Domain Adaptive Sparse Representation-Based Classification

Heng Zhang[1], Vishal M. Patel[1], Sumit Shekhar[2] and Rama Chellappa[1]
[1] Center for Automation Research, UMIACS, University of Maryland, College Park, MD 20742 USA
[2] Adobe Research, Bangalore, India
{hzhang98, pvishalm, sshekha, rama}@umiacs.umd.edu

Abstract— **In recent years, sparse representation and dictionary learning methods have produced state-of-the-art results in many biometric recognition problems such as face, gait and iris recognition. However, when sparse representation-based classification methods are confronted with situations where the training data has different distribution than the test data, their performance degrades significantly. In this paper, we propose a general sparse representation-based classification method that learns projections of data in a space where the sparsity of data is maintained. We propose an efficient iterative procedure for solving the proposed optimization problem. One of the key features of the proposed method is that it is computationally efficient as the learning is done in the lower-dimensional space. Various experiments on mobile active authentication datasets consisting of face and screen touch gestures show that our method is able to capture the meaningful structure of data and can perform significantly better than many competitive domain adaptation algorithms.**

I. INTRODUCTION

In biometrics recognition, one is often faced with scenarios where the training data used to learn a recognition engine has a different distribution from the test data. Examples of such cases include: recognizing and detecting faces under poor lighting conditions and poses while the algorithms are trained on well-illuminated frontal faces, recognizing low-resolution face images when recognition algorithms are instead optimized for high-resolution images, recognizing and detecting human faces on infrared images while algorithms are optimized for color images, etc. (see Figure 1). Regardless of the specific cause, any distributional change that occurs after learning a classifier can degrade its performance at test time. Domain adaptation essentially tries to mitigate this dataset shift problem [1].

Various domain adaptation methods have been proposed in the computer vision and machine learning literature. One of the simplest domain adaptation approaches is the feature augmentation work proposed in [2]. The goal is to make a domain specific copy of the original features for each domain. This work was extended for the heterogeneous data in [3]. The idea of feature augmentation has also been extended to consider a manifold of intermediate domains [4]. Rather than working with the information conveyed by the source and target domains alone, [4] proposed an incremental learning technique based on gradually following the geodesic path between the source and target domains. Geodesic flows were used to derive intermediate subspaces that interpolate between the source and target domains. Recently, the approach of [4] was kernelized and extended to the infinite case, defining a new kernel equivalent to integrating over all common subspaces that lie on the geodesic flow connecting the source and target subspaces, respectively [5].

This work was supported by cooperative agreement FA8750-13-2-0279 from DARPA.

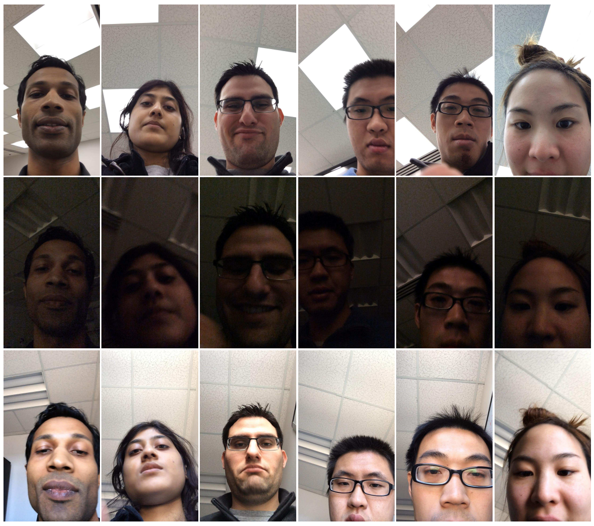


Fig. 1: Example of dataset shift in face recognition. Each row shows face images collected from a mobile device in a particular ambient condition. Images in each column correspond to the same individual. It can be seen from this figure that the images from different ambient conditions show very different characteristics. If a classifier is trained on images shown on the first row and tested on the images in the second or third row, then its performance degrades significantly (See Section IV).

Various feature transformation-based approaches have also been proposed in the literature [6], [7], [8]. The idea behind this method is to adapt features across general image domains by learning transformations. Another class of domain adaptation algorithms is based on parameter adaptation in which the Support Vector Machine (SVM) type of algorithms are proposed for domain adaptation. Algorithms such as adaptive SVM [9], domain transfer SVM [10], max-margin domain transfer [11] and domain adaptive multiple kernel learning [12] fall under this category.

Dictionary learning-based methods have also gained a lot of attention in recent years for domain adaptation. In [13],

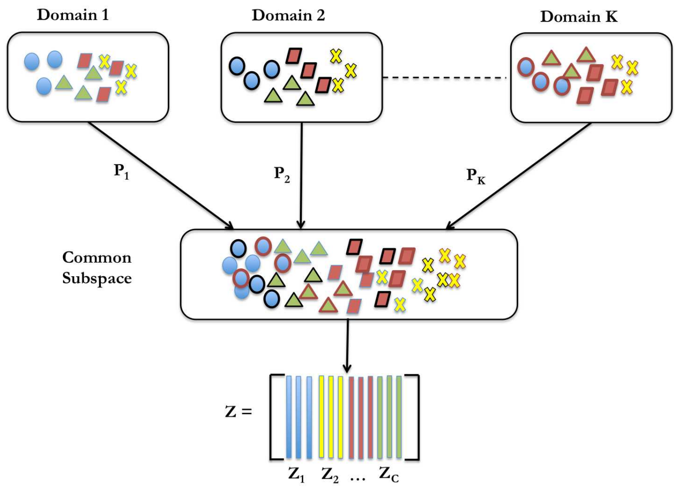


Fig. 2: An overview of the proposed domain adaptive sparse representation-based classification method.

the idea of sparse domain transfer under the framework of dictionary learning was proposed for image super-resolution and photo-sketch synthesis. A technique for jointly learning transformations of data in the source and target domains, and a latent discriminative dictionary that can succinctly represent both domains in the projected low-dimensional space was proposed in [14]. In [15], a function learning framework was presented for the task of transforming a dictionary learned from one visual domain to the other, while maintaining a domain-invariant representation of a signal. Another approach [16] proposed using concepts from dictionary learning to generate intermediate domains that bridge the domain shift. See [17], [18] and [1] for more detailed discussion of recent domain adaptation approaches.

In this paper, we propose a different approach to the problem of domain adaptation based on sparse representation. Our method learns projections of data in different domains in a way that preserves the sparse structure of data in the low-dimensional space. Note that similar idea has been explored under the framework of dictionary learning in [14]. We propose an efficient optimization method based on Alternating Direction Method of Multipliers (ADMM) and the Method of Splitting Orthogonality Constraints (SOC) for solving the proposed problem. One of the advantages of the proposed method compared to other dictionary-based domain adaptation methods is that it is very efficient as it does not require learning a dictionary. Our method can be viewed as a general version of the Sparse Representation-based Classification (SRC) [19] that accounts for domain shift. An overview of the proposed method is shown in Figure 2.

We demonstrate the effectiveness of the proposed domain adaptation approach through comparisons with other recently proposed state-of-the-art domain adaptation methods on a challenging mobile active authentication dataset consisting of faces and screen touch gestures.

This paper makes the following contributions:

- A sparse representation-based classification algorithm is proposed for domain adaptation.
- An efficient iterative method based on the ADMM and the method of SOC is proposed for solving the proposed optimization problem.
- The algorithm is evaluated on a new dataset consisting of face and touch gestures collected from 50 mobile phone users in different ambient conditions.

The rest of the paper is organized as follows: The proposed domain adaptive sparse representation-based classification problem is defined in Section II. Details of the optimization algorithm are given in Section III. Experimental results are given in Section IV. Finally, Sections V concludes the paper with a brief summary and discussion.

II. Problem Formulation

Let $\{(\mathbf{y}_i^{d_1}, c_i^{d_1})\}_{i=1}^{N_1}$, denote the collection of N_1 labeled data from the domain $\mathcal{D}_1$. Here, $\mathbf{y}_i^{d_1} \in \mathbb{R}^{M_1}$ is referred to as the ith observation and $c_i^{d_1}$ is the corresponding class label. Labeled data from the domain $\mathcal{D}_2$ is denoted by $\{(\mathbf{y}_i^{d_2}, c_i^{d_2})\}_{i=1}^{N_2}$ where $\mathbf{y}_i^{d_2} \in \mathbb{R}^{M_2}$. Denote

$$\mathbf{Y}_1 = [\mathbf{y}_1^{d_1}, \cdots, \mathbf{y}_{N_1}^{d_1}] \in \mathbb{R}^{M_1 \times N_1}$$

as the matrix of N_1 data points from $\mathcal{D}_1$. Similarly, denote

$$\mathbf{Y}_2 = [\mathbf{y}_1^{d_2}, \cdots, \mathbf{y}_{N_2}^{d_2}] \in \mathbb{R}^{M_2 \times N_2}$$

as the matrix of N_2 data from $\mathcal{D}_2$. It is assumed that the data from both domains pertain to C subjects or classes. We assume that there is always a relatively large amount of labeled data in the source domain and a small amount of labeled data in the target domain. As a result, if $\mathcal{D}_1$ corresponds to the source domain and $\mathcal{D}_2$ corresponds to the target domain then $N_1 \gg N_2$.

Let $\mathbf{P}_1 \in \mathbb{R}^{m \times M_1}$ and $\mathbf{P}_2 \in \mathbb{R}^{m \times M_2}$ be mappings represented as matrices that project the data from $\mathcal{D}_1$ and $\mathcal{D}_2$ to a common m-dimensional space, respectively. As a results, $\mathbf{P}_1\mathbf{Y}_1$ and $\mathbf{P}_2\mathbf{Y}_2$ lie on an m-dimensional space. Let

$$\mathbf{Z} = [\mathbf{P}_1\mathbf{Y}_1, \mathbf{P}_2\mathbf{Y}_2] = [\mathbf{z}_1, \cdots, \mathbf{z}_{N_1+N_2}] \in \mathbb{R}^{m \times (N_1+N_2)}$$

denote the samples in the m-dimensional space. In our method, we want to take advantage of the *self-expressiveness property* of the data in the low-dimensional space [20]. That is, each data $\mathbf{z}_i$ can be efficiently reconstructed by a combination of other points in $\mathbf{Z}$. More precisely, $\mathbf{z}_i$ can be written as

$$\mathbf{z}_i = \mathbf{Z}\mathbf{b}_i, \quad b_{i,i} = 0, \tag{1}$$

where $\mathbf{b}_i = [b_{i,1}, b_{i,2}, \cdots, b_{i,N_1+N_2}]^T$. Here, the constraint $b_{i,i} = 0$ eliminates the trivial solution that arises as a result of representing a point as a linear combination of itself in the projected m-dimensional space. Assuming that $N_1 + N_2 \gg m$, (1) has infinitely many solutions. One can look for the sparsest solution and restrict the set of solutions by minimizing the following sparse optimization problem

$$\min \|\mathbf{b}_i\|_1 \quad \text{s.t.} \quad \mathbf{z}_i = \mathbf{Z}\mathbf{b}_i, \quad b_{i,i} = 0, \tag{2}$$

where $\|\mathbf{b}_i\|_1 = \sum_j |b_{i,j}|$ is the ℓ_1-norm of $\mathbf{b}_i$. This problem can be solved using convex optimization methods. One can rewrite the sparse optimization problem (2) for all samples in the m-dimensional space as

$$\min \|\mathbf{B}\|_1 \quad \text{s.t.} \quad \mathbf{Z} = \mathbf{Z}\mathbf{B}, \;\; \text{diag}(\mathbf{B}) = \mathbf{0}, \tag{3}$$

where $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_{N_1+N_2}] \in \mathbb{R}^{(N_1+N_2)\times(N_1+N_2)}$ is the coefficient matrix whose ith column is the sparse coefficient corresponding to $\mathbf{z}_i$, diag$(\mathbf{B})$ is the vector of the diagonal elements of $\mathbf{B}$ and $\|\mathbf{B}\|_1 = \sum_{i,j} |B_{i,j}|$ is the ℓ_1-norm of $\mathbf{B}$.

In our approach, we desire to learn projections $\mathbf{P}_1$ and $\mathbf{P}_2$ along with the sparse coefficient matrix $\mathbf{B}$ simultaneously by minimizing the following cost function

$$(\hat{\mathbf{P}}, \hat{\mathbf{B}}) = \min_{\mathbf{P},\mathbf{B}} \mathcal{C}_1(\mathbf{P}, \mathbf{Y}, \mathbf{B}) + \beta \mathcal{C}_2(\mathbf{P}, \mathbf{Y}) + \mu \|\mathbf{P}\mathbf{Y}\|_F^2$$
$$+ \lambda \|\mathbf{B}\|_1 \;\; s.t. \;\; \mathbf{P}_1\mathbf{P}_1^T = \mathbf{P}_2\mathbf{P}_2^T = \mathbf{I}, \;\; \text{diag}(\mathbf{B}) = \mathbf{0}, \tag{4}$$

where

$$\mathbf{P} = [\mathbf{P}_1 \;\; \mathbf{P}_2] \in \mathbb{R}^{m\times(M_1+M_2)}, \;\; \mathbf{B} \in \mathbb{R}^{(N_1+N_2)\times(N_1+N_2)},$$
$$\mathbf{Y} = \begin{bmatrix} \mathbf{Y}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{Y}_2 \end{bmatrix} \in \mathbb{R}^{(M_1+M_2)\times(N_1+N_2)},$$
$$\mathcal{C}_1(\mathbf{P}, \mathbf{Y}, \mathbf{B}) = \|\mathbf{P}\mathbf{Y} - \mathbf{P}\mathbf{Y}\mathbf{B}\|_F^2,$$

and

$$\mathcal{C}_2(\mathbf{P}, \mathbf{Y}) = \|\mathbf{Y}_1 - \mathbf{P}_1^T\mathbf{P}_1\mathbf{Y}_1\|_F^2 + \|\mathbf{Y}_2 - \mathbf{P}_2^T\mathbf{P}_2\mathbf{Y}_2\|_F^2.$$

Here, β, μ and λ are the regularization parameters. After ignoring the constant terms in $\mathbf{Y}$, $\mathcal{C}_2$ can be rewritten as

$$\mathcal{C}_2(\mathbf{P}, \mathbf{Y}) = -\text{tr}((\mathbf{P}\mathbf{Y})(\mathbf{P}\mathbf{Y})^T).$$

The first part of the cost function $\mathcal{C}_1$ with the constraint that diag$(\mathbf{B}) = \mathbf{0}$ essentially exploits the self-expressiveness property of the data in the sense that each data point can be efficiently reconstructed by a combination of other points in the database. Similar ideas have been explored for subspace clustering using sparse representation in [20]. The second term $\mathcal{C}_2$ is a PCA-like regularization term, ensures that the projection does not loose too much information available in the original domain. Finally, $\|\mathbf{P}\mathbf{Y}\|_F^2$ is added to ensure the convexity of the cost function.

A. Multiple Domains

The above formulation can be extended from two domains to multiple domains. For K domain problem, we have data $\mathbf{Y}_1, \cdots, \mathbf{Y}_K$ from K different domains $\mathcal{D}_1, \cdots, \mathcal{D}_K$ and one can simply construct $\mathbf{P}$ and $\mathbf{Y}$ as

$$\mathbf{P} = [\mathbf{P}_1 \cdots \mathbf{P}_K], \;\; \mathbf{Y} = \begin{bmatrix} \mathbf{Y}_1 & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{Y}_K \end{bmatrix}.$$

With these definitions, (4) can be extended to multiple domains. Note that we do not require the dimensions from different domains to be the same. As a result, our method can be viewed as a heterogeneous domain adaptation method [1].

III. OPTIMIZATION

We solve the optimization problem (4) by optimizing over $\mathbf{P}$ and $\mathbf{B}$ iteratively. Note that the optimization problem is non-convex. However, numerical simulations have shown that the algorithm usually converges to a local minimum in a few iterations.

A. Update step for $\mathbf{B}$

In this step, we assume that $\mathbf{P}$ is fixed. As a result, the following problem needs to be solved

$$\min_{\mathbf{B}} \mathcal{C}_1(\mathbf{P}, \mathbf{Y}, \mathbf{B}) + \lambda \|\mathbf{B}\|_1 \;\; s.t. \;\; \text{diag}(\mathbf{B}) = \mathbf{0}. \tag{5}$$

This problem is similar to the Sparse Subspace Clustering (SSC) problem [20] which can be efficiently solved using the ADMM method [21].

B. Update step for $\mathbf{P}$

For a fixed $\mathbf{B}$, we have to solve the following problem to obtain $\mathbf{P}$

$$\min_{\mathbf{P}} \mathcal{C}_1(\mathbf{P}, \mathbf{Y}, \mathbf{B}) + \beta \mathcal{C}_2(\mathbf{P}, \mathbf{Y}) + \mu \|\mathbf{P}\mathbf{Y}\|_F^2$$
$$s.t. \;\; \mathbf{P}_1\mathbf{P}_1^T = \mathbf{P}_2\mathbf{P}_2^T = \mathbf{I}. \tag{6}$$

The cost in (6) can be rewritten as

$$\begin{aligned} &\mathcal{C}_1(\mathbf{P}, \mathbf{Y}, \mathbf{B}) + \beta \mathcal{C}_2(\mathbf{P}, \mathbf{Y}) + \mu \|\mathbf{P}\mathbf{Y}\|_F^2 \\ &= \|\mathbf{P}\mathbf{Y} - \mathbf{P}\mathbf{Y}\mathbf{B}\|_F^2 + (\mu - \beta)\text{tr}((\mathbf{P}\mathbf{Y})(\mathbf{P}\mathbf{Y})^T) \\ &= \text{tr}[(\mathbf{P}\mathbf{Y} - \mathbf{P}\mathbf{Y}\mathbf{B})^T(\mathbf{P}\mathbf{Y} - \mathbf{P}\mathbf{Y}\mathbf{B}) + (\mu - \beta)(\mathbf{P}\mathbf{Y})(\mathbf{P}\mathbf{Y})^T] \\ &= \text{tr}[\mathbf{P}(\mathbf{Y}(\mathbf{I} - 2\mathbf{B} + \mathbf{B}\mathbf{B}^T + (\mu - \beta)\mathbf{I})\mathbf{Y}^T)\mathbf{P}^T]. \end{aligned}$$

Let $\mathbf{H} = \mathbf{Y}(\mathbf{I} - 2\mathbf{B} + \mathbf{B}\mathbf{B}^T + (\mu - \beta)\mathbf{I})\mathbf{Y}^T$ be $\sum_i M_i \times \sum_i M_i$ matrix. Then, the optimization problem (6) can be rewritten as

$$\min_{\mathbf{P}} \text{tr}[\mathbf{P}\mathbf{H}\mathbf{P}^T] \;\; s.t. \;\; \mathbf{P}_1\mathbf{P}_1^T = \mathbf{P}_2\mathbf{P}_2^T = \mathbf{I}. \tag{7}$$

This optimization problem involves trace minimization with multiple orthogonality constraints. The cost function is convex when $\mathbf{H}$ is positive semi-definite; however multiple orthogonality constraints make the problem not convex and we cannot directly solve it as a classical eigen problem. In what follows, we present the method of SOC for solving this problem [22].

1) Optimization for (7): Let $\mathbf{O} = \mathbf{P}^T$. Then, the trace minimization problem (7) with K orthogonality constants can be rewritten as

$$\min_{\mathbf{O}} g(\mathbf{O}_1, \cdots, \mathbf{O}_K; \mathbf{H}) \;\; s.t. \;\; \mathbf{O}_i^T\mathbf{O}_i = \mathbf{I} \;\; \forall i = 1, \cdots, K, \tag{8}$$

where $\mathbf{O}_i \in \mathbb{R}^{M_i \times m}$, $m \le \min\{M_1, M_2, \cdots, M_K\}$,

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_{11} & \mathbf{H}_{12} & \cdots & \mathbf{H}_{1K} \\ \mathbf{H}_{21} & \mathbf{H}_{22} & \cdots & \mathbf{H}_{2K} \\ \mathbf{H}_{K1} & \mathbf{H}_{K2} & \cdots & \mathbf{H}_{KK} \end{bmatrix} \in \mathbb{R}^{\sum_i M_i \times \sum_i M_i},$$

$\mathbf{H}_{ij} \in \mathbb{R}^{M_i \times M_j}$ and

$$g(\mathbf{O}_1, \cdots, \mathbf{O}_K; \mathbf{H}) = \text{tr}[\mathbf{O}^T\mathbf{H}\mathbf{O}].$$

The SOC method solves the orthogonality constrained problems by iteratively optimizing the unconstrained and quadratic problems with analytic solutions using the combination of variable splitting and Bregman iteration [23]. It consists of three main steps.

Update $\mathbf{O}_i$: For updating $\mathbf{O}_i$ one at a time, we need to solve the following sub optimization problem

$$\mathbf{O}_i^t = \arg\min_{\mathbf{O}_i} g(\mathbf{O}_1^{t-1}, \cdots, \mathbf{O}_K^{t-1}) + \frac{\gamma}{2}\|\mathbf{O}_i - \mathbf{Q}_i^{t-1} + \mathbf{R}_i^{t-1}\|_F^2.$$

Where γ is a positive parameter that can be tuned. By taking the first derivative and setting it equal to zero, we get

$$\mathbf{O}_i^t = \left(\frac{\gamma}{2}\mathbf{I} + \mathbf{H}_{ii}\right)^{-1} \left[\frac{\gamma}{2}(\mathbf{Q}_i^{t-1} - \mathbf{R}_i^{t-1}) - \sum_{\substack{j=1\\ j\neq i}}^{K} \mathbf{H}_{ij}\mathbf{O}_j^{t-1}\right].$$

Update $\mathbf{Q}_i$: In order to update $\mathbf{Q}_i$, we need to solve the following optimization problem

$$\mathbf{Q}_i^t = \arg\min_{\mathbf{Q}} \frac{\gamma}{2}\|\mathbf{Q}_i - (\mathbf{O}_i^t - \mathbf{R}_i^{t-1})\|_F^2 \;\; s.t \;\; \mathbf{Q}_i^T\mathbf{Q}_i = \mathbf{I} \quad (9)$$

whose closed form solution is obtained as

$$\mathbf{Q}_i^t = \mathbf{U}_i \mathbf{I}_{M_i\times m}\mathbf{V}_i^T,$$

where $\mathbf{U}_i\mathbf{D}_i\mathbf{V}_i^T$ is the Singular Value Decomposition (SVD) of $(\mathbf{O_i^t} - \mathbf{R_i^{t-1}})$ and $\mathbf{U}_i \in \mathbb{R}^{M_i\times M_i}, \mathbf{D}_i \in \mathbb{R}^{M_i\times m}$, $\mathbf{V}_i \in \mathbb{R}^{m\times m}$.

Update $\mathbf{R}_i$: Finally, having updated $\mathbf{Q}_i$ and $\mathbf{O}_i$, $\mathbf{R}_i$ is updated as follows

$$\mathbf{R}_i^t = \mathbf{R}_i^{t-1} + \left(\mathbf{O}_i^t - \mathbf{Q}_i^t\right).$$

The entire procedure for solving (8) using the method of SOC is summarized in Algorithm 1.

Algorithm 1: The method of SOC for solving (8).

Input: $\mathbf{O}, \mathbf{H}, \gamma$
Initialization:$\mathbf{R}_0, \mathbf{O}_0, \mathbf{Q}_0$
While not converge do
1. Update $\mathbf{O}_i$:

$$\mathbf{O}_i^t = \left(\frac{\gamma}{2}\mathbf{I} + \mathbf{H}_{ii}\right)^{-1} \left[\frac{\gamma}{2}(\mathbf{Q}_i^{t-1} - \mathbf{R}_i^{t-1}) - \sum_{\substack{j=1\\ j\neq i}}^{K} \mathbf{H}_{ij}\mathbf{O}_j^{t-1}\right]$$

2. Update $\mathbf{Q}_i$:

$$\mathbf{Q}_i^t = \mathbf{U}_i \mathbf{I}_{M_i\times m}\mathbf{V}_i^T$$

3. Update $\mathbf{R}_i$:

$$\mathbf{R}_i^t = \mathbf{R}_i^{t-1} + (\mathbf{O}_i^t - \mathbf{Q}_i^t)$$

Output: $\hat{\mathbf{O}} = [\mathbf{O}_1^t, \cdots, \mathbf{O}_K^t]$

The training part of the Domain Adaptive Sparse Representation-based Classification (DASRC) algorithm is summarized in Algorithm 2.

C. Classification

Given a test sample $\mathbf{y}_t$ from domain k, we propose the following steps for classification.

1) Compute the embeddings of all the training samples from different domains in the common m-dimensional subspace using the corresponding projections as $\mathbf{P}_i\mathbf{Y}_i \in m \times N_i, \forall i$.

Algorithm 2: Training part of the Domain Adaptive Sparse Representation-based Classification (DASRC) algorithm.

Input: Data $\mathbf{Y}_1, \cdots, \mathbf{Y}_K$ and corresponding class labels, β, μ, λ
Initialization: P
While not converge do
1. Update $\mathbf{B}$: Solve the following ℓ_1 minimization problem using the ADMM procedure described in [20]

$$\min_{\mathbf{B}} \mathcal{C}_1(\mathbf{P}, \mathbf{Y}, \mathbf{B}) + \lambda\|\mathbf{B}\|_1 \;\; s.t. \;\; \mathrm{diag}(\mathbf{B}) = \mathbf{0}$$

2. Update $\mathbf{P}$: Solve the following optimization problem using the method of SOC as summarized in Algorithm 1.

$$\min_{\mathbf{P}} \mathrm{tr}[\mathbf{P}\mathbf{H}\mathbf{P}^T] \;\; s.t. \;\; \mathbf{P}_1\mathbf{P}_1^T = \mathbf{P}_2\mathbf{P}_2^T = \mathbf{I}$$

Output: $\hat{\mathbf{B}}$ and $\hat{\mathbf{P}} = [\hat{\mathbf{P}}_1, \hat{\mathbf{P}}_2, \cdots, \hat{\mathbf{P}}_K]$

2) Using the label information, form a training matrix in the low-dimensional subspace as follows

$$\mathbf{Z} = [\mathbf{Z}_1, \mathbf{Z}_2, \cdots, \mathbf{Z}_C] \in \mathbb{R}^{m\times\sum_i N_i},$$

where $\mathbf{Z}_i$ is the matrix corresponding to the training samples from class i in the m-dimensional space.

3) Compute the embedding of the test sample $\mathbf{y}_t$ in the common m-dimensional subspace using the projection $\mathbf{P}_k$ as

$$\mathbf{z}_t = \mathbf{P}_k\mathbf{y}_t.$$

4) Compute the sparse coefficient $\hat{\boldsymbol{\alpha}}_t$ of the embedded sample $\mathbf{z}_t$ over dictionary $\mathbf{Z}$ by solving the following optimization problem

$$\hat{\boldsymbol{\alpha}}_t = \min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}_t\|_0 \;\; s.t. \;\; \|\mathbf{z}_t - \mathbf{Z}\boldsymbol{\alpha}_t\|_F^2 \leq \eta, \qquad (10)$$

where η is the noise level and $\|\mathbf{x}\|_0$ is the ℓ_0-norm of $\mathbf{x}$ which counts the number of non-zero elements in $\mathbf{x}$. We use the Orthogonal Matching Pursuit (OMP) algorithm [24] to solve (10).

5) The sample can be assigned to class i if the reconstruction using the samples corresponding to class i is minimum

$$\text{Output class} = \hat{i} = \arg\min_i \|\mathbf{z}_t - \mathbf{Z}\delta_i(\hat{\boldsymbol{\alpha}}_t)\|_F^2,$$

where $\delta_i(\cdot)$ is the characteristic function that selects the coefficients associated with the ith class.

IV. EXPERIMENTAL RESULTS

In this section, we evaluate the proposed algorithm on a face and touch gesture dataset collected by the authors' group using a mobile device for active authentication in different ambient conditions. We compare our method with several recent domain adaptation algorithms including a metric learning-based method [6], a manifold-based method [4], and dictionary learning-based methods [16], [14]. We also use the SRC method [19] as a baseline comparison. For SRC, data from different domains are used without domain adaptation. This method essentially shows the performance of a sparsity-based method when training and test samples come from different domains. Comparison of our DASRC method with SRC will validate the effectiveness of the proposed domain adaptation approach.

A. Face and Screen Touch Gesture Dataset

Most mobile devices use passwords, pin numbers, or secret patterns for authenticating users. As long as the device remains active, there is no mechanism to verify that the user originally authenticated is still the user in control of the device. As a result, unauthorized individuals may improperly gain access to personal information of the user if the password is compromised. Active Authentication systems deal with this issue by continuously monitoring the user identity after the initial access has been granted. Examples of such systems include screen touch gesture-based recognition [25] and gait-based recognition [26].

Faces have shown to be very promising physiological biometric. In order to study the effectiveness of both faces and touch gestures for active authentication, we collected data from 50 iPhone 5s users in an application environment. The users were asked to perform difference tasks such as scrolling a document, viewing pictures, reading a long article etc. While users performed these tasks, their touch data sensed by the screen and face images acquired by the front-facing camera were simultaneously captured. The users were asked to perform these tasks in different sessions with different ambient conditions, namely in a well-lit room, in a dim-lit room, and in a room with natural daytime illumination. During data collection, users were free to use the phone in either orientation mode and hold the phone in any position of their choice. The goal was to simulate real-world scenarios to study how ambient changes can influence users' face data captured by the frontal camera and touch gestures on the screen. Data collection from 50 users over 3 sessions resulted in 750 videos consisting of facial data with each video lasting between 0.5 minute to 2 minutes and 15490 touch gestures. It is a very challenging dataset. Since facial video data were collected in an unconstrained manner, many faces exhibit different poses, blur and illuminations. In particular, partial faces are common in this dataset. Figures 1 shows sample face images from this dataset. Each row shows images from a particular ambient condition. Samples of touch data from this dataset are shown in Figure 3.

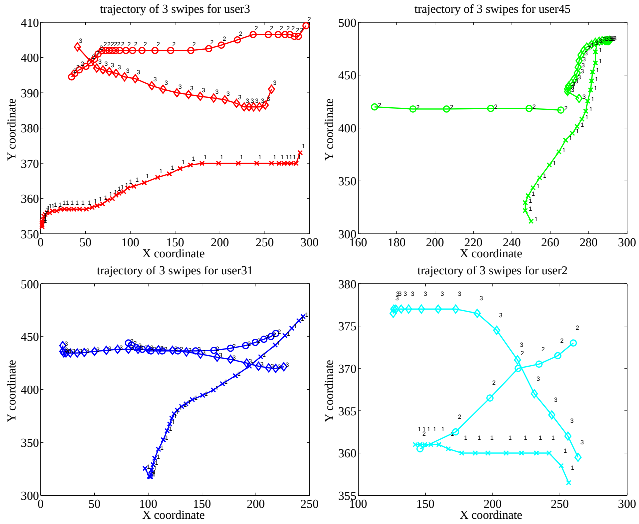
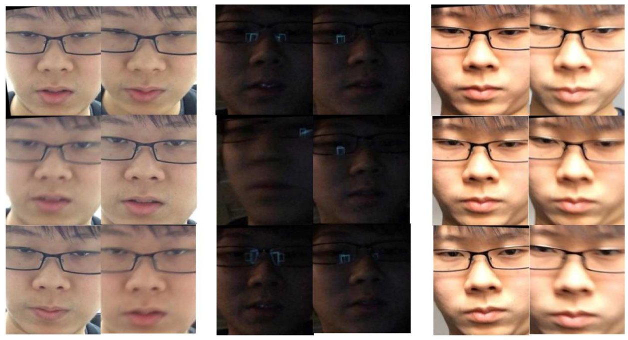


Fig. 3: Trajectory of touch data samples from our dataset.



(a) (b) (c)

Fig. 4: One subject's face images after alignment. (a), (b) and (c) show face images from different sessions (domains), respectively. As can be seen, the face images of the same person from different sessions vary significantly.

B. Preprocessing and Feature Extraction

Since this dataset consists of two modalities, we perform preprocessing and feature extraction for face and screen touch data separately.

1) Faces: For the face data, we first detect the landmarks of the face images frame by frame from the videos using the tree-based landmarks detector [27]. We then crop and align the faces using the methods described in [28] based on the landmarks' locations. The face images are then rescaled to dimension $192 \times 168 \times 3$. Some examples of detected and aligned faces are shown in Figure 4.

After aligning the face images we transformed them into grayscale images and applied the illumination normalization step described in [29]. Finally, we down sampled the images to 16 by 14 and used the whole image as a feature vector of dimension 224.

2) Touch Gestures: Every touch swipe $\mathbf{S}$ was encoded as a sequence of vectors

$$\mathbf{s}_i = (x_i, y_i, t_i, A_i, o_i^{ph}),$$

$i \in \{1, \cdots, N_c\}$ where x_i, y_i are the location points, t_i is the time stamp, A_i is the area occluded by the finger and o_i^{ph} is the orientation of the phone (e.g. landscape or portrait). Given these touch data, we extracted a 27 dimensional feature vector for every single swipe in the dataset using the method described in [25]. These features are summarized in Table III.

C. Experimental Setup

The new dataset is a relative large dataset. In order to evaluate the proposed domain adaptation methods, we sampled a small subset from this dataset. For the face component, for each user, we selected 30 faces from each session. As a result, in total we selected 4500 face images for 50 users across 3 different domains. For the touch signature, we also

Methods	$1 \rightarrow 2$	$1 \rightarrow 3$	$2 \rightarrow 3$	$2 \rightarrow 1$	$3 \rightarrow 1$	$3 \rightarrow 2$	Average
SRC [19]	73.52 ±1.49	85.12 ± 1.04	83.98± 0.91	80.83 ±1.08	80.73 ±1.35	72.57 ±1.13	79.46
Metric [6]	73.19 ±1.95	84.54 ± 1.27	80.36 ± 2.92	78.83 ±4.06	85.45 ±1.15	73.61 ±2.18	79.33
SGF [4]	56.57 ±1.22	62.58 ± 1.13	60.90 ± 1.05	54.94 ±2.19	65.66 ±1.75	62.69 ±1.33	60.56
SDDL [14]	55.48 ±4.40	71.67 ± 4.14	75.67 ± 3.72	71.71 ±4.46	77.74 ±4.15	66.74 ±2.91	69.84
Dict [16]	66.13 ±1.40	78.61 ± 1.42	76.26 ± 0.63	72.30 ±1.24	78.18 ±1.50	71.15 ±1.24	73.77
DASRC	$\mathbf{81.39 \pm 1.66}$	$\mathbf{89.06 \pm 1.31}$	$\mathbf{89.70 \pm 1.05}$	$\mathbf{87.36 \pm 0.82}$	$\mathbf{86.92 \pm 0.99}$	$\mathbf{82.16 \pm 0.69}$	**86.10**

TABLE I: Recognition accuracy on target domain with semi-supervised adaptation for the face component.

Methods	$1 \rightarrow 2$	$1 \rightarrow 3$	$2 \rightarrow 3$	$2 \rightarrow 1$	$3 \rightarrow 1$	$3 \rightarrow 2$	Average
SRC [19]	35.48 ±1.49	37.50 ± 0.86	40.18± 1.27	36.99 ±1.10	37.57 ±1.23	38.50 ±0.73	37.70
Metric [6]	24.58 ±1.75	25.71 ± 0.92	29.58 ± 2.22	22.45 ±2.07	24.25 ±1.90	28.59 ±1.47	25.86
SGF [4]	37.88 ±1.18	35.47 ± 1.25	37.00 ± 0.97	37.08 ±1.28	36.10 ±1.20	41.54 ±1.22	37.51
SDDL [14]	39.49 ±2.73	41.86 ± 2.36	42.28± 2.38	38.71±3.65	39.66 ±2.90	38.98 ±3.26	40.16
Dict [16]	30.31 ±1.39	31.00 ± 0.74	34.74 ± 1.05	30.58 ±0.94	32.55 ±0.73	36.21 ±0.82	32.57
DASRC	$\mathbf{41.54 \pm 1.89}$	$\mathbf{44.34 \pm 1.66}$	$\mathbf{44.77 \pm 1.17}$	$\mathbf{41.58 \pm 1.35}$	$\mathbf{41.82 \pm 1.61}$	$\mathbf{42.30 \pm 1.50}$	**42.74**

TABLE II: Recognition accuracy on target domain with semi-supervised adaptation for the touch component.

FeatureID	Description
feature 1	inter-stroke time
feature 2	stroke duration
feature 3	start x
feature 4	start y
feature 5	stop x
feature 6	stop y
feature 7	direct end-to-end distance
feature 8	mean resultant length
feature 9	up/down/left/right flag
feature 10	direction of end-to-end line
feature 11	20%-perc. pairwise velocity
feature 12	50%-perc. pairwise velocity
feature 13	80%-perc. pairwise velocity
feature 14	20%-perc. pairwise acceleration
feature 15	50%-perc. pairwise acceleration
feature 16	80%-perc. pairwise acceleration
feature 17	median velocity at last 3 points
feature 18	largest deviation from end-to-end line
feature 19	20%-perc. dev. from end-to-end line
feature 20	50%-perc. dev. from end-to-end line
feature 21	80%-perc. dev. from end-to-end line
feature 22	average direction
feature 23	length of trajectory
feature 24	ratio end-to-end dist and length of trajectory
feature 25	average velocity
feature 26	median acceleration at first 5 points
feature 27	mid-stroke area covered

TABLE III: Description of the 27 dimensional feature vector.

selected 4500 touch swipes of 50 users across 3 domains. All the experiment done will be based on these selected 4500 face images and 4500 touch swipes. This part of the data and the Matlab implementation of our DASRC method will be made available for research purposes. Because the underlying characteristics of the data collected in different sessions with different ambient conditions is very different, data in different sessions can be viewed as data from different domains. Therefore, it is necessary to apply domain adaptation methods to design classifiers that are robust to different sessions (domains). For the proposed DASRC algorithm, we choose $\mu - \beta = 4.5$, $\lambda = 50$ and $\gamma = 60$ which are the tuned results from the cross validation experiments. Parameters for the other domain adaptation methods were optimized according to the discussion provided in the corresponding papers.

D. Single-source Domain Adaptation Experiments

Following the standard domain adaptation protocol, we selected 20 samples for each user from one session as the source domain and 5 samples for each user from another session as the target domain to form the training data. The remaining data from the target domain were used for testing. We randomly split the training and testing datasets, and repeated each experiment 10 times and report the mean and the standard deviation of the classification accuracy. Since we have 3 sessions, there are 6 different combinations of source and target domains. The performance of our proposed method is compared with the other domain adaptation methods for the face and the touch data in Table I and Table II, respectively.

As can be seen from these tables, the proposed DASRC method outperforms the other methods on all 6 domain pairs. In some cases the improvement is over 10% compared to the other methods. Furthermore, comparison with the SRC method shows that the sparse coding framework is insufficient when the test data has different characteristics than the data used for training. Also, the performance on faces is better than the performance on touch gestures.

E. Multi-source Domain Adaptation Experiments

For multi-source domain adaptation experiments, we selected 20 samples for each user from source domains and 5 samples for each user from the target domain to form the training data. The remaining data from the target domain were used for testing. Like before, we repeated each experiment 10 times and report the mean and the standard deviation of the classification accuracy. Since we have 3 sessions, there are 3 different combinations of the source and the target domains. The experimental results comparing our proposed method with the other multi-source domain adaptation methods on the face data and the touch data are shown in Table IV and Table V, respectively.

Again, our DASRC method performs better than the other methods on all possible combinations. An interesting observation is that increasing the number of domains can be helpful, especially when compared to a single source and single target cases. This can be seen by comparing Tables I and II with tables IV and V. The gain is more apparent for faces.

F. Visualization of the Projection Matrices

To further gain insights regarding our method, we investigated the projection matrices $\mathbf{P}_i \in \mathbb{R}^{m \times M_i}, \forall i = 1, \cdots, K$ learned by our method in the case of multi-source domain adaptation using faces. For better visualization, we used grayscale face images rescaled to 128×128 from the original preprocessed face images of size $192 \times 168 \times 3$. We followed the multi-source domain adaptation experiment setup as described above. We chose session 1 and 2 to be the source domains and session 3 to be the target domain. We first randomly selected 20 images per subject in each source domain, and 5 images per subject in the target domain, and then fed these images to our proposed algorithm to learn the projection matrices $\mathbf{P}_1$, $\mathbf{P}_2$ and $\mathbf{P}_3$. Figure 5 shows the first 6 rows of the learned projection matrices reshaped as images. As can be seen from this figures, the projection matrices learn the internal structure of the different domains and can capture the shape, illumination and pose information. As a result, we are able to find better sparse representation in the projected m-dimensional space.

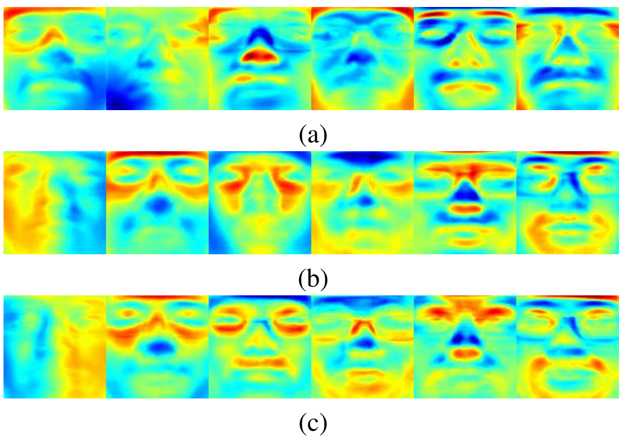


Fig. 5: First 6 components of the learned projection matrices for the multi-source domain adaptation experiment. (a) Components from $\mathbf{P}_1$, (b) Components from $\mathbf{P}_2$. (c) Components from $\mathbf{P}_3$.

G. Runtime Analysis and Computational Issue

In this section, we study the convergence properties of the proposed method and briefly discuss the computational timing compared to the dictionary-based domain adaptation algorithms [16], [14].

As discussed earlier, our method is non-convex and often converges to a local minima in a few iterations. To empirically show the convergence of our method, in Fig 6(a)-(c), we show the objective function vs iteration plots for the ADMM method for solving (5), the method of SOC for solving the trace minimization problem with multiple orthogonality constraints (6) and our proposed problem (4), respectively. As can be seen from this figure, both sub optimization problems as well as our overall algorithm do converge in a few iterations. Furthermore, compared to the previously proposed dictionary-based domain adaptation methods, our method is very efficient. On average, the proposed method takes about $6.5ms$ to recognize a test image of size 24×21 compared to $26ms$ and $11ms$ for [16] and [14], respectively. Experiments were done in 64bit Matlab R2013a environment on a laptop with 2.9GHz Intel Core i7-3520M CPU and 8GB Memory.

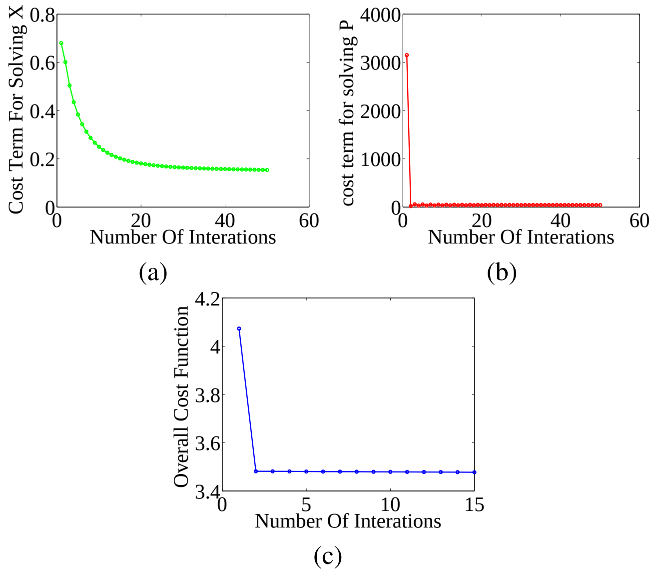


Fig. 6: Objective function versus number of iterations of the proposed optimization problems. (a) The ADMM method for solving (5). (b) The method of SOC for solving the trace minimization problem with multiple orthogonality constraints (6). (c) The proposed problem (4).

V. CONCLUSION

In this paper, we proposed a sparsity-based framework for solving the domain adaption problems. The proposed DASRC algorithm is applicable to single-source domain, multi-source and heterogeneous domain adaptation problems. We proposed an iterative algorithm consisting of the ADMM method and the SOC method for solving the optimization problem. Extensive experiments on a mobile dataset consisting of faces and touch gestures showed that our method can perform better than many state-of-the-art domain adaptation methods.

Our future work will investigate learning discriminative and manifold preserving projections for better classification. Furthermore, we will perform a theoretical analysis of the proposed work. We will also evaluate the performance of our method on other cross-domain object recognition dataset.

Methods	1 2 → 3	1 3 → 2	2 3 → 1	Average
SRC [19]	89.68 ±0.83	81.14 ± 0.86	88.20 ± 0.76	86.34
SGF [4]	69.57 ±1.35	64.05 ± 1.21	62.21 ± 2.12	65.28
SDDL [14]	75.08 ±3.82	55.34 ± 2.34	72.86 ± 3.27	67.76
LMSDA [30]	82.48 ±1.04	70.17 ± 0.66	77.18 ± 1.18	76.61
DASRC	$\mathbf{90.94 \pm 0.86}$	$\mathbf{83.03 \pm 0.74}$	$\mathbf{88.44 \pm 0.68}$	**87.47**

TABLE IV: Multi-source domain adaptation on face data.

Methods	1 2 → 3	1 3 → 2	2 3 → 1	Average
SRC [19]	39.88 ±1.10	38.26 ± 0.63	36.68 ± 1.28	38.27
SGF [4]	39.04 ±0.99	39.13 ± 1.11	35.96 ± 0.94	38.04
SDDL [14]	34.66 ±1.50	31.21 ± 2.98	31.26 ± 2.56	32.38
LMSDA [30]	40.86 ±1.21	39.20 ± 0.79	37.42 ± 1.22	39.16
DASRC	$\mathbf{43.62 \pm 1.75}$	$\mathbf{42.17 \pm 1.14}$	$\mathbf{42.40 \pm 0.83}$	**42.73**

TABLE V: Multi-source domain adaptation on touch data.

REFERENCES

[1] V. M. Patel, R. Gopalan, R. Li, and R. Chellappa, "Visual domain adaptation: A survey of recent advances," *IEEE Signal Processing Magazine*, 2015.

[2] H. Daume III, "Frustratingly easy domain adaptation," in *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Prague, Czech Republic: Association for Computational Linguistics, June 2007, pp. 256–263.

[3] W. Li, L. Duan, D. Xu, and I. Tsang, "Learning with augmented features for supervised and semi-supervised heterogeneous domain adaptation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 6, pp. 1134–1148, June 2014.

[4] R. Gopalan, R. Li, and R. Chellappa, "Unsupervised adaptation across domain shifts by generating intermediate data representations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 11, pp. 2288–2302, 2014.

[5] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2066–2073.

[6] K. Saenko, B. Kulis, M. Fritz, and T. Darrell., "Adapting visual category models to new domains," in *European Conference on Computer Vision*, vol. 6314, 2010, pp. 213–226.

[7] B. Kulis, K. Saenko, and T. Darrell, "What you saw is not what you get: Domain adaptation using asymmetric kernel transforms," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 1785–1792.

[8] I.-H. Jhuo, D. Liu, D. Lee, and S.-F. Chang, "Robust visual domain adaptation with low-rank reconstruction," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2168–2175.

[9] J. Yang, R. Yan, and A. G. Hauptmann, "Cross-domain video concept detection using adaptive SVMs." in *ACM Multimedia*. ACM, 2007, pp. 188–197.

[10] L. Duan, I. W.-H. Tsang, D. Xu, and S. J. Maybank, "Domain transfer svm for video concept detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1375–1381.

[11] J. Hoffman, E. Rodner, J. Donahue, T. Darrell, and K. Saenko, "Efficient learning of domain-invariant image representations," in *International Conference on Learning Representations*, 2013.

[12] L. Duan, I. Tsang, and D. Xu, "Domain transfer multiple kernel learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 3, pp. 465–479, 2012.

[13] S. Wang, L. Zhang, L. Y., and Q. Pan, "Semi-coupled dictionary learning with applications in image super-resolution and photo-sketch synthesis," in *International Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2012.

[14] S. Shekhar, V. M. Patel, H. V. Nguyen, and R. Chellappa, "Generalized domain-adaptive dictionaries," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[15] Q. Qiu, V. M. Patel, P. Turaga, and R. Chellappa, "Domain adaptive dictionary learning," in *European Conference on Computer Vision*, vol. 7575, 2012, pp. 631–645.

[16] J. Ni, Q. Qiu, and R. Chellappa, "Subspace interpolation via dictionary learning for unsupervised domain adaptation," in *IEEE International Conference on Computer Vision*, 2013.

[17] J. Jiang, "A literature survey on domain adaptation of statistical classifiers." *Tech report*, 2008.

[18] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[19] 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.

[20] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2765–2781, Nov 2013.

[21] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, Jan 2011.

[22] R. Lai and S. Osher, "A splitting method for orthogonality constrained problems," *Journal of Scientific Computing*, vol. 58, no. 2, pp. 431–449, Feb 2014.

[23] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin, "An iterative regularization method for total variation-based image restoration," *Multiscale Modeling & Simulation*, vol. 4, no. 2, pp. 460–489, 2005.

[24] Y. Pati, R. Rezaiifar, and P. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition," in *Asilomar Conference on Signals, Systems and Computers*, 1993.

[25] M. Frank, R. Biedert, E. Ma, I. Martinovic, and D. Song, "Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 1, pp. 136–148, Jan 2013.

[26] M. Derawi, C. Nickel, P. Bours, and C. Busch, "Unobtrusive user-authentication on mobile phones using biometric gait recognition," in *International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, Oct 2010, pp. 306–311.

[27] D. Ramanan, "Face detection, pose estimation, and landmark localization in the wild," in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, ser. CVPR '12, 2012, pp. 2879–2886.

[28] V. Štruc and N. Pavešić, "The complete gabor-fisher classifier for robust face recognition," *EURASIP Advances in Signal Processing*, vol. 2010, p. 26, 2010.

[29] X. Tan and B. Triggs, "Enhanced local texture feature sets for face recognition under difficult lighting conditions," in *Proceedings of the 3rd International Conference on Analysis and Modeling of Faces and Gestures*, ser. AMFG'07, 2007, pp. 168–182.

[30] J. Hoffman, B. Kulis, T. Darrell, and K. Saenko, "Discovering latent domains for multisource domain adaptation," 2012.