing function. (ii). $f(\rho) = 0$ iff $\rho = \rho^*$.

In addition, if ρ is negative and small enough, the matrix $\widetilde{\mathbf{K}}\widetilde{\mathbf{S}}_b\widetilde{\mathbf{K}}^T - \rho^*\widetilde{\mathbf{K}}\widetilde{\mathbf{S}}_w\widetilde{\mathbf{K}}^T - \rho\mu\mathbf{I}$ becomes a positive semi-definite matrix, hence $f(\rho)$ is bigger than 0. Similarly, if ρ is very big, the matrix $\widetilde{\mathbf{K}}\widetilde{\mathbf{S}}_b\widetilde{\mathbf{K}}^T - \rho^*\widetilde{\mathbf{K}}\widetilde{\mathbf{S}}_w\widetilde{\mathbf{K}}^T - \rho\mu\mathbf{I}$ becomes a negative semidefinite matrix, hence $f(\rho)$ is smaller than 0. Based on Lemma 1, we can see that ρ^* always exists.

Following [6] and [44], the root of the decreasing function $f(\rho)$ and the corresponding $\widetilde{\mathbf{A}}$ could be found by updating ρ and $\widetilde{\mathbf{A}}$ alternately during iterations. For a given ρ, let $\widetilde{\mathbf{A}}(\rho)$ be the solution of Eq.(25). We define a function

$$f'(\rho) = -tr\left(\widetilde{\mathbf{A}}(\rho)^T(\widetilde{\mathbf{K}}\widetilde{\mathbf{S}}_w\widetilde{\mathbf{K}}^T+\mu\mathbf{I})\widetilde{\mathbf{A}}(\rho)\right) \tag{26}$$

with $\widetilde{\mathbf{A}}(\rho)$, the root could be found by $\rho_{new} = \rho - \eta_1\frac{f(\rho)}{f'(\rho)}$, where η_1 is the step length. Now, let us describe how to find $\widetilde{\mathbf{A}}(\rho)$ for a given ρ.

With the constraint $\mathbf{A_i}^T\mathbf{K_i}\mathbf{A_i} = \mathbf{I}$, $\widetilde{\mathbf{A}}$ can be found by solving

$$\begin{aligned} &\max_{\widetilde{\mathbf{A}}}\ tr\left(\widetilde{\mathbf{A}}^T(\widetilde{\mathbf{K}}(\widetilde{\mathbf{S}}_b - \rho\widetilde{\mathbf{S}}_w)\widetilde{\mathbf{K}}^T - \rho\mu\mathbf{I})\widetilde{\mathbf{A}}\right). \\ &s.t.\ \mathbf{A_i}^T\mathbf{K_i}\mathbf{A_i} = \mathbf{I}, \quad \forall \mathbf{i} = \mathbf{1},\mathbf{2},\cdots,\mathbf{M}. \end{aligned} \tag{27}$$

In order to efficiently solve $\widetilde{\mathbf{A}}$, the formulation in Eq.(27) can be simplified expressed as follows

$$\begin{aligned} &\max_{\mathbf{G}}\ tr(\mathbf{G}^T\mathbf{H}\mathbf{G}) \\ &s.t.\ \mathbf{G_i}^T\mathbf{G_i} = \mathbf{I}, \quad \forall \mathbf{i} = \mathbf{1},\mathbf{2},\cdots,\mathbf{M}, \end{aligned} \tag{28}$$

where $\mathbf{H} = \mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{V}^T(\widetilde{\mathbf{K}}(\widetilde{\mathbf{S}}_b - \rho\widetilde{\mathbf{S}}_w)\widetilde{\mathbf{K}}^T - \rho\mu\mathbf{I})\mathbf{V}\mathbf{\Lambda}^{-\frac{1}{2}}$.

Proof: Let $\mathbf{G} = \mathbf{\Lambda}^{\frac{1}{2}}\mathbf{V}^T\widetilde{\mathbf{A}}$, where $\mathbf{V}$ and $\mathbf{\Lambda}$ come from the eigen decomposition of $\widetilde{\mathbf{K}} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$. Substituting $\mathbf{H}$ and $\mathbf{G}$ into Eq.(28), we get the required form of the optimization in Eq.(27). Thus, the solution for $\widetilde{\mathbf{A}}$ can be recovered simply by

$$\widetilde{\mathbf{A}} = \mathbf{V}\mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{G}. \tag{29}$$

Similar to [38], Eq.(28) can be solved efficiently using the algorithm proposed by [45]. The alternating projection steps for finding $\widetilde{\mathbf{A}}$ and ρ are described in Algorithm 2.

Algorithm 1 OCPD-SRC

Input: Training set $\{\mathbf{Y_i}\}_{\mathbf{i=1}}^{\mathbf{M}}$ and corresponding class label $\{c_\mathbf{i}\}_{\mathbf{i=1}}^{\mathbf{M}}$ for $\mathbf{M}$ domains, parameters λ, η_1, η_1, μ, dictionary size K and dimension m_f.

Initialization: Initialize $\widetilde{\mathbf{A}}$ such that $\widetilde{\mathbf{A}}_\mathbf{i}^T\mathbf{K_i}\widetilde{\mathbf{A}}_\mathbf{i} = \mathbf{I}$, $\forall \mathbf{i} = \mathbf{1},\mathbf{2},\cdots,\mathbf{M}$.
For this, find SVD of kernel matrix $\mathbf{K_i} = \mathbf{V_i}\mathbf{S_i}\mathbf{V_i}^T$, then set $\mathbf{A_i}$ as the matrix of eigenvectors with the first m_f largest eigenvalues as columns.
Random initialization $\widetilde{\mathbf{B}}$ and normalize the columns of $\widetilde{\mathbf{A}}^T\widetilde{\mathbf{K}}\widetilde{\mathbf{B}}$ and
$\mathbf{A_i}^T\mathbf{K_i}(\mathbf{Y_i},\mathbf{y})$ to have unit ℓ_2-norm.

Optimization

Repeat

1. For each sample from the source, leave it out and use the rest training samples to represent it and calculate its sparse representation coefficient vector by solving Eq.(36). Similar, calculating the coding for each labelled sample from the target domain.
2. Solve $\widetilde{\mathbf{A}}$ with fixed $\widetilde{\mathbf{B}}$ using Algorithm 2.
3. Solve $\widetilde{\mathbf{B}}$ with fixed $\widetilde{\mathbf{A}}$ via Eq.(35);

Until convergence

Output: Learning dictionary $\mathbf{D}$, and projection matrices $\{\mathbf{A}\}_{\mathbf{i=1}}^{\mathbf{M}}$.

Step 2 (Learn $\widetilde{B}$ With Fixed $\widetilde{A}$): We now assume that $\widetilde{\mathbf{A}}$ is fixed. The objective function can be written as:

$$J(\widetilde{\mathbf{B}}) = \max_{\widetilde{\mathbf{A}},\widetilde{\mathbf{B}}} \frac{tr(\widetilde{\mathbf{A}}^T\widetilde{\mathbf{K}}\widetilde{\mathbf{S}}_b\widetilde{\mathbf{K}}^T\widetilde{\mathbf{A}})}{tr(\widetilde{\mathbf{A}}^T\widetilde{\mathbf{K}}\widetilde{\mathbf{S}}_w\widetilde{\mathbf{K}}^T\widetilde{\mathbf{A}})}. \tag{30}$$

To learn the dictionary, we write the dictionary $\mathbf{D} = [\mathbf{D}_1,\mathbf{D}_2,\cdots,\mathbf{D}_c]$, where c is the number of classes. From the **Proposition 1**, we can get $\mathbf{D}_i = \widetilde{\mathbf{P}}\widetilde{\mathbf{Y}}\widetilde{\mathbf{B}}_i$ and $\mathbf{D}_s = \widetilde{\mathbf{P}}\widetilde{\mathbf{Y}}\widetilde{\mathbf{B}}_s$, where $\widetilde{\mathbf{B}} = [\widetilde{\mathbf{B}}_1,\widetilde{\mathbf{B}}_2,\cdots,\widetilde{\mathbf{B}}_c]$. $\mathbf{D}_i$ and $\mathbf{D}_s$ are the sub-dictionary

Algorithm 2 Alternating Projection

Input: $\widetilde{\mathbf{K}}$, $\widetilde{\mathbf{S}}_b$, $\widetilde{\mathbf{S}}_w$, step length η_1, maximum number of iterations t.
Initialize ρ.
Repeat
Compute $\widetilde{\mathbf{A}}$ by solving the corresponding eigenvalue problem in Eq.(27).
Compute$\rho = \rho - \eta_1 \frac{f(\rho)}{f'(\rho)}$.
Until $f(\rho) = 0$ or t reached.
Output: $\widetilde{\mathbf{A}}$.

associated with class i and s, respectively. Our objective function in Eq.(30) can be rewritten as

$$J(\widetilde{\mathbf{B}}_i) = \max_{\mathbf{B}_i} \sum_{i=1}^{c} \frac{tr(\widetilde{\mathbf{A}}^T \widetilde{\mathbf{S}}_b^i \widetilde{\mathbf{A}})}{tr(\widetilde{\mathbf{A}}^T \widetilde{\mathbf{S}}_w \widetilde{\mathbf{A}})}, \tag{31}$$

where $\widetilde{\mathbf{S}}_b^i = \sum_{s \neq i}(\widetilde{\mathbf{K}}_i - \widetilde{\mathbf{K}}\widetilde{\mathbf{B}}_s\widetilde{\mathbf{\Lambda}}_b^s)(\widetilde{\mathbf{K}}_i - \widetilde{\mathbf{K}}\widetilde{\mathbf{B}}_s\widetilde{\mathbf{\Lambda}}_b^s)^T$, $\widetilde{\mathbf{S}}_w = \sum_{i=1}^{c}\widetilde{\mathbf{S}}_w^i = \sum_{i=1}^{c}(\widetilde{\mathbf{K}}_i - \widetilde{\mathbf{K}}\widetilde{\mathbf{B}}_i\widetilde{\mathbf{\Lambda}}_b^i)(\widetilde{\mathbf{K}}_i - \widetilde{\mathbf{K}}\widetilde{\mathbf{B}}_i\widetilde{\mathbf{\Lambda}}_b^i)^T$ and $\widetilde{\mathbf{K}}_i = \widetilde{\mathbf{Y}}^T\widetilde{\mathbf{Y}}_i$. $\widetilde{\mathbf{\Lambda}}_b^s = [\mathbf{\Lambda}_b^{\mathbf{1},s}, \mathbf{\Lambda}_b^{\mathbf{2},s} \cdots, \mathbf{\Lambda}_b^{\mathbf{M},s}]$ and $\widetilde{\mathbf{\Lambda}}_w^i = [\mathbf{\Lambda}_w^{\mathbf{1},i}, \mathbf{\Lambda}_w^{\mathbf{2},i}, \cdots, \mathbf{\Lambda}_w^{\mathbf{M},i}]$ are coding coefficient matrices with respect to class s and class i, $s \neq i$, respectively. These two objective functions Eq.(30) and Eq.(31) are the same, but they are formulated in a different way for the convenience of optimization. (Please see **Appendix C**).

Here, $\mathbf{\Lambda}_b^{\mathbf{1},s} = [\boldsymbol{\alpha}_{1,1}^{\mathbf{1},s}, \boldsymbol{\alpha}_{1,2}^{\mathbf{1},s}, \cdots, \boldsymbol{\alpha}_{c,n_c^{\mathbf{1}}}^{\mathbf{1},s}]$, where $\boldsymbol{\alpha}_{i,j}^{\mathbf{1},s}$ is the representation coefficient vector associated with class s, $s \neq i$ from domain $\mathbf{1}$, $i = 1, \cdots, c$, $j = 1, \cdots, n_i^{\mathbf{1}}$. $\mathbf{\Lambda}_w^{\mathbf{1},i} = [\boldsymbol{\alpha}_{1,1}^{\mathbf{1},i}, \boldsymbol{\alpha}_{1,2}^{\mathbf{1},i}, \cdots, \boldsymbol{\alpha}_{c,n_c^{\mathbf{1}}}^{\mathbf{1},i}]$, where $\boldsymbol{\alpha}_{i,j}^{\mathbf{1},i}$ is the representation coefficient vector associated with class i.

We update $\widetilde{\mathbf{B}}$ class by class sequentially. When updating $\widetilde{\mathbf{B}}_i$, $\widetilde{\mathbf{B}}_s$, $s \neq i$ associated to the other class will be fixed. The optimization scheme is based on gradient ascent. Apply the chain rule, we have

$$\nabla_{\widetilde{\mathbf{B}}_i} J(\widetilde{\mathbf{B}}_i) = \frac{\partial J(\widetilde{\mathbf{B}}_i)}{\partial \widetilde{\mathbf{S}}_b^i}\frac{\partial \widetilde{\mathbf{S}}_b^i}{\partial \widetilde{\mathbf{B}}_i} + \frac{\partial J(\widetilde{\mathbf{B}}_i)}{\partial \widetilde{\mathbf{S}}_w}\frac{\partial \widetilde{\mathbf{S}}_w}{\partial \widetilde{\mathbf{B}}_i}. \tag{32}$$

Since there is no relations between $\widetilde{\mathbf{S}}_b^i$ and $\widetilde{\mathbf{B}}_i$, $\partial\widetilde{\mathbf{S}}_b^i/\partial\widetilde{\mathbf{B}}_i = 0$. It is easy to find that

$$\frac{\partial J(\widetilde{\mathbf{B}}_i)}{\partial \widetilde{\mathbf{S}}_w} = \frac{-tr(\widetilde{\mathbf{A}}^T\widetilde{\mathbf{S}}_b^i\widetilde{\mathbf{A}})\widetilde{\mathbf{A}}^T\widetilde{\mathbf{A}}}{\left(tr(\widetilde{\mathbf{A}}^T\widetilde{\mathbf{S}}_w\widetilde{\mathbf{A}})\right)^2}, \tag{33}$$

$$\frac{\partial \widetilde{\mathbf{S}}_w}{\partial \widetilde{\mathbf{B}}_i} = 2\widetilde{\mathbf{K}}(\widetilde{\mathbf{\Lambda}}_w^i)^T(\widetilde{\mathbf{K}}\widetilde{\mathbf{B}}_i\widetilde{\mathbf{\Lambda}}_w^i - \widetilde{\mathbf{K}}_i). \tag{34}$$

With the gradient in Eq.(32) calculated, we employ a projected gradient ascent procedure for updating $\widetilde{\mathbf{B}}_i$

$$\widetilde{\mathbf{B}}_i = \widetilde{\mathbf{B}}_i + \eta_2 \nabla_{\widetilde{\mathbf{B}}_i} J(\widetilde{\mathbf{B}}_i). \tag{35}$$

where η_2 is the step size for updating the sub-dictionary $\widetilde{\mathbf{B}}_i$.

When $\widetilde{\mathbf{A}}$ and $\widetilde{\mathbf{B}}$ have been learned, we then calculate the sparse representation coefficient for each input sample from the source and target domains in the transformed space.

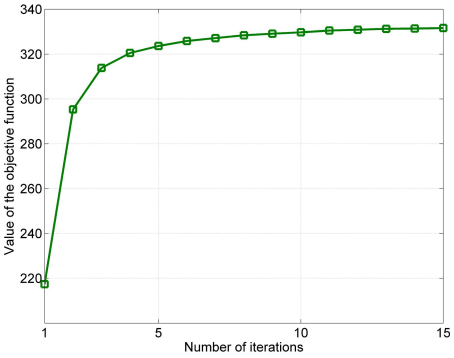


Fig. 3. Objective function versus number of iterations on CMU Multi-Pie dataset.

Eq.(2) can be changed into the following formula:

$$\begin{aligned}\hat{\boldsymbol{\alpha}} &= \arg\min_{\boldsymbol{\alpha}} \|\mathbf{P_i y} - \widetilde{\mathbf{P}}\widetilde{\mathbf{Y}}\widetilde{\mathbf{B}}\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1 \\ &= \arg\min_{\boldsymbol{\alpha}} \|\mathbf{A_i}^T\mathbf{K_i}(\mathbf{Y_i}, \mathbf{y}) - \widetilde{\mathbf{A}}^T\widetilde{\mathbf{K}}\widetilde{\mathbf{B}}\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1, \end{aligned} \tag{36}$$

where $\mathbf{K_i}(\mathbf{Y_i}, \mathbf{y}) = \mathbf{Y_i}^T\mathbf{y}$ and $\widetilde{\mathbf{K}} = \widetilde{\mathbf{Y}}^T\widetilde{\mathbf{Y}}$. We repeat the above two steps until the algorithm is convergent. As discussed earlier, our method is non-convex and often converges to a local maximum in a few iterations. To empirically show the convergence of our method, Fig. 3 shows the curve of the objective function versus the number of iterations. From Fig. 3 we can see that our method can achieve stable performance in a few iterations.

A. Nonlinear Extension

In many computer vision tasks, linear representations are inadequate, in particular, when the underlying data structure is often nonlinear. There are several approaches that can deal with nonlinear data [6], [9]. These essentially map the nonlinear data into high dimensional feature spaces using the kernel trick [46] such that samples of the same classes are easily grouped together and are linearly separable. We adopt the use of Mercer kernels to extend our analysis to the nonlinear case.

Let $\phi : \mathbf{y} \to \phi(\mathbf{y})$ be a nonlinear mapping function from original feature space to the high dimensional feature space $\mathcal{H}$. Then the source domain and target domain data in $\mathcal{H}$ can be expressed as $\phi(\mathbf{Y_1})$ and $\phi(\mathbf{Y_2})$, respectively. Since the feature space $\mathcal{H}$ has a very high or possibly infinite dimensional, it is necessary to perform dimensionality reduction in $\mathcal{H}$. The projection $\mathcal{P}_\mathbf{i}$ from the high dimensional space to the reduced space is no longer linear. Similar to proposition 1, by letting $\mathcal{K} = \langle \boldsymbol{\phi}(\widetilde{\mathbf{Y}}), \boldsymbol{\phi}(\widetilde{\mathbf{Y}})\rangle$, we can show that:

$$\mathcal{P}_\mathbf{i} = \mathbf{A_i}^T\boldsymbol{\phi}(\mathbf{Y_i})^T$$

and

$$\mathbf{D} = \widetilde{\mathbf{A}}^T\mathcal{K}\widetilde{\mathbf{B}}.$$

Similar to the linear case, we get the objective function as

$$\begin{aligned} J(\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}}) = \max_{\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}}} \ & \frac{tr(\widetilde{\mathbf{A}}^T\mathcal{K}\widetilde{\mathbf{S}}_b\mathcal{K}^T\widetilde{\mathbf{A}})}{tr(\widetilde{\mathbf{A}}^T\mathcal{K}\widetilde{\mathbf{S}}_w\mathcal{K}^T\widetilde{\mathbf{A}})} \\ s.t. \ & \widetilde{\mathbf{A}}_\mathbf{i}^T\mathcal{K}_\mathbf{i}\widetilde{\mathbf{A}}_\mathbf{i} = \mathbf{I}, \quad \forall \mathbf{i} = \mathbf{1}, \mathbf{2}, \cdots, \mathbf{M}. \end{aligned} \tag{37}$$

V. Classification

When the projections $\mathbf{P_i}$ and dictionary $\mathbf{D}$ are learned, we project the testing sample via $\mathbf{P_i}$ and encode the projected sample over the learned dictionary in the projected low-dimensional space. Once we obtain the coding coefficients $\hat{\boldsymbol{\alpha}}$, the reconstruction residual for each class can be used for classification. Given a testing sample, $\mathbf{y}_{te}$ from domain $\mathbf{k}$, we propose the following steps for classification. Similar to [39], we will consider the nonlinear setting.

1) Compute the embedding of the testing sample in the low-dimensional subspace using the corresponding projections $\mathbf{P_k}$

$$\mathbf{z}_{te} = \mathbf{P_k}\boldsymbol{\phi}(\mathbf{y}_{te}) = \mathbf{A}_\mathbf{k}^T\mathcal{K}_{te}, \tag{38}$$

where $\mathcal{K}_{te} = \langle\boldsymbol{\phi}(\mathbf{Y_k}), \boldsymbol{\phi}(\mathbf{y}_{te})\rangle$.

2) Compute the coding coefficient $\hat{\boldsymbol{\alpha}}_{te}$ over dictionary $\mathbf{D}$ by solving the following optimization problem

$$\hat{\boldsymbol{\alpha}}_{te} = \arg\min_{\boldsymbol{\alpha}_{te}} \|\mathbf{z}_{te} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1, \tag{39}$$

where λ is the regularization parameter to control the sparsity of $\boldsymbol{\alpha}$.

3) After obtaining the coding coefficients $\hat{\boldsymbol{\alpha}}_{te}$, the classification can be performed using minimum class-wise reconstruction residual, we classify the testing sample via

$$label(\mathbf{y}_{te}) = \arg\min_{i=1,\cdots,c} \|\mathbf{z}_{te} - \mathbf{D}_i\hat{\boldsymbol{\alpha}}_{te}^i\|_2^2, \tag{40}$$

where $\hat{\boldsymbol{\alpha}}_{te}^i$ is sparse code associated with the i-th class.

VI. Experiments

We evaluate the performance of OCPD-SRC by using two typical applications including face recognition and object recognition. For face recognition, performance are evaluated on the well-known CMU Multi-Pie dataset. Then we perform our method on domain adaptation datasets and compare it with existing adaptation algorithms.

A. Parameter Setting

We have four parameters, λ, η_1, η_2 and μ in the proposed OCPD-SRC model. To achieve the best performance, in all experiments, the sparsity regularization parameter λ is determined by 5-fold cross-validation on the training data and fixed for each dataset. We found that using $\lambda = 0.05$ for training and $\lambda = 0.01$ for testing yield a better performance. η_2 is the step size for updating the sub-dictionary $\widetilde{\mathbf{B}}_i$, η_1 is the step length for updating ρ. μ is a regularization parameter. We empirically set the parameters $\eta_1 = 1, \eta_2 = 0.1$, which work well in all of our experiments. The parameter μ is set to be 0.001 for all experiments. For all the compared methods, we use their original settings provided in the corresponding papers.

B. Face Recognition

The CMU Multi-Pie dataset [61] contains images of 337 subjects captured in 4 sessions with simultaneous variation in 15 poses, 6 expressions and 20 illuminations. Following the same experimental protocol [38], we use 129 subjects common to both Session 1 and Session 2. The experiment is done on 5 poses, ranging from frontal to 75°. Frontal faces are taken as the source domain, while different off-frontal poses are taken as target domains. Dictionaries are trained using illuminations {1, 4, 7, 12, 17} from the source and the target poses, in Session 1 per subject. All the illumination images from Session 2, for the target pose, are taken as the testing image. Fig. 4 shows some images of one person under different poses (frontal and 60°) and different illuminations {1, 4, 7, 12, 17} from session 1. We set the dictionary size is 5 and the final dimension is 140.

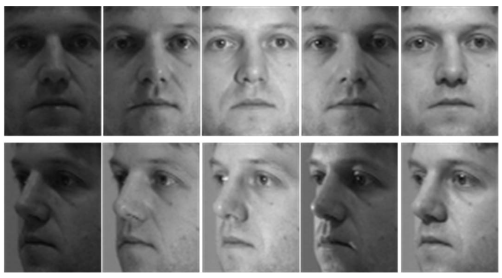


Fig. 4. Some images of one person under different poses and illuminations.

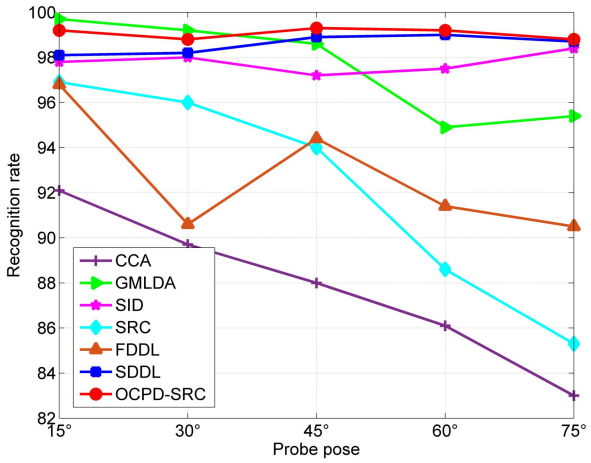


Fig. 5. Recognition rate of the proposed method with other algorithms for face recognition across poses.

Fig. 5 shows the results of our method and several recently proposed multi-view recognition algorithms [62]. It can be seen from Fig. 5 that our method achieves the best results in most of cases and outperforms SDDL at various poses. SRC is less effective and its performance decreases rapidly with an increase in the level of rotation. This indicates that the sparse coding framework is insufficient when the testing data has different characteristics than the data used for training. FDDL also is not optimal here as it is not designed based on domain adaptation.

C. Object Recognition

The proposed method is evaluated using a recent domain adaptation dataset which was created by combining the Office dataset [23] with Caltech-256 [47] dataset. The Office dataset contains 3 domains: Amazon, Webcam and DSLR. In each domain there are a total of 31 categories,[1] such as headphones,

[1]The 31 categories in the database are: backpack, bike, bike helmet, bookcase, bottle, calculator, desk chair, desk lamp, computer, file cabinet, headphones, keyboard, laptop, letter tray, mobile phone, monitor, mouse, mug, notebook, pen, phone, printer, projector, puncher, ring binder, ruler, scissors, speaker, stapler, tape, and trash can.

TABLE I
RECOGNITION RATES (%) ON SINGLE SOURCE DOMAIN ADAPTATION

Methods	C → A	C → D	A → C	A → W	W → C	W → A	D → A	D → W
SRC [1]	42.7±1.2	63.4±0.9	24.3±1.3	61.2±2.1	22.6±1.2	38.0±1.7	33.8±2.0	63.1±1.9
FDDL [15]	39.3±2.9	55.0±2.8	24.3±2.2	50.4±3.5	22.9±2.6	41.1±2.6	36.7±2.5	65.9±4.9
Metric [23]	33.7±0.8	35.0±1.1	27.3±0.7	36.0±1.0	21.7±0.5	32.3±0.8	30.3±0.8	55.6±0.7
SGF [22]	40.2±0.7	36.6±0.8	37.7±0.5	37.9±0.7	29.2±0.7	38.2±0.6	39.2±0.7	69.5±0.9
GFK [24]	46.1±0.6	55.0±0.9	39.6±0.4	56.9±1.0	32.8±0.1	46.2±0.6	46.2±0.6	**80.2±0.4**
HFA [28]	45.5±0.9	51.9±1.1	31.1±0.6	58.6±1.0	31.1±0.6	45.9±0.7	45.8±0.9	62.1±0.7
DASRC [36]	54.3±2.7	77.1±3.4	37.6±2.8	71.6±3.9	28.2±2.1	44.4±1.3	46.0±2.2	71.3±1.7
SDDL [39]	49.5±2.6	76.7±3.9	27.4±2.4	72.0±4.8	29.7±1.9	49.4±2.1	48.9±3.8	72.6±2.1
OCPD-SRC	**61.1±1.8**	**79.5±3.6**	**44.3±2.1**	**75.4±4.2**	**45.8±2.1**	**62.1±1.9**	**55.3±2.6**	78.7±2.6
DASH-N [63]	71.6±2.2	81.4±3.5	54.9±1.8	75.5±4.2	50.2±3.3	70.4±3.2	68.9±2.9	77.1±2.8
OCPD-SRC(Hierarchical)	**75.3±1.9**	**84.7±3.3**	**58.6±2.4**	**79.1±3.7**	**52.3±2.4**	**75.7±2.7**	**71.2±2.8**	**80.8±2.5**

Fig. 6. Sample images from the Headphones and Computer-Monitor categories in Amazon, DSLR, Webcam and Caltech-256. Amazon and Caltech-256 datasets have diverse images; DSLR and Webcam are similar datasets with mostly images from offices.

monitor, keyboard, cycle, ect. To validate the proposed method on a wide range of datasets, we use the Caltech 256 as the fourth domain. The Caltech 256 dataset [47] contains 30,607 images of 256 categories. There are at least 80 images per category. Fig. 6 shows some sample images from these datasets, and clearly highlights the differences between them.

In order to clearly illustrate the advantage of OCPD-SRC, we compare our method with state-of-the-art adaptation algorithms such as [22]–[24], [28], [36], and [39] and two other non-domain adaptation methods SRC [1] and FDDL [15]. In addition, the results obtained by using DASH-N [63] are also included in the comparison. By using the idea of hierarchical networks, DASH-N jointly learns a hierarchy of features together with transformations that address the mismatch between different domains. For fair comparison, similar to DASH-N, in our experiments we use a two-layer networks to learn the feature representation and then perform the classification using the concatenated features, denoted as OCPD-SRC (hierarchical). For each dataset, the average recognition accuracy is used as the criterion to compare the performances of different state-of-the-art algorithms, and we denote these four domains as A, C, W and D for Amazon, Caltech 256, Webcam and DSLR respectively. Furthermore, in order to test the ability of the proposed method to a wide range of domains, we create two new datasets by performing half-toning [48] and edge detection [49] algorithms on images from the Office dataset. The following describes the details of the experiments and results.

1) Experimental Setup: Following the experiment setting in [38], we evaluate the proposed method by using three step-ups. In the first setup, 10 overlapping categories: Backpack, Touring Bike, Calculator, Headphones, Computer-Keyboard, Laptop 101, Computer Monitor, Computer Mouse, Coffee Mug, and Video Projector, between the Office dataset and Caltech 256 dataset are used. There are 8 to 151 samples per category per domain, and 2533 images in total. In the second setup, all 31 categories from Amazon, Webcam, and DSLR are used to evaluate various algorithms. Finally, we evaluate our method for adaptation using multiple domains. In the third cases, if the source domain is Amazon or Caltech, 20 samples per category are selected for training, Otherwise, only 8 training samples per category are selected for DSLR and Webcam. 3 training samples for all of them when used for target domain. The remaining images from the target domain are used for testing. We randomly split the training and testing datasets, and repeat each experiment 20 times for each pair of source and target domains.

In all our experiments, we use the precomputed 800-bin SURF (Speeded-Up Robust Features) features provided in [17] and [23] for all the dataset. The simple non-parametric histogram intersection kernel is used in our method. When compared with DASH-N, we follow the experimental setup of [63] and cross-validation on the training data is performed to obtain the optimal parameters.

2) Single-Source Domain Adaptation Experiment: In this experiment, the number of dictionary atoms is set as 50, i.e., five atoms per category. SRC and DASRC use all the training samples as the dictionary. The feature dimension after projection is set as 65. The recognition results of different algorithms on 8 pairs of source-target domains are shown in Table I. From Table I, we can see that OCPD-SRC is consistently better than that of other algorithms for all domain pairs except the results obtained for DLSR-Webcam pair. GFK [24] gains the best performance on this pair of source-target domains. Our method always outperforms SDDL [39] in all pairs of source-target domains, especially for pairs such as Caltech-Amazon, Amazon-Caltech, Webcam-Caltech, and Webcam-Amazon, we achieves at least 10% improvements over SDDL. The main reason is that our proposed OCPD-SRC

TABLE II
SINGLE SOURCE RECOGNITION RATES ON ALL 31 CLASSES

Methods	A → W	D → W	W → D
Metric [23]	44	31	27
RDALR [50]	50.7±0.8	36.9±19.9	32.9±1.2
SGF [22]	57±3.5	36±1.1	37±2.3
GFK [24]	46.4±0.5	61.3±0.4	**66.3±0.4**
DASRC [36]	53.3±1.9	47.6±2.4	47.1±2.8
SDDL [39]	50.1±2.5	51.2±2.1	50.6±2.6
OCPD-SRC	**56.7±2.2**	**61.8±1.9**	63±2.5
DASH-N [63]	60.6±3.5	67.9±1.1	71.1±1.7
OCPD-SRC(Hierarchical)	**63.1±3.3**	**71.2±1.7**	**74.6±2.0**

TABLE III
RECOGNITION RATES (%) ON MULTIPLE SOURCES DOMAIN ADAPTATION

Methods	{D,A}→ W	{A,W}→ W	{W,D}→W
SGF [22]	52±2.5	39±1.1	28±0.8
RDALR [50]	36.9±1.1	31.2±1.3	20.9±0.9
SRC [1]	40.4±0.6	36.2±1.1	21.3±0.9
FDDL [15]	41.0±2.4	38.4±3.4	19.0±1.2
DASRC [36]	56.5±1.3	56.1±1.6	27.9±2.2
SDDL [39]	57.8±2.4	56.7±2.3	24.1±1.6
OCPD-SRC	**61.3±2.1**	**59.5±2.2**	**38.7±1.9**
DASH-N [63]	64.5±2.3	68.6±3.7	41.8±1.1
OCPD-SRC (hierarchical)	**65.9±2.3**	**69.0±1.9**	**46.3±2.2**

is designed based the decision rule of SRC, thus, the learned projections of data and dictionary by using our method can capture more discriminative information in the reduced subspace which is meaningful to classification. In addition, the low dimensional representation can well fit SRC, while SDDL cannot. DASRC achieves better performance on pairs of Caltech-Amazon, Caltech-DLSR, and Amazon-Caltech. This is because in these pairs the training images are sufficient, using these training samples as the dictionary contains enough discriminant information and has better representation ability. We also observe that when the test data has a different distribution than the training data, SRC and FDDL achieve poor performance.

From Table I, We can also observe that DASH-N provide better results than OCPD-SRC. The main reason is that DASH-N employ a multi-layer network to jointly learn the feature representation of data and domain shifts in each layer of the hierarchy and a better representation of data from different domains can be obtained. Since high-level features are sometimes more useful than low-level ones. However, our algorithm only contain a single layer and may not capture adequately the shift between the source and target domains. In addition, during the learning process, OCPD-SRC employs the hand-crafted features which requires a deep understanding of domain knowledge. In many applications, this requirements may be impractical. By employing the hierarchical network, OCPD-SRC (hierarchical) achieves reasonably good performance over DASH-N and obtains the best results for all pairs. Since DASH-N can be viewed as a generalization of the SDDL, thus in each layer, the learned transformations and dictionary by using our method have stronger discrimination power than DASH-N. This is the reason why our method can obtained better performance.

We also compare the recognition results for all 31 classes, as shown in Table II. OCPD-SRC outperforms all compared method in 2 out of 3 pairs of source-target domains except the results obtained by using multi-layer networks. For pairs such as Webcam-Amazon, OCPD-SRC achieve more than 10% improvements over SDDL. As was expected, OCPD-SRC (hierarchical) also performs better than DASH-N and achieves the best results for all domain pairs. This proves that hierarchical structure is helpful for transferring knowledge from source domain to target domain.

3) Multi-Source Domain Adaptation Experiment: For multi-source domain adaptation experiment, only the Office dataset is used and various algorithms are tested on all 31 classes. We set the number of dictionary atoms as 186 (i.e., 6 atoms per category) and the feature dimension is set as 90. For SDDL and FDDL, we follow the experiment setting in [38] and [39]. Table III lists the results of our proposed method and other multi-source domain adaptation methods. It can be seen from Table III that OCPD-SRC achieves the best results except the results obtained by using multi-layer networks for all the settings. This proves the effectiveness of our method. Especially, in the case of adapting from Webcam and DLSR to Amazon, our method significantly outperforms DASRC (from 27.9% to 38.7%). Compare to SDDL, it achieves an improvement of 14%. Similarly, SRC and FDDL are inefficient, when the testing data has a different distribution than the data used for training. When using the multi-layer networks, our methods achieve the best performances and is consistently better than DASH-N.

4) Parameter Analysis: We carry out several experiments to examine the performances of our method by using different parameters. We first evaluate the performance of OCPD-SRC versus different number of source images. Following [39], we choose Amazon/Webcam domain pair. Fig. 7 (a) shows the recognition rates of our method, SDDL, DASRC, FDDL and SRC over different number of source images. We can see that our method is consistently better than those of other methods, irrespective of the variations of source images. With increase of the number of source images, the performance of our method and SDDL slightly increase, while FDDL decreases with more source images. This indicates that we can improve the recognition performance of our method by increasing the number of source images. In addition, DASRC and SRC also increase when increase the number of source images.

We also evaluate our method OCPD-SRC using different dictionary size. We vary the dictionary size under six different source-target pairs. As shown in Fig. 7 (b), our approach can maintain high recognition accuracy when the dictionary size reaches 3 in most of pairs of source-target domains. With the increase of dictionary size, the performance of OCPD-SRC varies in a small range. In our experiments, we set 5 dictionary atoms per class.

Finally, we investigate the effect of the dimension of the learned feature projections of our method. The recognition rates of OCPD-SRC with respect to different dimensions are shown in Fig. 7 (c). We test our method on eight different source-target pairs. From Fig. 7 (c) we can see that with the increase of the dimensions, the performance of OCPD-SRC

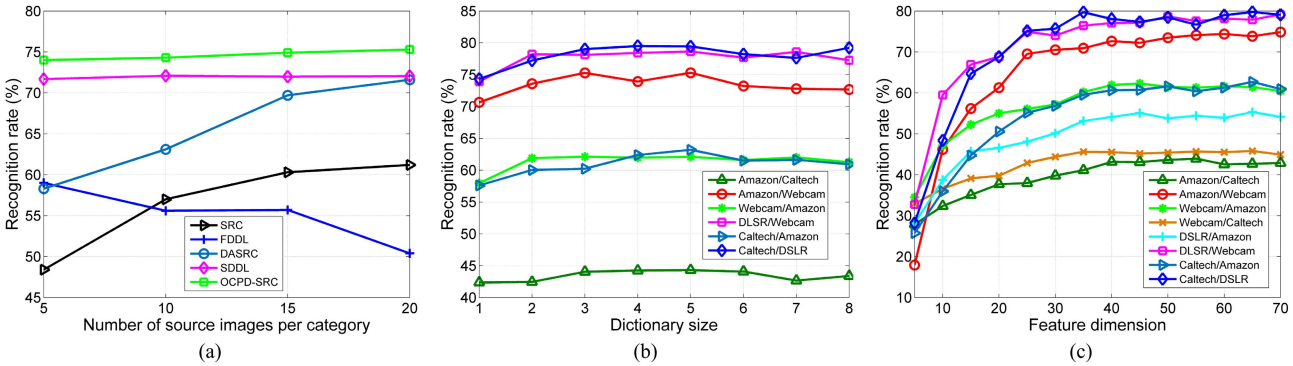


Fig. 7. Recognition rate of our method under different parameters: (a) number of source images, (b) dictionary size, and (c) feature dimension.

TABLE IV
RECOGNITION RATES OF DIFFERENT APPROACHES ON THE HALF-TONING DATASET. 10 COMMON CLASSES ARE USED

Methods	C → A	C → D	A → C	A → W	W → C	W → A	D → A	D → W
KNN	50.1±5.1	41.9±5.5	29.8±4.3	42.9±2.8	28.9±2.5	48.3±2.1	48.6±1.1	41.4±4.1
Metric [23]	41.1±6.4	38.8±5.6	31.9±5.4	49.4±3.4	32.8±3.2	49.9±3.3	43.8±2.6	49.3±2.6
DASRC [36]	59.2±4.7	68.2±5.1	36.6±2.3	67.5±3.9	33.7±2.2	48.7±2.8	52±3.7	68.5±4.1
SDDL [39]	52.2±3.9	66.7±5.5	34.1±3.5	69.2±4.2	34.6±2.8	51.2±3.4	54.1±2.7	71.6±5.3
OCPD-SRC	**64.3±3.3**	**73.4±3.8**	**45.8±2.7**	**77.7±3.0**	**43.0±2.6**	**60.9±3.5**	**59.7±2.7**	**74.7±4.4**
HMP [51]	65.0±5.5	68.7±3.7	44.7±3.1	67.9±3.2	40.3±2.3	59.6±3.9	62.0±4.1	74.7±2.9
DASH-N [63]	70.2±2.7	79.6±4.3	52.4±2.3	86.2±4.1	43.3±3.9	66.1±3.7	67.2±3.5	80.7±2.1
OCPD-SRC (Hierarchical)	**74.0±3.1**	**82.5±4.1**	**56.3±2.3**	**88.1±3.6**	**47.7±3.7**	**70.4±3.4**	**70.8±2.9**	**82.9±2.5**

TABLE V
RECOGNITION RATES OF DIFFERENT APPROACHES ON THE HALF-TONING DATASET. 10 COMMON CLASSES ARE USED

Methods	C → A	C → D	A → C	A → W	W → C	W → A	D → A	D → W
KNN	50.8±1.8	50.4±1.4	32.8±2.9	47.5±4.2	30.4±3.3	51.9±3.1	48.9±1.8	50.2±2.1
Metric [23]	42.8±2.7	43.8±2.5	35.2±2.1	53.6±1.4	36.8±1.8	53.2±3.1	40.8±3.9	54.5±2.7
DASRC [36]	58.3±5.4	61.9±5.0	34.1±3.6	61.1±4.7	32.8±2.9	52.4±3.3	51.8±2.9	60.6±3.4
SDDL [39]	52.9±5.2	63.8±6.3	32.4±3.2	62.5±5.7	33.5±2.9	55.2±2.8	55.4±3.3	65.3±4.7
OCPD-SRC	**65.2±4.7**	**70.7±5.2**	**40.9±2.9**	**66.6±4.1**	**42.2±3.4**	**61.4±3.0**	**59.1±3.5**	**66.4±3.8**
HMP [51]	65.1±2.5	61.4±4.9	43.7±4.7	64.2±2.6	37.7±4.9	62.3±6.8	59.4±4.7	70.9±2.7
DASH-N [63]	74.2±3.9	75.7±2.6	44.3±2.5	74.2±4.5	46.7±3.1	68.9±2.2	67.7±3.4	74.5±3.2
OCPD-SRC (Hierarchical)	**78.8±4.1**	**79.1±2.7**	**48.6±2.6**	**75.3±3.9**	**50.4±3.6**	**71.8±2.8**	**71.0±3.1**	**76.3±3.3**

also increase, and when the dimension over 40, our method tends to be stable.

D. Halftone and Edge Images

In order to test the effectiveness of OCPD-SRC in adapting to different domains, we conduct experiments on two new datasets generated by applying half-toning and edge detection from the office dataset. Fig. 8 shows sample of images from the keyboard class from these datasets. Half-toning images, which imitate the effect of jet-printing technology in the past, are generated using the dithering algorithm in [48]. Edge images are obtained by applying the Canny edge detector [49] with the threshold set to 0.07. We first extract 800-bin SURF features for both the domains, following the same approach as for the original dataset. And then use a two-layer networks to learn the feature representation as provided in [63].

Table 4 and Table 5 show the performances of different algorithms when adapting to half-tone and edge images datasets. From Table 4 and Table 5, we can see that employing the manually designed features, OCPD-SRC outperforms all compared method in 8 pairs of source-target domains. For pairs such as Caltech-Amazon, Amazon-Webcam, and Webcam-Amazon,

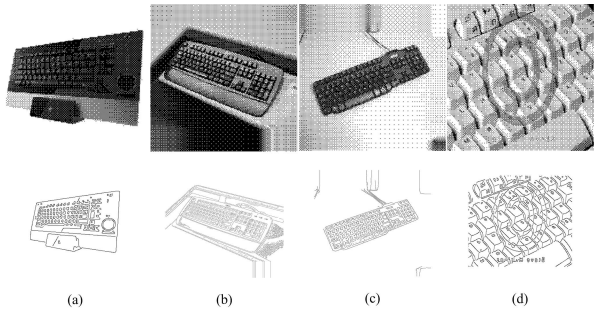


Fig. 8. Example images from the keyboard class in different domains. First row: Halftone images, second row: edge images. (a) Amazon. (b) Caltech. (c) DSLR. (d) Webcam.

there is more than 8% improvement over SDDL [40]. This proves the ability of our method for adapting well to new domains.

In our experiments, we also compared with Hierarchical Matching Pursuit (HMP) cite55, without performing domain adaptation. The HMP method builds a feature hierarchy layer by layer using an efficient matching pursuit encoder. As a result, it is robust to some of the variations present in the

images such as illumination changes, pose variations, and resolution variations. It can been seen from these tables that HMP achieves good performance on half-tone and edge images datasets. This demonstrates the effectiveness of learning feature representation. OCPD-SRC (Hierarchical)achieves better performance than DASH-N and HMP in all scenarios.

VII. Conclusion

We presented a novel domain adaptation dictionary learning method (OCPD-SRC). By jointly learning the projections of data from both source and target domains and a common structured dictionary, the proposed method obtained a better representation of data from different domain in the reduced space, and extracted more discriminant information for object classification. OCPD-SRC is designed according to the decision of SRC; it maximizes the between-class sparse reconstruction residuals of data with different domains and minimizes the within-class sparse reconstruction residuals of data in the projected low dimension subspace. Hence, the learned projections of data and dictionary can fit SRC well and improve the recognition performance of SRC for domain adaptation. In addition, our method can be easily extended to multiple domains and can be easily kernelized so that it can deal with the non-linear structure of data.

When no labels are available for the target domain the proposed method is inefficient. Hence, how to extend the proposed method for unsupervised adaptation is our further work. Furthermore, our methods only consider in a single layer, which ignores the possibility of transferring at multiple levels of the feature hierarchy. How to design an appropriate network to learn the feature representation is another further direction.

Appendix A
Proof of Proposition 1

Proposition 1: There exists an optimal solution $\mathbf{P_1^*}$, $\mathbf{P_2^*}$, $\cdots$, $\mathbf{P_M^*}$, $\mathbf{D}^*$ to Eq.(17), which has he following form:

$$\mathbf{P_i^*} = (\mathbf{Y_i A_i})^T, \forall \mathbf{i} = \mathbf{1}, \mathbf{2}, \cdots, \mathbf{M}, \tag{41}$$
$$\mathbf{D}^* = \widetilde{\mathbf{P}}^* \widetilde{\mathbf{Y}} \widetilde{\mathbf{B}}, \tag{42}$$

where $\widetilde{\mathbf{P}}^* = [\mathbf{P_1^*}, \mathbf{P_2^*}, \cdots, \mathbf{P_M^*}]$, for some $\mathbf{A_i} \in R^{N_i \times m_f}$ and some $\widetilde{\mathbf{B}} \in R^{\sum N_i \times K}$.

Proof (Form for D^*): First we will show the form for $\mathbf{D}^*$. We can decompose $\mathbf{D}^*$ into orthogonal components as follows

$$\mathbf{D}^* = \mathbf{D}_\parallel + \mathbf{D}_\perp \tag{43}$$

where $\mathbf{D}_\parallel = (\widetilde{\mathbf{P}}\widetilde{\mathbf{Y}})\widetilde{\mathbf{B}}$, $\mathbf{D}_\perp^T(\widetilde{\mathbf{P}}\widetilde{\mathbf{Y}}) = 0$,
for some $\widetilde{\mathbf{B}} \in R^{\sum_{i=1}^{M} N_i \times K}$. Substituting the value of $\mathbf{D}^*$ into the value of Eq.(17), we get for the two terms of $J(\widetilde{\mathbf{P}}, \mathbf{D})$.

Numerator Term:

$$\begin{aligned}
&= tr\left((\widetilde{\mathbf{P}}\widetilde{\mathbf{Y}} - \mathbf{D}\widetilde{\mathbf{\Lambda}}_b)^T(\widetilde{\mathbf{P}}\widetilde{\mathbf{Y}} - \mathbf{D}\widetilde{\mathbf{\Lambda}}_b)\right) \\
&= tr(\widetilde{\mathbf{Y}}^T\widetilde{\mathbf{P}}^T\widetilde{\mathbf{P}}\widetilde{\mathbf{Y}} + \widetilde{\mathbf{Y}}^T\widetilde{\mathbf{P}}^T\mathbf{D}_\parallel\widetilde{\mathbf{\Lambda}}_b + \widetilde{\mathbf{\Lambda}}_b^T\mathbf{D}_\parallel^T\mathbf{D}_\parallel\widetilde{\mathbf{\Lambda}}_b \\
&\quad + \widetilde{\mathbf{\Lambda}}_b^T\mathbf{D}_\perp^T\mathbf{D}_\perp\widetilde{\mathbf{\Lambda}}^T) \\
&\geq tr(\widetilde{\mathbf{Y}}^T\widetilde{\mathbf{P}}^T\widetilde{\mathbf{P}}\widetilde{\mathbf{Y}} + \widetilde{\mathbf{Y}}^T\widetilde{\mathbf{P}}^T\mathbf{D}_\parallel\widetilde{\mathbf{\Lambda}}_b + \widetilde{\mathbf{\Lambda}}_b^T\mathbf{D}_\parallel^T\mathbf{D}_\parallel\widetilde{\mathbf{\Lambda}}_b).
\end{aligned} \tag{44}$$

Denominator Term:

$$\begin{aligned}
&= tr\left((\widetilde{\mathbf{P}}\widetilde{\mathbf{Y}} - \mathbf{D}\widetilde{\mathbf{\Lambda}}_w)^T(\widetilde{\mathbf{P}}\widetilde{\mathbf{Y}} - \mathbf{D}\widetilde{\mathbf{\Lambda}}_w)\right) \\
&= tr(\widetilde{\mathbf{Y}}^T\widetilde{\mathbf{P}}^T\widetilde{\mathbf{P}}\widetilde{\mathbf{Y}} + \widetilde{\mathbf{Y}}^T\widetilde{\mathbf{P}}^T\mathbf{D}_\parallel\widetilde{\mathbf{\Lambda}}_w + \widetilde{\mathbf{\Lambda}}_w^T\mathbf{D}_\parallel^T\mathbf{D}_\parallel\widetilde{\mathbf{\Lambda}}_w \\
&\quad + \widetilde{\mathbf{\Lambda}}_w^T\mathbf{D}_\perp^T\mathbf{D}_\perp\widetilde{\mathbf{\Lambda}}^T) \\
&\geq tr(\widetilde{\mathbf{Y}}^T\widetilde{\mathbf{P}}^T\widetilde{\mathbf{P}}\widetilde{\mathbf{Y}} + \widetilde{\mathbf{Y}}^T\widetilde{\mathbf{P}}^T\mathbf{D}_\parallel\widetilde{\mathbf{\Lambda}}_w + \widetilde{\mathbf{\Lambda}}_w^T\mathbf{D}_\parallel^T\mathbf{D}_\parallel\widetilde{\mathbf{\Lambda}}_w).
\end{aligned} \tag{45}$$

The equality is reached when $\mathbf{D}_\perp = 0$. Hence, the form of $\mathbf{D}^*$ is:
$\mathbf{D}^* = \widetilde{\mathbf{P}}\widetilde{\mathbf{Y}}\widetilde{\mathbf{B}}$.

Form for P_i^*: For each $\mathbf{i} = \mathbf{1}, \cdots, \mathbf{M}$, $\mathbf{P_i^*}$ can be decomposed as:

$$\mathbf{P_i^*} = \mathbf{P}_{\parallel,\mathbf{i}} + \mathbf{P}_{\perp,\mathbf{i}} \tag{46}$$

where $\mathbf{P}_{\parallel,\mathbf{i}} = (\mathbf{Y_i A_i})^T$, $\mathbf{P}_{\perp,\mathbf{i}}\mathbf{Y_i} = 0$.

Let $\widetilde{\mathbf{P}}_\parallel = [\widetilde{\mathbf{P}}_{\parallel,\mathbf{1}}, \widetilde{\mathbf{P}}_{\parallel,\mathbf{2}}, \cdots, \widetilde{\mathbf{P}}_{\parallel,\mathbf{M}}]$ and $\widetilde{\mathbf{P}}_\perp = [\widetilde{\mathbf{P}}_{\perp,\mathbf{1}}, \widetilde{\mathbf{P}}_{\perp,\mathbf{2}}, \cdots, \widetilde{\mathbf{P}}_{\perp,\mathbf{M}}]$. Substituting the value for $\mathbf{P}^*$ into Eq.(17), we can write the term of $J(\widetilde{\mathbf{P}}, \mathbf{D})$ as:

Numerator Term:

$$\begin{aligned}
&= tr\left((\widetilde{\mathbf{P}}^*\widetilde{\mathbf{Y}} - \mathbf{D}\widetilde{\mathbf{\Lambda}}_b)(\widetilde{\mathbf{P}}^*\widetilde{\mathbf{Y}} - \mathbf{D}\widetilde{\mathbf{\Lambda}}_b)^T\right) \\
&= \|\widetilde{\mathbf{P}}^*\widetilde{\mathbf{Y}}(\mathbf{I} - \widetilde{\mathbf{B}}\widetilde{\mathbf{\Lambda}}_b)\|_F^2 \\
&= \|(\widetilde{\mathbf{P}}_\parallel + \widetilde{\mathbf{P}}_\perp)\widetilde{\mathbf{Y}}(\mathbf{I} - \widetilde{\mathbf{B}}\widetilde{\mathbf{\Lambda}}_b)\|_F^2 \\
&= \|\widetilde{\mathbf{P}}_\parallel\widetilde{\mathbf{Y}}(\mathbf{I} - \widetilde{\mathbf{B}}\widetilde{\mathbf{\Lambda}}_b)\|_F^2 \\
&= tr\left(\widetilde{\mathbf{P}}_\parallel\widetilde{\mathbf{Y}}(\mathbf{I} - \widetilde{\mathbf{B}}\widetilde{\mathbf{\Lambda}}_b)(\mathbf{I} - \widetilde{\mathbf{B}}\widetilde{\mathbf{\Lambda}}_b)^T\widetilde{\mathbf{Y}}^T\widetilde{\mathbf{P}}_\parallel^T\right).
\end{aligned} \tag{47}$$

Denominator Term:

$$\begin{aligned}
&= tr\left((\widetilde{\mathbf{P}}^*\widetilde{\mathbf{Y}} - \mathbf{D}\widetilde{\mathbf{\Lambda}}_w)(\widetilde{\mathbf{P}}^*\widetilde{\mathbf{Y}} - \mathbf{D}\widetilde{\mathbf{\Lambda}}_w)^T\right) \\
&= \|\widetilde{\mathbf{P}}^*\widetilde{\mathbf{Y}}(\mathbf{I} - \widetilde{\mathbf{B}}\widetilde{\mathbf{\Lambda}}_w)\|_F^2 \\
&= \|(\widetilde{\mathbf{P}}_\parallel + \widetilde{\mathbf{P}}_\perp)\widetilde{\mathbf{Y}}(\mathbf{I} - \widetilde{\mathbf{B}}\widetilde{\mathbf{\Lambda}}_w)\|_F^2 \\
&= \|\widetilde{\mathbf{P}}_\parallel\widetilde{\mathbf{Y}}(\mathbf{I} - \widetilde{\mathbf{B}}\widetilde{\mathbf{\Lambda}}_w)\|_F^2 \\
&= tr\left(\widetilde{\mathbf{P}}_\parallel\widetilde{\mathbf{Y}}(\mathbf{I} - \widetilde{\mathbf{B}}\widetilde{\mathbf{\Lambda}}_b)(\mathbf{I} - \widetilde{\mathbf{B}}\widetilde{\mathbf{\Lambda}}_b)^T\widetilde{\mathbf{Y}}^T\widetilde{\mathbf{P}}_\parallel^T\right).
\end{aligned} \tag{48}$$

Combining the two terms together, the objective function becomes:

$$\max \frac{tr\left(\widetilde{\mathbf{P}}_\parallel\widetilde{\mathbf{Y}}(\mathbf{I} - \widetilde{\mathbf{B}}\widetilde{\mathbf{\Lambda}}_b)(\mathbf{I} - \widetilde{\mathbf{B}}\widetilde{\mathbf{\Lambda}}_b)^T\widetilde{\mathbf{Y}}^T\widetilde{\mathbf{P}}_\parallel^T\right)}{tr\left(\widetilde{\mathbf{P}}_\parallel\widetilde{\mathbf{Y}}(\mathbf{I} - \widetilde{\mathbf{B}}\widetilde{\mathbf{\Lambda}}_b)(\mathbf{I} - \widetilde{\mathbf{B}}\widetilde{\mathbf{\Lambda}}_b)^T\widetilde{\mathbf{Y}}^T\widetilde{\mathbf{P}}_\parallel^T\right)}. \tag{49}$$

It can be seen that from Eq.(49), that the object function is independent of $\mathbf{P}_{\perp,\mathbf{i}}$, hence it can be safely set to be $\mathbf{0}$. Hence,

$$\mathbf{P_i^*} = (\mathbf{Y_i A_i})^T$$

Appendix B
Proof of Lemma 1

Lemma 1: (1). $f(\rho)$ is a decreasing function.
(2). $f(\rho) = 0$ iff $\rho = \rho^*$.

Proof: For any ρ, denote $\widetilde{\mathbf{A}}^*$ that maximmize $f(\rho)$. To prove (1) we need to prove $G(\widetilde{\mathbf{A}}^*, \rho_2) < G(\widetilde{\mathbf{A}}^*, \rho_1)$ for $\rho_2 > \rho_1$.

We compare the sums of the m_f largest eigenvalues of $\widetilde{\mathbf{K}}(\widetilde{\mathbf{S}}_b - \rho_2\widetilde{\mathbf{S}}_w)\widetilde{\mathbf{K}}^T - \rho_2\mu\mathbf{I}$ and $\widetilde{\mathbf{K}}(\widetilde{\mathbf{S}}_b - \rho_1\widetilde{\mathbf{S}}_w)\widetilde{\mathbf{K}}^T - \rho_1\mu\mathbf{I}$ for $\rho_2 > \rho_1$. We have

$$G(\rho_2) - G(\rho_1) = (\rho_1 - \rho_2)\widetilde{\mathbf{K}}\widetilde{\mathbf{S}}_w\widetilde{\mathbf{K}}^T + (\rho_1 - \rho_2)\mu\mathbf{I} \prec 0 \tag{50}$$

and therefore $G(\rho_2) < G(\rho_1)$, property 1 is proved.

(2). To prove (2), we start by observing that the sufficient condition is trivial, i.e., according to the definition of ρ^*, $\rho = \rho^*$ implies $f(\rho) = 0$. Next, since $tr\left(\widetilde{\mathbf{A}}^T(\widetilde{\mathbf{K}}\widetilde{\mathbf{S}}_w\widetilde{\mathbf{K}}^T + \mu\mathbf{I})\widetilde{\mathbf{A}}\right) > 0$ for any $\widetilde{\mathbf{A}}$ satisfies $\mathbf{A_i}^T\mathbf{K_i}\mathbf{A_i} = \mathbf{I}$, we can write

$$tr\left(\widetilde{\mathbf{A}}^T(\widetilde{\mathbf{K}}(\widetilde{\mathbf{S}}_b - \rho\widetilde{\mathbf{S}}_w)\widetilde{\mathbf{K}}^T - \rho\mu\mathbf{I})\widetilde{\mathbf{A}}\right) \le 0 \quad iff$$

$$\frac{tr(\widetilde{\mathbf{A}}^T\widetilde{\mathbf{K}}\widetilde{\mathbf{S}}_b\widetilde{\mathbf{K}}^T\widetilde{\mathbf{A}})}{tr\left(\widetilde{\mathbf{A}}^T(\widetilde{\mathbf{K}}\widetilde{\mathbf{S}}_w\widetilde{\mathbf{K}}^T + \mu\mathbf{I})\widetilde{\mathbf{A}}\right)} \le \rho. \tag{51}$$

This can be restated as

$$f(\rho) \le 0 \; iff \; \rho^* \le \rho. \tag{52}$$

Suppose now that $f(\rho) > 0$ for a certain ρ. Then, there is a $\widetilde{\mathbf{A}}_0$ such that

$$tr\left(\widetilde{\mathbf{A}}_0^T(\widetilde{\mathbf{K}}(\widetilde{\mathbf{S}}_b - \rho\widetilde{\mathbf{S}}_w)\widetilde{\mathbf{K}}^T - \rho\mu\mathbf{I})\widetilde{\mathbf{A}}_0\right) > 0 \Rightarrow$$

$$\frac{tr(\widetilde{\mathbf{A}}_0^T\widetilde{\mathbf{K}}\widetilde{\mathbf{S}}_b\widetilde{\mathbf{K}}^T\widetilde{\mathbf{A}}_0)}{tr\left(\widetilde{\mathbf{A}}_0^T(\widetilde{\mathbf{K}}\widetilde{\mathbf{S}}_w\widetilde{\mathbf{K}}^T + \mu\mathbf{I})\widetilde{\mathbf{A}}_0\right)} > \rho. \tag{53}$$

This means that

$$\max_{\widetilde{\mathbf{A}}} \frac{tr(\widetilde{\mathbf{A}}^T\widetilde{\mathbf{K}}\widetilde{\mathbf{S}}_b\widetilde{\mathbf{K}}^T\widetilde{\mathbf{A}})}{tr\left(\widetilde{\mathbf{A}}^T(\widetilde{\mathbf{K}}\widetilde{\mathbf{S}}_w\widetilde{\mathbf{K}}^T + \mu\mathbf{I})\widetilde{\mathbf{A}}\right)} > \rho, \tag{54}$$

and therefore, $\rho^* > \rho$. It can be expressed as

$$f(\rho) > 0 \Rightarrow \rho^* > \rho. \tag{55}$$

Thus, $f(\rho) = 0$ implies $\rho^* = \rho$. This proves property 2.

APPENDIX C

The within-class sparse reconstruction residual of source domain in Eq.(10) can be rewritten as follows:

$$\begin{aligned} J_w^{\mathbf{1}} &= tr\left(\sum_{i=1}^{c}\sum_{j=1}^{n_i^{\mathbf{1}}}(\mathbf{P_1}\mathbf{y}_{i,j}^{\mathbf{1}} - \mathbf{D}\delta_i(\boldsymbol{\alpha}_{i,j}^{\mathbf{1}}))(\mathbf{P_1}\mathbf{y}_{i,j}^{\mathbf{1}} - \mathbf{D}\delta_i(\boldsymbol{\alpha}_{i,j}^{\mathbf{1}}))^T\right) \\ &= tr\left(\sum_{i=1}^{c}\sum_{j=1}^{n_i^{\mathbf{1}}}(\mathbf{P_1}\mathbf{y}_{i,j}^{\mathbf{1}} - \mathbf{D}_i\boldsymbol{\alpha}_{i,j}^{\mathbf{1},i})(\mathbf{P_1}\mathbf{y}_{i,j}^{\mathbf{1}} - \mathbf{D}_i\boldsymbol{\alpha}_{i,j}^{\mathbf{1},i})^T\right) \\ &= \sum_{i=1}^{c} tr(\mathbf{P_1}\mathbf{Y}_i^{\mathbf{1}} - \mathbf{D}_i\boldsymbol{\Lambda}_w^{\mathbf{1},i})(\mathbf{P_1}\mathbf{Y}_i^{\mathbf{1}} - \mathbf{D}_i\boldsymbol{\Lambda}_w^{\mathbf{1},i})^T \\ &= \sum_{i=1}^{c} \|\mathbf{P_1}\mathbf{Y}_i^{\mathbf{1}} - \mathbf{D}_i\boldsymbol{\Lambda}_w^{\mathbf{1},i}\|_F^2, \end{aligned} \tag{56}$$

where $\boldsymbol{\Lambda}_w^{\mathbf{1},i} = [\boldsymbol{\alpha}_{1,1}^{\mathbf{1},i}, \boldsymbol{\alpha}_{1,2}^{\mathbf{1},i}, \cdots, \boldsymbol{\alpha}_{c,n_c^{\mathbf{1}}}^{\mathbf{1},i}]$, $\boldsymbol{\alpha}_{i,j}^{\mathbf{1},i}$ is the representation coefficient vector associated with class i, $i = 1, \cdots, c$, $j = 1, \cdots, n_i$.

Then the between-class sparse reconstruction residual of source domain in Eq.(11)

$$\begin{aligned} J_b^{\mathbf{1}} &= tr\left(\sum_{i=1}^{c}\sum_{j=1}^{n_i^{\mathbf{1}}}\sum_{s\neq i}(\mathbf{P_1}\mathbf{y}_{i,j}^{\mathbf{1}} - \mathbf{D}\delta_s(\boldsymbol{\alpha}_{i,j}^{\mathbf{1}}))(\mathbf{P_1}\mathbf{y}_{i,j}^{\mathbf{1}} - \mathbf{D}\delta_s(\boldsymbol{\alpha}_{i,j}^{\mathbf{1}}))^T\right) \\ &= \sum_{i=1}^{c} tr\left(\sum_{j=1}^{n_i^{\mathbf{1}}}\sum_{s\neq i}(\mathbf{P_1}\mathbf{y}_{i,j}^{\mathbf{1}} - \mathbf{D}_s\boldsymbol{\alpha}_{i,j}^{\mathbf{1},s})(\mathbf{P_1}\mathbf{y}_{i,j}^{\mathbf{1}} - \mathbf{D}_s\boldsymbol{\alpha}_{i,j}^{\mathbf{1},s})^T\right) \\ &= \sum_{i=1}^{c} \|\mathbf{P_1}\mathbf{Y}_i^{\mathbf{1}} - \mathbf{D}_s\boldsymbol{\Lambda}_b^{\mathbf{1},s}\|_F^2, \end{aligned} \tag{57}$$

where $\boldsymbol{\Lambda}_b^{\mathbf{1},s} = [\boldsymbol{\alpha}_{1,1}^{\mathbf{1},s}, \boldsymbol{\alpha}_{1,2}^{\mathbf{1},s}, \cdots, \boldsymbol{\alpha}_{c,n_c^{\mathbf{1}}}^{\mathbf{1},s}]$, $\boldsymbol{\alpha}_{i,j}^{\mathbf{1},s}$ is the representation coefficient vector associated with class s, $s \neq i$.

Similarly, the within-class sparse reconstruction residual and the between-class sparse reconstruction residual of target domain in Eq.(12) and Eq.(13) can be rewritten as:

$$J_w^{\mathbf{2}} = \sum_{i=1}^{c} \|\mathbf{P_2}\mathbf{Y}_i^{\mathbf{2}} - \mathbf{D}_i\boldsymbol{\Lambda}_w^{\mathbf{2},i}\|_F^2, \tag{58}$$

and

$$J_b^{\mathbf{2}} = \sum_{i=1}^{c} \|\mathbf{P_2}\mathbf{Y}_i^{\mathbf{2}} - \mathbf{D}_s\boldsymbol{\Lambda}_b^{\mathbf{2},s}\|_F^2, \tag{59}$$

where $\boldsymbol{\Lambda}_w^{\mathbf{2},i} = [\boldsymbol{\alpha}_{1,1}^{\mathbf{2},i}, \boldsymbol{\alpha}_{1,2}^{\mathbf{2},i}, \cdots, \boldsymbol{\alpha}_{c,n_c^{\mathbf{2}}}^{\mathbf{2},i}]$, and $\boldsymbol{\Lambda}_b^{\mathbf{2},s} = [\boldsymbol{\alpha}_{1,1}^{\mathbf{2},s}, \boldsymbol{\alpha}_{1,2}^{\mathbf{2},s}, \cdots, \boldsymbol{\alpha}_{c,n_c^{\mathbf{2}}}^{\mathbf{2},s}]$. $\boldsymbol{\alpha}_{i,j}^{\mathbf{2},i}$, $\boldsymbol{\alpha}_{i,j}^{\mathbf{2},s}$ are the representation coefficients which associated with class i can class s, respectively. Finally, we maximize the following cost function

$$\begin{aligned} \max J_b &= \max\{J_b^{\mathbf{1}} + J_b^{\mathbf{2}}\} \\ &= \max \sum_{i=1}^{c}\sum_{s\neq i}\left(\|\widetilde{\mathbf{P}}\widetilde{\mathbf{Y}}_i - \mathbf{D}_s\widetilde{\boldsymbol{\Lambda}}_b^s\|_F^2\right), \end{aligned} \tag{60}$$

and simultaneous minimize the following cost function:

$$\begin{aligned} \max J_w &= \max\{J_w^{\mathbf{1}} + J_w^{\mathbf{2}}\} \\ &= \max \sum_{i=1}^{c}\left(\|\widetilde{\mathbf{P}}\widetilde{\mathbf{Y}}_i - \mathbf{D}_i\widetilde{\boldsymbol{\Lambda}}_w^i\|_F^2\right), \end{aligned} \tag{61}$$

where $\widetilde{\mathbf{P}} = [\mathbf{P_1}, \mathbf{P_2}]$, $\widetilde{\mathbf{Y}} = \begin{bmatrix} \mathbf{Y}_i^{\mathbf{1}} & \mathbf{0} \\ \mathbf{0} & \mathbf{Y}_i^{\mathbf{2}} \end{bmatrix}$, $\widetilde{\boldsymbol{\Lambda}}_b^s = [\boldsymbol{\Lambda}_b^{\mathbf{1},s}, \boldsymbol{\Lambda}_b^{\mathbf{2},s}]$ and $\widetilde{\boldsymbol{\Lambda}}_w^i = [\boldsymbol{\Lambda}_w^{\mathbf{1},s}, \boldsymbol{\Lambda}_w^{\mathbf{2},s}]$. Thus, Combine the proposition 1, the objective function can be rewritten as follows:

$$\begin{aligned} J &= \max_{\widetilde{\mathbf{A}},\mathbf{D}} \frac{\sum_{i=1}^{c}\sum_{s\neq i} tr\left(\widetilde{\mathbf{A}}^T(\widetilde{\mathbf{K}}_i - \mathbf{B}_s\widetilde{\boldsymbol{\Lambda}}_b^s)(\widetilde{\mathbf{K}}_i - \mathbf{B}_s\widetilde{\boldsymbol{\Lambda}}_b^s)^T\widetilde{\mathbf{A}}\right)}{\sum_{i=1}^{c} tr\left(\widetilde{\mathbf{A}}^T(\widetilde{\mathbf{K}}_i - \mathbf{B}_i\widetilde{\boldsymbol{\Lambda}}_w^i)(\widetilde{\mathbf{K}}_i - \mathbf{B}_i\widetilde{\boldsymbol{\Lambda}}_w^i)^T\widetilde{\mathbf{A}}\right)} \\ &= \max_{\mathbf{B}_i} \sum_{i=1}^{c} \frac{tr(\widetilde{\mathbf{A}}^T\widetilde{\mathbf{S}}_b^i\widetilde{\mathbf{A}})}{tr(\widetilde{\mathbf{A}}^T\widetilde{\mathbf{S}}_w\widetilde{\mathbf{A}})}, \end{aligned} \tag{62}$$

where $\widetilde{\mathbf{K}}_i = \widetilde{\mathbf{Y}}^T\widetilde{\mathbf{Y}}_i$.

REFERENCES

[1] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009.

[2] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Non-local sparse models for image restoration," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep./Oct. 2009, pp. 2272–2279.

[3] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Trans. Image Process.*, vol. 19, no. 11, pp. 2861–2873, Nov. 2010.

[4] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, Dec. 2006.

[5] S. Bahrampour, N. M. Nasrabadi, A. Ray, and W. K. Jenkins, "Multimodal task-driven dictionary learning for image classification," *IEEE Trans. Image Process.*, vol. 25, no. 1, pp. 24–38, Jan. 2006.

[6] G. Zhang, H. Sun, G. Xia, and Q. Sun, "Multiple kernel sparse representation-based orthogonal discriminative projection and its cost-sensitive extension," *IEEE Trans. Image Process.*, vol. 25, no. 9, pp. 4271–4285, Sep. 2016.

[7] K. Huang and S. Aviyente, "Sparse representation for signal classification," in *Proc. Adv. Neural Inf. Process. Syst.*, 2006, pp. 609–616.

[8] S. Gao, I. W.-H. Tsang, and Y. Ma, "Learning category-specific dictionary and shared dictionary for fine-grained image categorization," *IEEE Trans. Image Process.*, vol. 23, no. 2, pp. 623–634, Feb. 2014.

[9] G. Zhang, H. Sun, G. Xia, and Q. Sun, "Kernel collaborative representation based dictionary learning and discriminative projection," *Neurocomputing*, vol. 207, pp. 300–309, Sep. 2016.

[10] G. Zhang, H. Sun, Z. Ji, G. Xia, L. Feng, and Q. Sun, "Kernel dictionary learning based discriminant analysis," *J. Vis. Commun. Image Represent.*, vol. 40, pp. 470–484, Oct. 2016.

[11] G. Zhang, H. Sun, Z. Ji, Y.-H. Yuan, and Q. Sun, "Cost-sensitive dictionary learning for face recognition," *Pattern Recognit.*, vol. 60, pp. 613–629, Dec. 2016.

[12] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.

[13] Q. Zhang and B. Li, "Discriminative K-SVD for dictionary learning in face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2691–2698.

[14] Z. Jiang, Z. Lin, and L. S. Davis, "Label consistent K-SVD: Learning a discriminative dictionary for recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2651–2664, Nov. 2013.

[15] M. Yang, L. Zhang, X. Feng, and D. Zhang, "Fisher discrimination dictionary learning for sparse representation," in *Proc. IEEE 13th Int. Conf. Comput. Vis.*, Nov. 2011, pp. 543–550.

[16] M. Yang, D. Dai, L. Shen, and L. Van Gool, "Latent dictionary learning for sparse representation based classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 4138–4145.

[17] B. Kulis, K. Saenko, and T. Darrell, "What you saw is not what you get: Domain adaptation using asymmetric kernel transforms," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 1785–1792.

[18] H. Daumé, III, "Frustratingly easy domain adaptation," in *Proc. ACL*, 2007, pp. 256–263.

[19] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, "Analysis of representations for domain adaptation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 137–144.

[20] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[21] A. Bergamo and L. Torresani, "Exploiting weakly-labeled Web images to improve object classification: A domain adaptation approach," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 181–189.

[22] R. Gopalan, R. Li, and R. Chellappa, "Domain adaptation for object recognition: An unsupervised approach," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 999–1006.

[23] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 213–226.

[24] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2066–2073.

[25] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars, "Unsupervised visual domain adaptation using subspace alignment," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2960–2967.

[26] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars. (Sep. 18, 2014). "Subspace alignment for domain adaptation." [Online]. Available: https://arxiv.org/abs/1409.5241

[27] J. Ni, Q. Qiu, and R. Chellappa, "Subspace interpolation via dictionary learning for unsupervised domain adaptation," in *Proc. IEEE Int. Conf. Comput. Vis.*, Jun. 2013, pp. 692–699.

[28] W. Li, L. Duan, D. Xu, and I. W. Tsang, "Learning with augmented features for supervised and semi-supervised heterogeneous domain adaptation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 6, pp. 1134–1148, Jun. 2014.

[29] Y. Shi and F. Sha, "Information-theoretical learning of discriminative clusters for unsupervised domain adaptation," in *Proc. 29th ICML*, 2012, pp. 1079–1086.

[30] A. Shrivastava, S. Shekhar, and V. M. Patel, "Unsupervised domain adaptation using parallel transport on Grassmann manifold," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Mar. 2014, pp. 277–284.

[31] J. Yang, R. Yan, and A. G. Hauptmann, "Cross-domain video concept detection using adaptive SVMs," in *Proc. 15th Int. Conf. Multimedia*, 2007, pp. 188–197.

[32] L. Duan, I. W. Tsang, D. Xu, and S. J. Maybank, "Domain transfer SVM for video concept detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 1375–1381.

[33] J. Hoffman, E. Rodner, J. Donahue, T. Darrell, and K. Saenko, "Efficient learning of domain-invariant image representations," in *Proc. IEEE Conf. Comput. Learn. Represent.*, 2013, pp. 1–9.

[34] L. Duan, I. W. Tsang, and D. Xu, "Domain transfer multiple kernel learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 3, pp. 465–479, Mar. 2012.

[35] V. M. Patel, R. Gopalan, R. Li, and R. Chellappa, "Visual domain adaptation: A survey of recent advances," *IEEE Signal Process. Mag.*, vol. 32, no. 3, pp. 53–69, May 2015.

[36] H. Zhang, V. M. Patel, S. Shekhar, and R. Chellappa, "Domain adaptive sparse representation-based classification," in *Proc. Autom. Face Gesture Recognit. Workshops*, May 2015, pp. 1–8.

[37] Q. Qiu, V. M. Patel, P. Turaga, and R. Chellappa, "Domain adaptive dictionary learning," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 631–645.

[38] S. Shekhar, V. M. Patel, H. V. Nguyen, and R. Chellappa, "Generalized domain-adaptive dictionaries," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 361–368.

[39] S. Shekhar, V. M. Patel, H. Van Nguyen, and R. Chellappa, "Coupled projections for adaptation of dictionaries," *IEEE Trans. Image Process.*, vol. 24, no. 10, pp. 2941–2954, Oct. 2015.

[40] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. S. Huang, and S. Yan, "Sparse representation for computer vision and pattern recognition," *Proc. IEEE*, vol. 98, no. 6, pp. 1031–1044, Jun. 2010.

[41] M. Yang, L. Zhang, J. Yang, and D. Zhang, "Metaface learning for sparse representation based face recognition," in *Proc. IEEE Conf. Image Process.*, Sep. 2010, pp. 1601–1604.

[42] P. Sprechmann and G. Sapiro, "Dictionary learning and sparse coding for unsupervised clustering," in *Proc. ICASSP*, 2010, pp. 2042–2045.

[43] T. T. Ngo, M. Bellalij, and Y. Saad, "The trace ratio optimization problem," *SIAM Rev.*, vol. 54, no. 3, pp. 545–569, 2012.

[44] W. Jiang and F.-L. Chung, "A trace ratio maximization approach to multiple kernel-based dimensionality reduction," *Neural Netw.*, vol. 49, pp. 96–106, Jan. 2014.

[45] Z. Wen and W. Yin, "A feasible method for optimization with orthogonality constraints," *Math. Program.*, vol. 142, no. 1, pp. 397–434, 2013.

[46] K.-R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf, "An introduction to kernel-based learning algorithms," *IEEE Trans. Neural Netw.*, vol. 12, no. 2, pp. 181–201, Mar. 2001.

[47] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," California Inst. Technol., Pasadena, CA, USA, Tech. Rep. 7694, 2007.

[48] V. Monga, N. Damera-Venkata, H. Rehman, and B. L. Evans. (2005). *Halftoning Toolbox for MATLAB*. [Online]. Available: http://users.Ece.utexas.edu/~bevans/projects/halftoning/toolbox/

[49] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.

[50] I.-H. Jhuo, D. Liu, D. T. Lee, and S.-F. Chang, "Robust visual domain adaptation with low-rank reconstruction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2168–2175.

[51] L. Bo, X. Ren, and D. Fox, "Hierarchical matching pursuit for image classification: Architecture and fast algorithms," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 2115–2123.

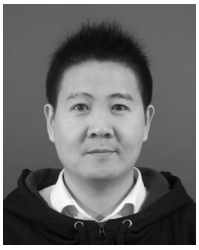
[52] H. V. Nguyen, V. M. Patel, N. M. Nasrabadi, and R. Chellappa, "Sparse embedding: A framework for sparsity promoting dimensionality reduction," in *Proc. 12th Eur. Conf. Comput. Vis.*, Oct. 2012, pp. 414–427.

[53] X. Glorot, A. Bordes, and Y. Bengio, "Domain adaptation for large-scale sentiment classification: A deep learning approach," in *Proc. 28th Int. Conf. Mach. Learn.*, 2011, pp. 513–520.

[54] M. Chen, Z. Xu, K. Weinberger, and F. Sha, "Marginalized denoising autoencoders for domain adaptation," in *Proc. 29th Int. Conf. Mach. Learn.*, 2012, pp. 1–8.

[55] J. Hoffman, E. Tzeng, J. Donahue, Y. Jia, K. Saenko, and T. Darrell. (2013). "One-shot adaptation of supervised deep convolutional models." [Online]. Available: http://arxiv.org/abs/1312.6204

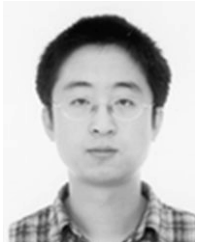
[56] J. Donahue *et al.*, "DeCAF: A deep convolutional activation feature for generic visual recognition," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 647–655.
[57] S. Chopra, S. Balakrishnan, and R. Gopalan, "DLID: Deep learning for domain adaptation by interpolating between domains," in *Proc. ICML Workshop Challenges Represent. Learn.*, 2013, pp. 1–8.
[58] B. Sun and K. Saenko. (2016). "Deep CORAL: Correlation alignment for deep domain adaptation." [Online]. Available: https://arxiv.org/abs/1607.01719
[59] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell. (2014). "Deep domain confusion: Maximizing for domain invariance." [Online]. Available: https://arxiv.org/abs/1412.3474
[60] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko, "Simultaneous deep transfer across domains and tasks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 4068–4076.
[61] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker, "Multi-PIE," *Image Vis. Comput.*, vol. 28, no. 5, pp. 807–813, 2010.
[62] A. Sharma, A. Kumar, H. Daume, and D. W. Jacobs, "Generalized multiview analysis: A discriminative latent space," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2160–2167.
[63] H. V. Nguyen, H. T. Ho, V. M. Patel, and R. Chellappa, "DASH-N: Joint hierarchical domain adaptation and feature learning," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5479–5491, Dec. 2015.



Guoqing Zhang received the B.S. and master's degrees in information engineering from Yangzhou University, Yangzhou, China, in 2009 and 2012, respectively, and the Ph.D. degree in pattern recognition and intelligence system from the Nanjing University of Science and Technology, Nanjing, China, in 2017. His current research interests include computer vision, pattern recognition, and machine learning.

Huaijiang Sun received the B.Eng. and Ph.D. degrees from the School of Marine Engineering, Northwestern Polytechnical University, Xi'an, China, in 1990 and 1995, respectively. He is currently a Professor with the Department of Computer Science and Engineering, Nanjing University of Science and Technology. His research interests include computer vision and pattern recognition, image and video processing, and intelligent information processing.

Fatih Porikli (M'99–F'13) received the Ph.D. degree from New York University in 2002. He was a Distinguished Research Scientist with the Mitsubishi Electric Research Laboratories. He is currently a Professor with the Research School of Engineering, Australian National University. He is also acting as the Chief Scientist with Huawei, Santa Clara. He has authored over 200 publications, invented 71 U.S. patents, and co-edited two books. His research interests include computer vision, pattern recognition, manifold learning, image enhancement, robust and sparse optimization, and online learning with commercial applications in autonomous vehicles, video surveillance, visual inspection, robotics, consumer electronics, satellite imaging, and medical systems. He was a recipient of the R&D 100 Scientist of the Year Award in 2006. He received five best paper awards at premier IEEE conferences. He is serving as the associate editor of five journals for the past ten years.



Yazhou Liu received the B.S. degree in mechanical engineering from Harbin Engineering University, Harbin, China, in 2002, and the M.E. and Ph.D. degrees in computer science from the Harbin Institute of Technology, Harbin, China, in 2004 and 2009, respectively. From 2009 to 2011, he was a Post-Doctoral Research Fellow with the Machine Vision Group, Oulu University, Finland. Since 2011, he has been a Faculty Member with the Department of Computer Science and Engineering, Nanjing University of Science and Technology.

Quansen Sun received the Ph.D. degree in pattern recognition and intelligence system from the Nanjing University of Science and Technology, Nanjing, China, in 2006. He is currently a Professor with the School of Computer Science and Engineering, Nanjing University of Science and Technology. His research interests include pattern recognition, image processing, computer vision, and data fusion.