Cross-Domain Recognition by Identifying Joint Subspaces of Source Domain and Target Domain

Yuewei Lin, *Student Member, IEEE*, Jing Chen, Yu Cao, *Member, IEEE*, Youjie Zhou, Lingfeng Zhang, *Student Member, IEEE*, Yuan Yan Tang, *Fellow, IEEE*, and Song Wang, *Senior Member, IEEE*

***Abstract*—This paper introduces a new method to solve the cross-domain recognition problem. Different from the traditional domain adaption methods which rely on a global domain shift for all classes between the source and target domains, the proposed method is more flexible to capture individual class variations across domains. By adopting a natural and widely used assumption that the data samples from the same class should lay on an intrinsic low-dimensional subspace, even if they come from different domains, the proposed method circumvents the limitation of the global domain shift, and solves the cross-domain recognition by finding the joint subspaces of the source and target domains. Specifically, given labeled samples in the source domain, we construct a subspace for each of the classes. Then we construct subspaces in the target domain, called anchor subspaces, by collecting unlabeled samples that are close to each other and are highly likely to belong to the same class. The corresponding class label is then assigned by minimizing a cost function which reflects the overlap and topological structure consistency between subspaces across the source and target domains, and within the anchor subspaces, respectively. We further combine the anchor subspaces to the corresponding source subspaces to construct the joint subspaces. Subsequently, one-versus-rest support vector machine classifiers are trained using the data samples belonging to the same joint subspaces and applied to unlabeled data in the target domain. We evaluate the proposed method on two widely used datasets: 1) object recognition dataset for computer vision tasks and 2) sentiment classification dataset for natural language processing tasks. Comparison results demonstrate that the proposed method outperforms the comparison methods on both datasets.**

***Index Terms*—Cross domain recognition, joint subspace, unsupervised.**

I. Introduction

MANY machine learning methods often assume that the training data (labeled) and testing data (unlabeled) are from the same feature space and follow similar distributions. However, this assumption may not be true in many real applications. Namely, the training data is obtained from one domain, while the testing data come from a different domain. As a visual example, Fig. 1 shows coffee-mug images collected from four different domains [Amazon, Caltech, digital single-lens reflex (DSLR), and Webcam], which present different image resolutions (Webcam versus DSLR), viewpoints (Webcam versus Amazon), background complexities (Amazon versus Caltech), and object layout patterns, etc.

On the other hand, the data samples also show different distributions in the feature space, as illustrated in Fig. 2. The 2-D plots are the first two feature dimensions reduced from original 800-D speeded up robust features (SURF) feature space (described in Section IV-D), using principal component analysis (PCA). The first and the second rows of Fig. 2 show the distributions of "monitor" and "projector" and "mouse" and "mug" in different domains, respectively. It is clear to see that the data samples from different domains have different distributions. Moreover, the relations between two classes in different domains are also different. Taking the second row of Fig. 2 as an example, in Webcam domain (the last column), the mug samples (green circles) are usually located at the top-right side of the mouse ones (red circles); but in Amazon domain (the first column), the mug samples (green circles) are usually located at the left side of the mouse ones (red circles).

These domain differences lead to a dilemma that: 1) directly applying the classifiers trained from one domain to another may result in significant degraded performance [2] and 2) labeling data in each domain as training samples would be very expensive, especially in large-scale applications. The dilemma consequently poses the cross-domain recognition problem, namely how to utilize the labeled data in a source

Manuscript received December 13, 2015; revised February 20, 2016; accepted March 1, 2016. Date of publication March 21, 2016; date of current version March 15, 2017. This work was supported in part by the Air Force Office of Scientific Research under Grant FA9550-11-1-0327, in part by the National Science Foundation under Grant IIS-1017199, in part by the U.S. Army Research Laboratory under Grant W911NF-10-2-0060 (U.S. Defense Advanced Research Projects Agency Minds Eye Program), in part by the Research Grants of University of Macau under Grant MYRG2015-00049-FST, Grant MYRG2015-00050-FST, and Grant RDG009/FST-TYY/2012, in part by the Science and Technology Development Fund of Macau under Grant 100-2012-A3 and Grant 026-2013-A, in part by Macau-China join Project under Grant 008-2014-AMJ, and in part by the National Natural Science Foundation of China under Grant 61273244. A preliminary version of this paper has appeared in [1]. This paper was recommended by Associate Editor D. Tao.

Y. Lin, Y. Zhou, and S. Wang are with the Department of Computer Science and Engineering, University of South Carolina, Columbia, SC 29208 USA (e-mail: ywlin.cq@gmail.com; zhou42@email.sc.edu; songwang@cec.sc.edu).

J. Chen is with Chongqing University, Chongqing 400030, China (e-mail: chenjingmc@gmail.com).

Y. Cao is with IBM Almaden Research Center, San Jose, CA 95120 USA (e-mail: caoy@us.ibm.com). He participated this work while he was doing Ph.D. degree in the University of South Carolina, Columbia, SC, USA. His contact information only refers to affiliation rather than any intellectual property.

L. Zhang is with the University of Houston, Houston, TX 77004 USA (e-mail: lzhang34@uh.edu).

Y. Y. Tang is with the University of Macau, Macau, China (e-mail: yytang@umac.mo).

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/TCYB.2016.2538199

Fig. 1. Sample images from four different domains: Amazon, Caltech, DSLR, and Webcam.

domain to classify/recognize the unlabeled data in a target domain.

To achieve cross-domain recognition, a number of domain adaption (DA) methods have been developed to adapt the classifier from one domain to another, which can be considered as a specific type of transfer learning [3]–[7]. The subspace-based DA methods have been found to be very effective to handle cross-domain problem [8]–[14]. They either constructed a set of intermediate subspaces for modeling the shifts between domains [8], [9], [11], [13], or generated a domain-invariant subspace in which the data from the source and target domains can represent each other well [10], [12], [14]. All these methods mentioned above utilize the data from a domain all together to generate a single subspace for the domain. In practice, however, the intrinsic feature shift of each class may not be exactly the same. The existing methods can obtain a global domain shift, but ignore the individual class difference across domains.

To circumvent the limitation of the global domain shift, we adopt a natural and widely used assumption that the data samples from the same class should lay on an intrinsic low-dimensional subspace, even if they come from different domains [15]. This assumption is not only held on many computer vision tasks, such as face recognition under varying illumination [16] and handwritten digit recognition [17] but also used as a human cognitive mechanism for visual object recognition [18]. Note that this assumption does not mean that the target data samples exactly lay on the intrinsic low-dimensional subspace of the source samples, since different domains show subspace shift [11]. Fig. 3 gives an toy illustration of a joint subspace covering the source domain and target domain for a specific class. The source and target subspaces have the overlap bases which implicitly represents the intrinsic characteristics of the considered class. They have their own exclusive bases because of the domain shift, such as the varying illumination or the change of the view perspective.

We also give an example to show the bases of subspaces using the sparse-subspace representation [15] of the data. Given N data samples $X = \{x_i\}_{i=1}^{N}$, each data sample can be reconstructed by a sparse linear combination of other samples based on the self-expressiveness property of the data. That is

$$\min \|C\|_0 \quad \text{s.t. } X = XC, \ \operatorname{diag}(C) = 0 \tag{1}$$

where c_{ij} is the coefficient of the sparse representation for reconstructing the x_i by x_j. Therefore, the average value of ith row of C, i.e., $R_i = (1/N)\sum_j \|c_{ij}\|$, denotes the importance of the ith data sample used for reconstructing other data samples. We treat the ith data sample as one basis of the intrinsic low-dimensional subspace of X, if $R_i \geq \epsilon$.

We consider the object "backpack" from Amazon dataset as the source domain S (92 data samples) and the same object from Caltech as the target domain T (151 data samples). Using the sparse-subspace representation of the union dataset $J = S \cup T$, we can obtain the bases of the joint subspace. We treat the ith data sample in J as the overlap bases if they are important for reconstructing the data samples both from S and T, i.e., $(1/92)\sum_{j=1}^{92} \|c_{ij}\| \geq \epsilon$ and $(1/151)\sum_{j=93}^{243} \|c_{ij}\| \geq \epsilon$, since the ith data sample belongs to both domains. Fig. 4(a) and (b) shows the overlap bases of the joint subspace in S and T, respectively. Fig. 4(c) and (d) shows the bases of the source and target intrinsic subspaces, respectively. The red bases are the same as the overlap ones, therefore, the blue ones are the exclusive bases for their own subspaces.

Based on the above assumption, we propose a new method that solves the cross-domain recognition by finding the joint subspaces of the source and target domains. Specifically, given labeled samples in the source domain, we construct an intrinsic subspace for each of the classes. Then we construct subspaces in the target domain, called anchor subspaces, by collecting unlabeled samples that are close to each other and high-likely belong to the same class. The corresponding class label is then assigned by minimizing a cost function which reflects the overlap and topological structure consistency between subspaces across the source and target domains, and within the anchor subspaces, respectively. We further combine the anchor subspaces to corresponding source subspaces to construct the joint subspaces for each class. Subsequently, an support vector machine (SVM) classifier is trained by using the samples in the joint subspace and applied to the unlabeled data in the target domain for classification.

The contributions of this paper are as follows.

1) By assuming that the data samples from one specific class, even though they come from different domains, should lay on an intrinsic low-dimensional subspace, we generate one joint subspace for each class independently. Each joint subspace carries the information not only about the intrinsic characteristics of the corresponding class, but also about the specificity for each domain.
2) To construct the joint subspaces, we first generate anchor subspaces in the target domain, assign labels to them, and combine these anchor subspaces to the corresponding source subspaces.
3) We propose a cost function that implicitly maximizes the overlap between the source subspace and the target subspace for each class as well as maintaining the

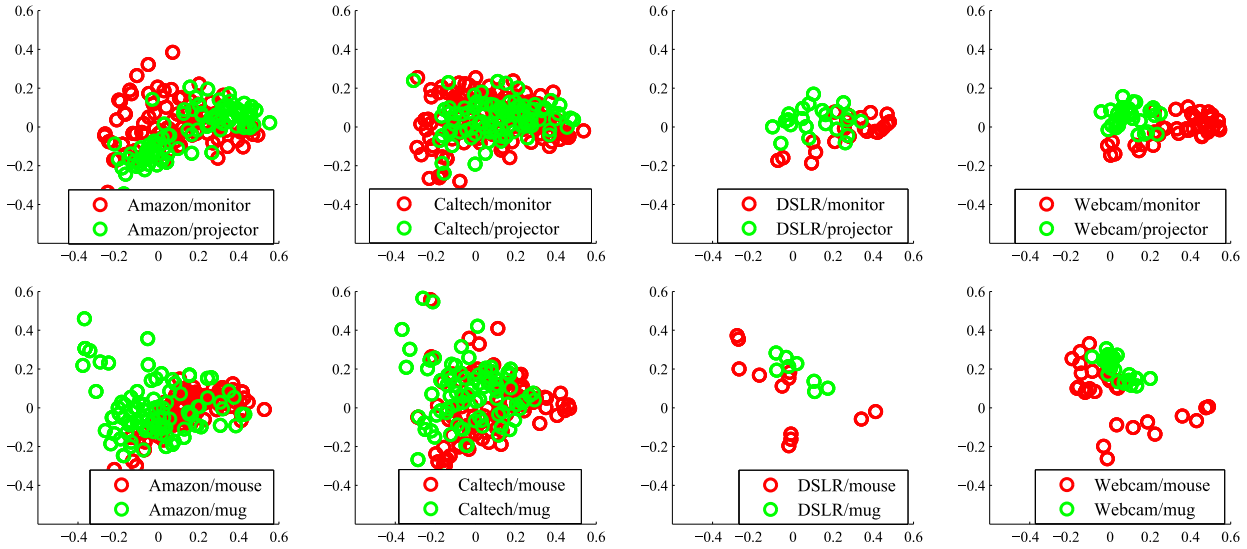


Fig. 2. Illustrations of sample distributions of different domains in feature space. The 2-D plots are the first two feature dimensions reduced from original 800-D SURF feature space (described in Section IV-D), using PCA. The first row are the distributions of monitor and projector in four domains. The second row are the distributions of mouse and mug in four domains. Best viewed in color.

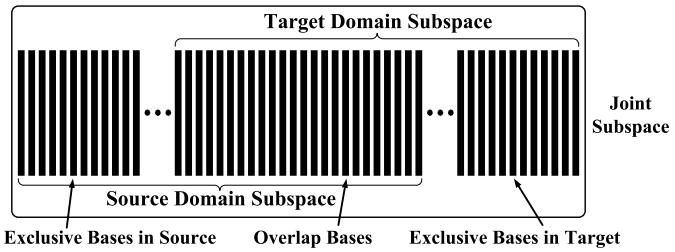


Fig. 3. Illustration of a joint subspace between the source and target domains for a specific class. This joint subspace consists of overlap bases between domains, which represent the intrinsic characteristics of this class implicitly, and exclusive bases of different domains, which represent the exclusive characteristics for each domain.

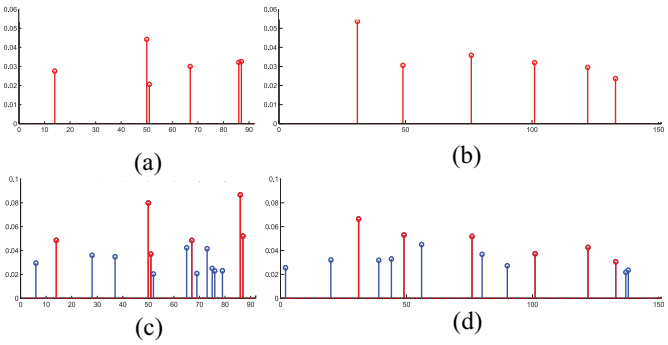


Fig. 4. (a) and (b) Overlap bases of the joint subspace in the source and target domains, respectively. (c) and (d) Bases of the source and target domain, respectively. The red bases are the same as the overlap bases, while the blue bases are the exclusive bases of their own subspaces. Best viewed in color.

topological structure in the target domain. We use the principal angles as the subspace distances in the cost function instead of the data-sample distances that were usually used in previous methods.

Note that, in this paper, when we say the subspace of a set of data samples, we do not explicitly get the bases for its intrinsic low-dimensional subspace. Instead, we use the original data samples to represent the intrinsic subspace implicitly. There are two reasons that we directly use the original data.

1) There is no need to have the bases since we do not project the original data to a new space. We only implicitly use the bases of the subspace when we calculate the distance between subspaces in Section IV-B. Using the original data avoids the additional computational cost, such as calculating the bases and projection.
2) All the data samples used either for training or testing should be in the same space. Therefore, we do not project them into different subspaces.

The remainder of this paper is organized as follows. We elaborate related works in Section II. Section III describes the proposed method. Quantitative experimental results are demonstrated in Section IV. Section V concludes this paper.

II. Related Work

For the cross-domain recognition problem, domain adaptation is the most closely related work which is known as a type of fundamental methods in machine learning and computer vision. Here, we give a brief review of this topic. Please refer to [19] for a comprehensive survey.

The traditional DA algorithms can be categorized into two types, i.e., (semi-)supervised domain adaptation and unsupervised domain adaptation, based on the availability of labeled data from the target domain. Both semisupervised and supervised DAs assume that the labeled data are available in the target domain. There are relatively large number of labeled data in the target domain for the supervised DA. For the semisupervised DA, there are only a very few number of labeled data available, motivated by the standard semisupervised learning [20]–[24]. Daumé [25] proposed to map the data from both source and target domains to a high-dimensional feature space, and trained classifiers in this new feature space. Saenko *et al*. [26] proposed a metric learning approach that can adapt labeled data of few classes from the target domain to the

unlabeled target classes. Kumar *et al*. [27] proposed a coregularization model that augments the feature space to jointly model the source and target domains. Chen *et al*. [28] proposed a cotraining-based domain adaptation method. They first train an initial category model on samples from the source domain and then use it for labeling samples from the target domain. The category model is then updating using the newly labeled target samples through cotraining. Pan *et al.* [29] analyzed the transfer component that maps both domains on kernel space to preserve some properties of domain-specific data distributions. Duan *et al*. [30] proposed an SVM-based method which minimized the mismatches between source and target domains, using both labeled and unlabeled data. Shekhar *et al*. [12] proposed to learn a single dictionary to represent both source and target domains. The work in [31] proposed to use a linear transformation to map features from the target domain to the source domain and generate the classification model trained on the source domain and target samples based on feature transformation. Motivated by the recent success of deep learning, some hierarchical domain adaptation methods are also proposed [32], which need large scale data to (pre-)train the deep neural network model. Xiao and Guo [33] proposed a semisupervised kernel matching-based domain adaptation method that learns a prediction function on the source domain while mapping the target samples to similar source samples by matching the target kernel matrix to the source kernel matrix.

Unsupervised DA, on the other hand, does not use any label information in the target domain, which is also considered as more challenging and more useful in real-world applications. Gopalan *et al*. [8], [34] constructed a set of intermediate subspaces along the geodesic path that links the source and target domains on the Grassmann manifold. Gong *et al*. [9] proposed a geodesic flow kernel (GFK) to model shift between the source and target domains. In [11], a new method was proposed to construct the subspaces by gradually reducing the reconstruction error of the target data instead of using the manifold walking strategies. Rather than using the geodesic path in [8] and [34], Caseiro *et al.* [35] proposed to compute the spline curve along with the Grassmann manifold. Jhuo *et al*. [10] learned a transformation so that the source samples can be represented by target samples in a low-rank way. Fernando *et al*. [36] proposed to learn a mapping function which aligns the sample representations from the source and target domains. Tommasi and Caputo [37] proposed a naive Bayes nearest neighbor-based domain adaptation algorithm that iteratively learns a class metric while inducing a large margin separation among classes for each sample. Baktashmotlagh *et al.* [38] proposed to use the Riemannian metric as a measure of distance between the distributions of source and target domains. Long *et al*. [14] proposed to learn a domain invariant representation by jointly performing the feature matching and instance weighting. Cui *et al.* [13] treated samples from each domain as one sample (i.e., covariance matrices) on a Riemannian manifold, and then interpolate some intermediate points along the geodesic path, which are used to bridge the two domains.

The algorithm proposed in this paper is an unsupervised cross-domain recognition method. It is different from the traditional DA methods by constructing an intrinsic low-dimensional joint subspace for each class independently. This avoids the global domain shift limitation and captures individual domain variations for each class.

III. Proposed Model

A. Problem Setting

Suppose there are two sets of data samples, one from source domain $\mathcal{S}$, denoted as $\{x_i^{\mathcal{S}}\}_{i=1}^{N_S} \in R^{d\times N_S}$, and the other one from target domain $\mathcal{T}$, denoted as $\{x_i^{\mathcal{T}}\}_{i=1}^{N_T} \in R^{d\times N_T}$, where d is the data dimension, N_S and N_T denote the number of data samples in the source and target domains, respectively. The labels of all data samples in the source domain, denoted as $Y^{\mathcal{S}} = \{y_i^{\mathcal{S}}\}_{i=1}^{N_S} \in R^{C\times N_S}$, are known, where C is the number of classes, $y_i^{\mathcal{S}} \in \{0, 1\}^C$ is a C bit binary code of the ith data sample in source domain. If this data sample belongs to class j, the jth bit of $y_i^{\mathcal{S}}$ is 1 and all the other bits are 0. Our aim is to estimate $Y^{\mathcal{T}} \in R^{C\times N_T}$, the labels of all the data samples in the target domain.

B. Overview

The proposed algorithm aims to construct a set of joint subspaces $\{M_i^{\mathcal{S}\mathcal{S}}\}_{i=1}^{C}$, which cover the source and target domains, one for each of the C classes, and then train the classifiers on the data samples belonging to these joint subspaces. As shown in Fig. 3, since a joint subspace is constituted by a source subspace and a target subspace, we need to construct the source and target subspaces first. Hence, the proposed algorithm consists of five steps.

1) Constructing Subspaces in Source Domain: We construct a set of subspaces $\{M_i^{\mathcal{S}}\}_{i=1}^{C}$, one for each class in the source domain. As mentioned above, we do not care about how to get a set of bases to represent a subspace. We implicitly construct the intrinsic low-dimensional subspace for each class by using original data samples, i.e., take all the data that belong to this class. Hence, each source subspace $M_i^{\mathcal{S}} = \{x_j^{\mathcal{S}}|_{x_j^{\mathcal{S}}\in C_i}\}$, where C_i denotes the ith class, illustrated in Fig. 5(a).

2) Constructing Anchor Subspaces in Target Domain: To estimate the target subspace for each class, we construct a number of anchor subspaces in the target domain, denoted as $\{M_i^{\mathcal{T}}\}_{i=1}^{K}$, as illustrated in Fig. 5(b). These anchor subspaces are expected to: 1) carry the information of target exclusive characteristics and 2) be compact. The data samples in the target domain naturally satisfy the first expectation. To satisfy the second expectation, the data samples in one anchor subspace should be from the same class, since the subspace constructed by samples from different class is usually less compact than the one constructed by samples from the same class. Thus, the basic idea is to ensure that each anchor subspace only contains data samples from a single class such that it can be combined to a source subspace for constructing the joint subspace compactly. Since the data in the target domain are unlabeled, we construct the anchor subspaces by grouping target data samples with high similarities. This is motivated by the locality principle—a data sample usually lies in close proximity to a small number of samples from the same class [39].

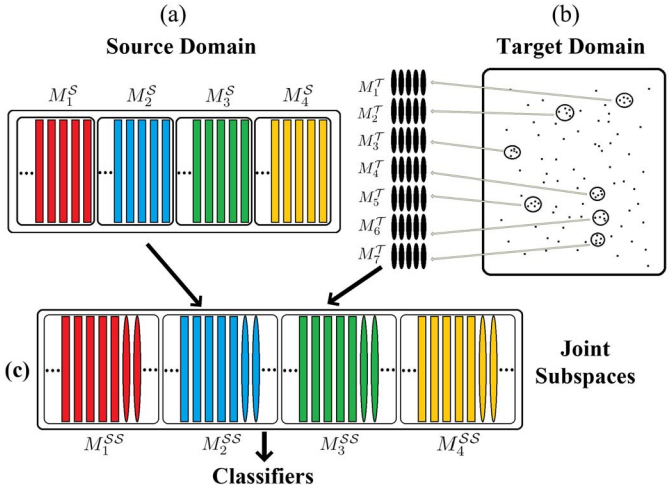


Fig. 5. Overview of the proposed model. (a) Subspaces for each class in source domain. The bars with the same color denote the data samples of one class. (b) Anchor subspaces construction. The points in target domain are the data samples. Data samples in each circle denote a core subgroup and they construct an anchor subspace as one row of black ellipses. (c) Joint subspaces construction. The ellipses denote the data samples from anchor subspaces (target domain). Best viewed in color.

3) Labeling the Anchor Subspaces: Since the joint subspaces are constructed independently for each class, we assign a label for each anchor subspace. For this purpose, we propose a cost function that reflects: 1) the cross-domain distance between the anchor subspace and the corresponding source subspaces and 2) the within-domain topological relation of the anchor subspaces in the target domain. Shorter cross-domain distance actually reflects the desirability of more overlap bases in constructing a joint subspace. In other words, the minimization of the proposed cost function implicitly reflects the maximization of the overlap between the source and target subspaces.

4) Constructing Joint Subspaces: We construct joint subspace $M_i^{\mathcal{SS}} = \{M_i^{\mathcal{S}}, M_j^{\mathcal{T}}|_{M_j^{\mathcal{T}} \in C_i}\}$, where C_i denotes the ith class, as illustrated in Fig. 5(c). As mentioned before, we simply take all the original data samples from all the involved subspaces to implicitly construct the joint subspaces.

5) Training Classifiers on the Joint Subspaces: We train one-versus-rest linear SVM classifiers for each class using the labeled data samples in the joint subspace. Specifically, for a class i, a linear SVM classifier is trained by using the labeled data samples belonging to $M_i^{\mathcal{SS}}$, i.e., the source data samples with the label of i and the target data samples assigned to the label i. Then we apply the linear SVM classifiers to the unlabeled data in the target domain for classification.

C. Anchor Subspaces Obtained in Target Domain

We construct each anchor subspace by selecting one target data sample and combining it with its nearest neighbors. This way, the obtained compact group of data samples are likely to be from the same class [39]. Specifically, we first apply the K-means algorithm to cluster all the target data into a large number of Z groups. We set $Z = (N_T/\gamma)$, where γ is the desired average group size. In each group of data, we find a compact core subgroup consisting of a small number of N samples, e.g., $N = 5$, which are taken for constructing an anchor subspace. In this paper, the core subgroup for the group L is constructed by the following two steps.

1) Estimate the center of the core subgroup by finding the data sample x^* to

$$\min_{x \in L} \sum_{x' \in \mathcal{N}_{(N-1)}(x)} \|x - x'\|_2 \tag{2}$$

where $\mathcal{N}_{(N-1)}(x)$ denotes the $N-1$ nearest neighbors of x in L.

2) Take $x^* \cup \mathcal{N}_{(N-1)}(x^*)$ as the core subgroup for constructing an anchor subspace.

For the groups that contain less than N data samples, we do not construct an anchor subspace.

D. Labeling Each Anchor Subspace

Note that, we have constructed C subspaces in the source domain, one for each class, denoted as $\{M_i^{\mathcal{S}}\}_{i=1}^{C}$. Their corresponding labels, denoted as $Y = \{y_i\}_{i=1}^{C} \in R^{C \times C}$, which is an identity matrix, i.e., the ith bit of y_i is 1 and all the other bits of y_i are 0. In this section, we developed a new strategy to assign class labels $Y' = \{y_i'\}_{i=1}^{K} \in R^{C \times K}$ for the K anchor subspaces $\{M_i^{\mathcal{T}}\}_{i=1}^{K}$ constructed in target domain. This strategy includes two main components: 1) the similarity between subspaces and 2) the cost function for subspace label assignment.

1) Distance Between Subspaces: To calculate the distance between two subspaces, principal angles are usually used [9], [40]. Principal angles between subspaces, which particularly defined between two orthonormal subspaces, serve as a classic tool in many areas of in computer science, such as computer vision and machine learning.

We follow the definition in [40], given two orthonormal matrices M_1 and M_2, the principal angles $0 \leq \theta_1 \leq \cdots \theta_m \leq \pi/2$ between two subspaces span(M_1) and span(M_2) is defined by

$$\begin{aligned} \cos\theta_k &= \max_{u_k \in \text{span}(M_1)} \max_{v_k \in \text{span}(M_2)} u_k^{\mathrm{T}} v_k \\ \text{s.t. } & u_k^{\mathrm{T}} u_k = 1, v_k^{\mathrm{T}} v_k = 1 \\ & u_k^{\mathrm{T}} u_i = 0, v_k^{\mathrm{T}} v_i = 0, (i = 1, \ldots, k-1). \end{aligned} \tag{3}$$

The principal angles are related to the geodesic distance between M_1 and M_2 as $\sum_i \theta_i^2$ [40].

In practice, the principal angles are usually computed from the singular value decomposition (SVD) of $M_1^{\mathrm{T}} M_2$, i.e., $M_1^{\mathrm{T}} M_2 = U(\cos\Theta)V^{\mathrm{T}}$, where $U = [u_1, \ldots, u_m]$, $V = [v_1, \ldots, v_n]$, and $\cos\Theta$ is the diagonal matrix $\text{diag}(\cos\theta_1 \cdots \cos\theta_{\min(m,n)})$.

In this paper, as shown in Section III-D2, we need both cross-domain subspace distance and within-domain subspace distance to define the cost function for anchor subspaces labeling. For the cross-domain subspace distance, e.g., between subspace $M_i^{\mathcal{S}}$ with s_i data samples in the source domain and $M_j^{\mathcal{T}}$ with t_j data samples in the target domain, we first orthogonalize both of them to obtain $\mathcal{M}_i^{\mathcal{S}}$ and $\mathcal{M}_j^{\mathcal{T}}$, and then calculate

the distance as

$$D\left(M_i^{\mathcal{S}}, M_j^{\mathcal{T}}\right) \triangleq \sum_{i=1}^{\min(s_i,t_j)} \sin\theta_i \tag{4}$$

where θ_i come from the SVD of $(\mathcal{M}_i^{\mathcal{S}})^{\mathrm{T}}\mathcal{M}_j^{\mathcal{T}}$ that $(\mathcal{M}_i^{\mathcal{S}})^{\mathrm{T}}\mathcal{M}_j^{\mathcal{T}} = U(\cos\Theta)V^{\mathcal{T}}$.

Similarly, for the within-domain subspace distance, e.g., between two target (anchor) subspaces $M_i^{\mathcal{T}}$ and $M_j^{\mathcal{T}}$ that both of them have N data samples, we first orthogonalize both of them to get $\mathcal{M}_i^{\mathcal{T}}$ and $\mathcal{M}_j^{\mathcal{T}}$, and then calculate the distance as

$$D\left(M_i^{\mathcal{T}}, M_j^{\mathcal{T}}\right) \triangleq \sum_{i=1}^{N} \sin\theta_i' \tag{5}$$

where θ_i' come from the SVD of $(\mathcal{M}_i^{\mathcal{T}})^{\mathrm{T}}\mathcal{M}_j^{\mathcal{T}}$ that $(\mathcal{M}_i^{\mathcal{T}})^{\mathrm{T}}\mathcal{M}_j^{\mathcal{T}} = U(\cos\Theta)V^{\mathrm{T}}$.

The above defined subspace distance follows the assumption that the samples within the same class share the same subspace even though they are from different domains. Consequently, the distance between subspaces across source and target domains of a specific class tends to be smaller than that between different classes. We will show quantitative comparison in Section IV-B to demonstrate this advantage.

We further generate two affinity matrices, $C \times K$ matrix A^{ST} to reflect the distances between K anchor subspaces and C source subspaces, and $K \times K$ matrix A^{TT} to reflect the pairwise distances among K anchor subspaces. More specifically, we have $A^{ST}(i,j) = \exp(-(D(M_i^{\mathcal{S}}, M_j^{\mathcal{T}})/2\sigma^2))$ and $A^{TT}(i,j) = \exp(-(D(M_i^{\mathcal{T}}, M_j^{\mathcal{T}})/2\sigma^2))$.

2) Cost Function and Optimization: Two important issues are considered in assigning a label to each anchor subspace: 1) the distance between an anchor subspace and the same-label source subspace should be small and 2) the local topological structures in the target domain should be preserved [41], i.e., anchor subspaces with shorter distance are more preferable to be assigned to the same class. Considering these two issues, we propose the cost function as follows:

$$\mathcal{C}(Y') = \sum_{i=1}^{C}\sum_{j=1}^{K}\left\|y_i - y_j'\right\|^2 A_{ij}^{ST} + \rho\sum_{j=1}^{K}\sum_{j'=1}^{K}\left\|y_j - y_j'\right\|^2 A_{jj'}^{TT} \tag{6}$$

where A^{ST} and A^{TT} are the affinity matrices of interdomain subspace pairs and subspace pairs within target domain, respectively.

Adding a constant term $\sum_{i=1}^{C}\sum_{j=1}^{C}\|y_i - y_j\|^2 I_{ij}$ into $\mathcal{C}$ and splitting the first term into two parts, we get

$$\begin{aligned}\mathcal{C}(Y') = {} & \frac{1}{2}\sum_{i=1}^{C}\sum_{j=1}^{K}\left\|y_i - y_j'\right\|^2 A_{ij}^{ST} + \frac{1}{2}\sum_{j=1}^{C}\sum_{i=1}^{K}\left\|y_j - y_i'\right\|^2 A_{ij}^{ST} \\ & + \rho\sum_{j=1}^{K}\sum_{j'=1}^{K}\left\|y_i' - y_j'\right\|^2 A_{jj'}^{TT} + \sum_{i=1}^{C}\sum_{j=1}^{C}\left\|y_i - y_j\right\|^2 I_{ij}\end{aligned} \tag{7}$$

where I_{ij} is the ijth element of the identity matrix I.

Note that the first and second terms are equal. The cost function $\mathcal{C}$ can be further written as

$$\mathcal{C}(Y') = \sum_{i=1}^{C+K}\sum_{j=1}^{C+K}\|\mathcal{Y}_i - \mathcal{Y}_j\|^2\mathcal{A}_{ij},\ \text{s.t. } \mathcal{Y}^{\mathrm{T}}\mathbf{1} = \mathbf{1} \tag{8}$$

where $\mathcal{Y} = [Y, Y']$, $\mathcal{A} = \begin{bmatrix} I & \frac{1}{2}A^{ST} \\ \frac{1}{2}\left(A^{ST}\right)^{\mathrm{T}} & \rho A^{TT} \end{bmatrix}$. We also relax the constraint to this cost function by only requiring the sum of each column in $\mathcal{Y}$ to be 1.

By including the constraint term, the cost function can be written in a matrix form [42]

$$\mathcal{L}(\mathcal{Y}, \lambda) = Tr\left(\mathcal{Y}\Delta\mathcal{Y}^{\mathrm{T}}\right) + \lambda^{\mathrm{T}}\left(\mathcal{Y}^{\mathrm{T}}\mathbf{1} - \mathbf{1}\right) + \frac{\mu}{2}\left\|\mathcal{Y}^{\mathrm{T}}\mathbf{1} - \mathbf{1}\right\|_2^2 \tag{9}$$

where $\Delta = \mathcal{D} - \mathcal{C}$ is the Laplacian matrix of $\mathcal{A}$. $\mathcal{D}$ is the degree matrix which is a diagonal matrix with $\mathcal{D}_{ii} = \sum_j \mathcal{A}_{ij}$. $\lambda \in \mathcal{R}^{C+K}$ is the Lagrange multiplier. To minimize the objective function $\mathcal{L}$, we separate it into two steps to update its two unknowns $\mathcal{Y}$ and λ alternately:

Step 1: Having λ fixed, optimize $\mathcal{Y}$ by computing the derivative of $\mathcal{L}$ with the respect to $\mathcal{Y}$ and setting it to be zero

$$\frac{\partial\mathcal{L}(\mathcal{Y}, \lambda)}{\partial\mathcal{Y}} = 0 \Rightarrow \mathcal{Y}\Delta + \mathbf{1}\lambda^{\mathrm{T}} + \mu\left(\mathbf{1}\mathbf{1}^{\mathrm{T}}\mathcal{Y} - \mathbf{1}\mathbf{1}^{\mathrm{T}}\right) = 0. \tag{10}$$

Note that $\mathcal{Y}$ contains two parts Y and Y', with Y is known. Thus, to solve it, as in [43], we first split the Laplacian matrix Δ into four blocks along the Cth row and column

$$\Delta = \begin{bmatrix} \Delta_{CC} & \Delta_{CK} \\ \Delta_{KC} & \Delta_{KK} \end{bmatrix}.$$

Similarly, we separate λ into two parts

$$\lambda_1 = [\lambda_1, \lambda_2, \ldots, \lambda_C]^{\mathrm{T}} \quad \text{and} \quad \lambda_2 = [\lambda_{C+1}, \lambda_{C+2}, \ldots, \lambda_{C+K}]^{\mathrm{T}}.$$

Then Y' can be updated by solving the following equation:

$$Y'^{(k+1)}\Delta_{KK} + \mu\mathbf{1}\mathbf{1}^{\mathrm{T}}Y'^{(k+1)} = \mu\mathbf{1}\mathbf{1}^{\mathrm{T}} - Y\Delta_{CK} - \mathbf{1}\lambda_2^{(k)\mathrm{T}}. \tag{11}$$

The solution is given by the Lyapunov equation [44]. With this solution, $\mathcal{Y}^{(k+1)}$ can be achieved by putting Y and $Y'^{(k+1)}$ together as $\mathcal{Y}^{(k+1)} = [Y, Y'^{(k+1)}]$.

Step 2: Having $\mathcal{Y}$ fixed, perform a gradient ascending update with the step of μ on Lagrange multipliers as

$$\lambda^{(k+1)} = \lambda^{(k)} + \mu\left(\mathcal{Y}^{(k+1)^{\mathrm{T}}}\mathbf{1} - \mathbf{1}\right). \tag{12}$$

To initialize this optimization process, we simply set $\lambda^{(0)}$, $Y'^{(0)}$ to be zero, and set the maximal number of iteration maxIter to be 1000. This whole algorithm is summarized in Algorithm 1.

Since we only require the sum of each column in $\mathcal{Y}$ to be 1, after we get Y', we set the bit with the maximal value in each column to 1 and all the other bits to 0.

IV. Experimental Results

In this section, we first give the evaluation results on two components of the proposed algorithm, i.e., the subspace distance we used and the anchor subspaces labeling. Then we evaluate the proposed algorithm comprehensively on two widely used cross domain recognition datasets: 1) object

Algorithm 1 Labeling the Anchor Subspaces

Input: Affinity matrix $\mathcal{A}$, *maxIter*, labels Y for $M^{\mathcal{S}}$
Output: Labels Y' for $M^{\mathcal{T}}$.

Initialization: $\lambda^{(0)}$, $Y'^{(0)}$ to zero
1: **Do**
2: Update the $Y'^{(k+1)}$ by solving the following equation:

$$Y'^{(k+1)}\Delta_{KK} + \mu \mathbf{1}\mathbf{1}^{\mathrm{T}} Y'^{(k+1)} = \mu \mathbf{1}\mathbf{1}^{\mathrm{T}} - Y\Delta_{CK} - \mathbf{1}\lambda_2^{(k)\mathrm{T}}$$

3: Update $\mathcal{Y}^{(k+1)} = [Y, Y'^{(k+1)}]$.
4: Update $\lambda^{(k+1)} = \lambda^{(k)} + \mu\left(\mathcal{Y}^{(k+1)^{\mathrm{T}}}\mathbf{1} - \mathbf{1}\right)$.
5: Check the convergence.
6: **Until convergence or maxIter iterations reached.**

TABLE I
NUMBERS OF DATA SAMPLES IN EACH CLASS FROM FOUR DOMAINS

	$C1$	$C2$	$C3$	$C4$	$C5$	$C6$	$C7$	$C8$	$C9$	$C10$	Total
Amazon	92	82	94	99	100	100	99	100	94	98	958
Caltech	151	110	100	138	85	128	133	94	87	97	1123
DSLR	12	21	12	13	10	24	22	12	8	23	157
Webcam	29	21	31	27	27	30	43	30	27	30	295

recognition image dataset for computer vision tasks and 2) sentiment classification dataset for natural language processing tasks.

A. Dataset and Experimental Configurations

The first dataset that we evaluate on is an image dataset. The whole dataset has four subdatasets, which we use as four domains, with 2533 images from ten classes in total, as shown in Table I. The first three subdatasets were collected by [26], which include images from amazon.com (Amazon), collected with a digital SLR (DSLR) and a webcam (Webcam). The fourth domain is Caltech dataset (Caltech) [45]. For simplicity, hereafter, we use "A," "C," "D," and "W" to denote the "Amazon," "Caltech," "DSLR," and "Webcam" domains, respectively.

In the natural language processing task, customers' reviews on four different products (kitchen applications, DVDs, books, and electronics) are collected as four domains [46]. Each review consists of comment texts and a rating from 0 to 5. Reviews with rating higher than 3 are classified as positive samples, and the remaining reviews are classified as negative samples. In total, there are 1000 positive reviews and 1000 negative reviews in each domain. The goal of this task is to adapt the classifier training on one domain and use it for classifying data samples in the other domain.

We ran our algorithm 20 times for each object-recognition task and gave the average accuracy rate (%) and standard deviation (%). In all of our experiments, we consistently set the parameter γ to be 20 and N to be 5. We show that the performance of the proposed algorithm is not very sensitive to these two parameters in Section IV-D3.

TABLE II
DISTANCE MATRIX BETWEEN CLASSES ACROSS SOURCE AND TARGET DOMAINS

	$C1$	$C2$	$C3$	$C4$	$C5$	$C6$	$C7$	$C8$	$C9$	$C10$
$C1$	**0.86**	0.92	0.92	0.91	0.91	0.90	0.90	0.91	0.91	0.91
$C2$	0.92	**0.85**	0.92	0.90	0.94	0.92	0.93	0.92	0.93	0.94
$C3$	0.92	0.92	**0.85**	0.92	0.90	0.90	0.91	0.92	0.92	0.92
$C4$	0.91	0.90	0.92	**0.87**	0.92	0.90	0.90	0.91	0.91	0.91
$C5$	0.91	0.94	0.90	0.92	**0.86**	0.90	0.91	0.92	0.92	0.92
$C6$	0.90	0.92	0.90	0.90	0.90	**0.86**	0.88	0.90	0.92	0.90
$C7$	0.90	0.93	0.91	0.90	0.91	0.88	**0.87**	0.90	0.92	0.90
$C8$	0.91	0.92	0.92	0.91	0.92	0.90	0.90	**0.90**	0.92	0.92
$C9$	0.91	0.93	0.92	0.91	0.92	0.92	0.92	0.92	**0.91**	0.92
$C10$	0.91	0.94	0.92	0.91	0.92	0.90	0.90	0.92	0.92	**0.90**

TABLE III
RESULTS OF THE TARGET DATA SAMPLES LABELING

Model	Labeled #	Total #	Labeling Accuracy	Baseline (SVM)	Proposed
C→A	85	958	68.2	48.2	59.1
C→W	30	295	56.7	43.7	52.2
C→D	20	157	80.0	42.7	53.0
A→C	80	1123	51.3	37.2	47.6
A→W	30	295	68.2	34.6	42.2
A→D	20	157	85.0	36.3	47.9
W→C	75	1123	46.7	31.7	33.5
W→A	85	958	50.6	35.1	39.5
W→D	25	157	80.0	70.1	89.4
D→C	80	1123	46.3	32.6	34.5
D→A	95	958	50.0	34.2	37.9
D→W	35	295	97.1	80.0	89.3

B. Evaluation on the Distance we Used in Proposed Algorithm

The distance matrix in Table II is given to show that the subspace-based distance is suitable for our method. In this table, we use the data from object recognition dataset. Each column denotes the distance between a specific class $(C1, \ldots, C10)$ in source domain and a class $(C1, \ldots, C10)$ in target domain. Note that all the numbers are the average of all 12 pairs of the source and target domain (described in Section IV-D1). We can see that the distances, across two domains, between the same class are relatively smaller than those between different classes. Therefore, the results also demonstrate the assumption that the samples with the same class share the same subspace even though they are from different domains, i.e., the distance between subspaces across source and target domains of a specific class tends to be smaller than that between different classes.

C. Evaluation on Anchor Subspaces Labeling

We also evaluate the effectiveness of the proposed anchor subspaces labeling algorithm introduced in Section III-D. In Table III, "Labeled #" denotes the number of target data samples which are used for generating the anchor subspaces and

TABLE IV
RESULTS OF SINGLE SOURCE AND TARGET DOMAIN ON THE OBJECT RECOGNITION DATASET. "–" DENOTES THAT THERE IS NO RESULT REPORTED BEFORE

Model	C→A	C→W	C→D	A→C	A→W	A→D
Baseline(SVM)	48.2	43.7	42.7	37.2	34.6	36.3
K-SVD [47]	20.5±0.8	-	19.8±1.0	20.2±0.9	16.9±1.0	-
SGF [8]	48.9±0.7	42.9±0.8	44.0±1.0	40.0±0.3	35.0±0.7	34.9±0.6
GFK [9]	40.4±0.7	35.8±1.0	41.1±1.3	37.9±0.4	35.7±0.9	35.2±0.9
Metric [26]	33.7±0.8	-	35.0±1.1	36.0±1.0	21.7±0.5	-
ITL [48]	49.2±0.6	43.3±0.7	44.4±1.2	38.5±0.4	40.0±1.3	39.6±0.6
SI [11]	45.4±0.3	37.0±5.1	42.3±0.4	40.4±0.5	37.9±0.9	32.1±4.5
SA(SVM) [36]	46.1	38.9	39.4	39.9	39.6	38.8
DASC [13]	49.8±0.4	45.4±0.9	48.5±0.8	39.1±0.3	37.7±0.7	39.3±0.8
TJM [14]	46.8	39.0	44.6	39.5	42.0	45.2
CJS (ours)	**59.1±1.2**	**52.2±2.6**	**53.0±3.5**	**47.6±1.1**	**42.2±2.9**	**47.9±2.2**
Model	W→C	W→A	W→D	D→C	D→A	D→W
Baseline(SVM)	31.7	35.1	70.1	32.6	34.2	80.0
K-SVD [47]	13.2±0.6	14.2±0.7	-	-	14.3±0.3	46.8±0.8
SGF [8]	32.3±0.4	35.1±0.5	72.9±0.7	34.9±0.3	34.7±0.4	82.0±0.6
GFK [9]	29.3±0.4	35.5±0.7	71.2±0.9	32.7±0.4	36.1±0.4	79.1±0.7
Metric [26]	32.3±0.8	38.6±0.8	-	-	30.3±0.8	55.6±0.7
ITL [48]	32.2±0.3	35.2±0.3	75.6±0.8	34.7±0.3	39.6±0.4	83.6±0.5
SI [11]	**36.3±0.3**	38.3±0.3	79.5±2.0	**35.5±1.8**	39.1±0.5	86.2±1.0
SA(SVM) [36]	31.8	39.3	77.9	35.0	**42.0**	82.3
DASC [13]	33.3±0.3	36.3±0.4	71.2±0.9	32.7±0.4	36.5±0.3	88.3±0.4
TJM [14]	30.2	30.0	89.2	31.4	32.8	85.4
CJS (ours)	33.5±1.6	**39.5±1.3**	**89.4±1.8**	34.5±1.9	37.9±1.6	**89.3±1.7**

then assigned labels, and "Labeling Accuracy" denotes the labeling accuracy (%). "Baseline (SVM)" denotes the accuracy by directly utilizing the SVM classifier trained from the source domain to the target domain. As shown in Table III, we can see that the number of the target data samples used for labeling (Labeled # column) are only about 10% out of the total number of the target data samples (the "Total #" column), as well as the accuracies of the labeling are not perfect (the "Accuracy" column). This shows that the proposed algorithm benefits from involving the information from the target domain by labeling the target data samples, even through the number of the labeled target data are relatively small and labeling accuracies are not perfect. Especially, the accuracy of proposed method is even better than the label assignment in the case of W → D. We argue that the labeled target data samples provide the information from the target domain with noise. SVM classifier takes the information from the target domain with noise-tolerant to improve the performance comparing with the classifier without using target information.

D. Cross-Domain Recognition on Object Dataset

Following the way of feature extraction for each image in [11], we first use an SURF [49] detector to extract points of interest from each image. We then randomly select a subset of the points of interest and quantize their descriptors to 800 visual words using the *K*-means clustering. Finally, we construct a 800-D feature vector for each image using the bag-of-visual-words technique.

1) Single Source Domain and Single Target Domain: We report the results on all 12 possible pairs of source- and target-domain combinations. We compare our algorithm with nine other methods, including K-singular value decomposition [47], sampling geodesic flow (SGF) [8], GFK [9], Metric [26], information-theoretical learning [48], subspace interpolation [11], subspace alignment [36], domain adaptation by shifting covariance [13], and transfer joint matching [14]. Also, we give the accuracy rate by directly utilizing the SVM trained from source domain to the target domain. Their results in Table IV are obtained from previous papers, mostly by the original authors. It can be seen that our algorithm performs best in 9 out 12 domain pairs. In particular, in four domain pairs our algorithm significantly outperforms (by more than 5%) all the comparison methods, i.e., C→A, C→D, A→C, and C→W. Our algorithm shows a comparable performance with the best performed method in the other three domain pairs. Note that the "metric" method [26] is a semisupervised method.

2) Multiple Source/Target Domains: We then evaluate the performance when there are multiple source/target domains. To get the fair comparison with other method, we also directly get the results from the previous literature. Thus, we only conduct the multiple source/target domains cross-domain recognition on six possible different source- and target-domain combinations, followed [34], among which three combinations include two source domains and one target domain, and the other three combinations include one source domain and two target domains. When there are multiple source/target domains, we simply merge the data samples in all the source/target domains as a single domain.

TABLE V
RESULTS OF MULTISOURCE DOMAIN ADAPTATION ON THE OBJECT RECOGNITION DATASET. NOTE THAT ALL THE COMPARISON METHODS ARE SEMISUPERVISED DOMAIN ADAPTATION METHOD EXCEPT THE US MARK ONE

Model	D+A→W	A+W→D	W+D→A
SGF [8] (US)	31.0±1.6	25.0±0.4	15.0±0.4
SGF [8]	52.0±2.5	39.0±1.1	28.0±0.8
RDALR [10]	36.9±1.1	31.2±1.3	20.9±0.9
FDDL [50]	41.0±2.4	38.4±3.4	19.0±1.2
SDDL [12]	57.8±2.4	56.7±2.3	24.1±1.6
HMP [51]	47.2±1.9	51.3±1.4	37.3±1.4
Gopalan et al. [34]	51.3	36.1	35.8
CJS (ours)	**73.2±2.5**	**81.3±1.3**	**41.1±1.1**

When there are multiple source domains, we report the results of four comparison methods, including SGF [8], robust domain adaptation with low-rank (RDALR) [10], Fisher discrimination dictionary learning (FDDL) [50], shared domain-adapted dictionary learning (SDDL) [12], hierarchical matching pursuit [51], and the model in [34], as shown in Table V. For SGF [8], we report its performance under both unsupervised and semisupervised settings. Note that RDALR, FDDL, and SDDL are all semisupervised methods, while our proposed method is unsupervised. It is clearly to see that the proposed method outperforms all the comparison methods significantly.

In principle, using multiple source domains should provide more information for each class, which should result in higher performance than using a single source domain. For example, for the domain combination of "W + D→A" (41.3%), it shows a marginal performance improvement over the single-source domain cases, "D→A" (38.5%) and "W→A" (39.1%). In practice, however, we do not always achieve higher performance when using multiple source domains. For example, comparing the results from Tables IV and V, the performance of "D + A→W" (73.2%) lies in between the performances of two single-source domain cases "D→W" (89.5%) and "A→W" (42.4%). This is an interesting problem to be studied in our future work.

When there are multiple target domains, we only find two comparison methods, SGF [8] and the method from [34]. Both unsupervised and semisupervised settings were developed for SGF, and the model from [34] is the semisupervised-based method. We take the performance from [8] and [34], and include that in Table VI. It can be seen that the proposed algorithm performs better than both settings of SGF and two out of three cases of the model from [34].

In principle, using multiple target domains does provide more information, since the labels are only available in the source domain. Accordingly, the performance of using multiple target domains should be the weighted (based on numbers of data samples in each involved target domain) average of the performance of using each target domain separately. The results in Tables IV and VI are largely aligned with this expectation.

3) Performance Under Different Parameter Settings: There are two main parameters in the proposed algorithm: 1) the desired average size of each group constructed in the target domain using the K-means algorithm, i.e., γ and 2) the number of data samples in each anchor subspace, i.e., N. In order to investigate the sensitivity to different parameter settings, we tune each of the two parameters, respectively, and report the performance of each parameter setting. For each parameter setting, we report the accuracy rate by percentage, for eight combinations of single source- and target-domains, and six combinations of multiple source/target domains.

We take the value of γ in the range of 5–30 with the step length of 5, and the results are shown in Fig. 6(a). We can see that, the performance only varies in a small range for almost all the domain combinations, except for "D, A→W."

For N, we take its value in the range of 3 to 10 with the step length of 1 and the results are reported in Fig. 6(b). It is clear to see that the performance only varies in a small range

TABLE VI
RESULTS OF MULTITARGET DOMAIN ADAPTATION ON THE 2-D OBJECT RECOGNITION DATASET. "SS" AND "US" DENOTE THE SEMISUPERVISED AND UNSUPERVISED SETTING, RESPECTIVELY

Model	W→A+D	D→A+W	A→D+W
SGF [8] (US)	28.0±1.9	35.0±1.7	22.0±0.2
SGF [8] (SS)	42.0±2.8	46.0±2.3	32.0±0.9
Gopalan et al. [34] (SS)	44.0	**49.5**	30.0
CJS (ours)	**45.1±1.2**	48.4±2.2	**44.2±2.0**

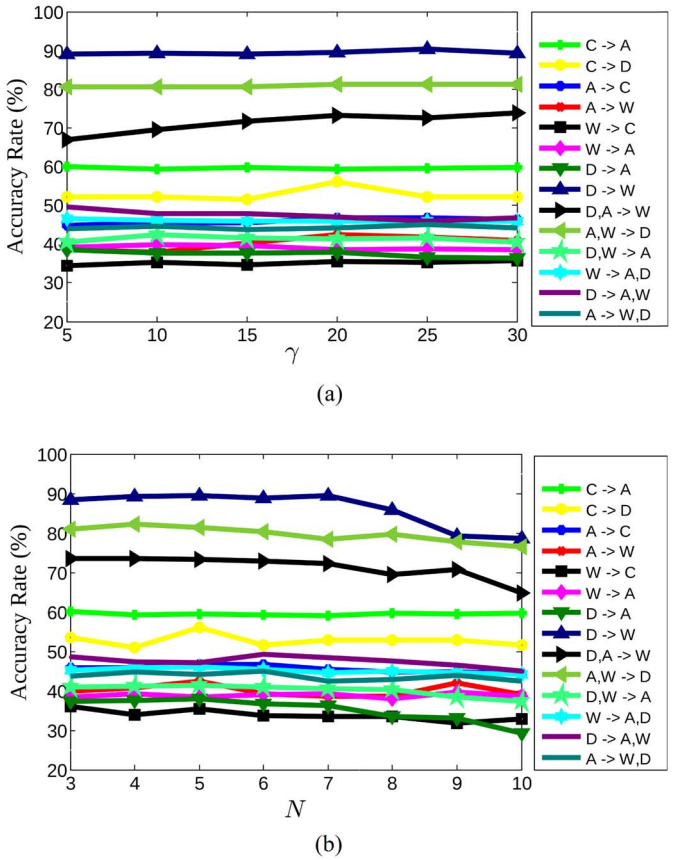


Fig. 6. Results when varying value for (a) γ, which determines the number of groups obtained by the K-means algorithm, and (b) N, which indicates the number of samples in each anchor subspace.

TABLE VII
DOMAIN ADAPTATION RESULTS ON THE SENTIMENT CLASSIFICATION. K: KITCHEN, D: DVD, B: BOOKS, AND E: ELECTRONICS

Model	K→D	D→B	B→E	E→K
TCA [29]	60.4	61.4	61.3	68.7
SGF [8]	67.9	68.6	66.9	75.1
FGK [9]	69.0	71.3	68.4	78.2
SCL [53]	72.8	76.2	75.0	82.9
KMM [54]	72.2	78.6	76.9	83.5
Metric [26]	70.6	72.0	72.2	77.1
Landmark [52]	75.1	**79.0**	78.5	83.4
CJS (ours)	**77.8**	77.0	**83.2**	**84.1**

for almost all the domain combinations, except for D→W and D, A→W when $N > 8$.

Therefore, we can conclude that the performance of the proposed algorithm is not very sensitive to the two parameters γ and N.

4) Running Time: All the running times are estimated on a laptop with Intel i7-2620M CPU and 4 GB RAM. We mainly implemented our algorithm by MATLAB 2014, with several core functions implemented by C++ for speeding up. There are five major steps in the proposed algorithm as introduced in Section III-B. The running time for the first and the fourth steps can be ignored. The second step requires the K-means algorithm, which takes an average time of 0.2 s (varying along with the number of target data samples). The third step, i.e., labeling the anchor subspaces, includes two substeps. The first one is to calculate the affinity matrices which reflect the distances between anchor subspaces and interdomain subspaces, respectively. Each distance calculation requires the orthogonalization and SVD. In total, it takes average time of 0.02 s for calculating these two matrices. The second substep is to label anchor subspaces based on those two affinity matrices by using Algorithm 1. This algorithm is iterative based and its time complexity is known as $O(T(C^6+L^6))$, where T is the maximal iteration number. Fortunately, the values of C and L are both small, i.e., $C = 10$ and $L < 20$ based on Table III. This algorithm takes an average time of 0.3 s. The most time consuming step in the proposed algorithm is the SVM classifier training, which takes an average time of 45 s. The running times for different domain pairs vary along with the number of training data samples, from the 3 s for D→W to 100 s for "C→A."

E. Cross-Domain Recognition on Sentiment Classification Dataset

Although the proposed algorithm is originally designed for vision tasks, it can be easily utilized for cross domain tasks in other areas. In this section, as an example, we compare the proposed algorithm with other seven methods in a domain adaptation task from the natural language processing area.

We follow the same experiment setup described in [52]. In each domain, 1600 reviews including 800 positive reviews and 800 negative reviews, are used as the training set, and the rest 400 reviews are used as the testing set. We extract unigram and bigram features on the comment texts, and the feature dimension is reduced to 400. Finally, each comment text is represented by a 400-D feature using the bag-of-words technique.

We conduct experiments on four pairs of source- and target-domain combinations. The same experiment has been also conducted in [52]. The performance is reported in Table VII. It is clear to see that overall the proposed algorithm outperforms other seven methods. From this, we can see that the proposed algorithm can also be used for DA problems in nonvision areas.

V. CONCLUSION

This paper introduces a new subspace-based domain adaptation algorithm. The joint subspace is independently constructed for each class, which covers both source and target domains. The joint subspace carries the information not only about the intrinsic characteristics of the considered class, but also about the specificity for each domain. Classifiers are trained on these joint subspaces. The proposed algorithm has been evaluated on two widely used datasets. Comparison results show that the proposed algorithm outperforms several existing methods on both datasets.

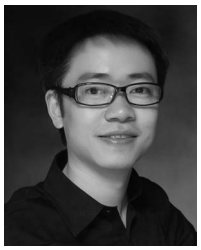
REFERENCES

[1] Y. Lin *et al.*, "Cross-domain recognition by identifying compact joint subspaces," in *Proc. Int. Conf. Image Process.*, Quebec City, QC, Canada, 2015, pp. 3461–3465.
[2] A. Torralba and A. A. Efros, "Unbiased look at dataset bias," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, 2011, pp. 1521–1528.
[3] S. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.
[4] Y. Luo, D. Tao, B. Geng, C. Xu, and S. J. Maybank, "Manifold regularized multitask learning for semi-supervised multilabel image classification," *IEEE Trans. Image Process.*, vol. 22, no. 2, pp. 523–536, Feb. 2013.
[5] Y. Luo, T. Liu, D. Tao, and C. Xu, "Decomposition-based transfer distance metric learning for image classification," *IEEE Trans. Image Process.*, vol. 23, no. 9, pp. 3789–3801, Sep. 2014.
[6] W. Liu, D. Tao, J. Cheng, and Y. Tang, "Multiview Hessian discriminative sparse coding for image annotation," *Comput. Vis. Image Understand.*, vol. 118, pp. 50–60, Jan. 2014.
[7] B. Li, X. Zhu, R. Li, and C. Zhang, "Rating knowledge sharing in cross-domain collaborative filtering," *IEEE Trans. Cybern.*, vol. 45, no. 5, pp. 1068–1082, May 2015.
[8] R. Gopalan, R. Li, and R. Chellappa, "Domain adaptation for object recognition: An unsupervised approach," in *Proc. IEEE Int. Conf. Comput. Vis.*, Barcelona, Spain, 2011, pp. 999–1006.
[9] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, 2012, pp. 2066–2073.
[10] 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.*, Providence, RI, USA, 2012, pp. 2168–2175.
[11] J. Ni, Q. Qiu, and R. Chellappa, "Subspace interpolation via dictionary learning for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Portland, OR, USA, 2013, pp. 692–699.
[12] S. Shekhar, V. M. Patel, H. V. Nguyen, and R. Chellappa, "Generalized domain-adaptive dictionaries," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Portland, OR, USA, 2013, pp. 361–368.
[13] Z. Cui *et al.*, "Flowing on Riemannian manifold: Domain adaptation by shifting covariance," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2264–2273, Dec. 2014.
[14] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer joint matching for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Columbus, OH, USA, 2014, pp. 1410–1417.

[15] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2765–2781, Nov. 2013.
[16] R. Basri and D. W. Jacobs, "Lambertian reflectance and linear subspaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 3, pp. 218–233, Feb. 2003.
[17] T. Hastie and P. Y. Simard, "Metrics and models for handwritten character recognition," *Stat. Sci.*, vol. 13, no. 1, pp. 54–65, 1998.
[18] J. J. DiCarlo, D. Zoccolan, and N. C. Rust, "How does the brain solve visual object recognition?" *Neuron*, vol. 73, no. 3, pp. 415–434, Feb. 2012.
[19] 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.
[20] D. Tao, X. Tang, X. Li, and X. Wu, "Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 7, pp. 1088–1099, Jul. 2006.
[21] D. Tao, X. Li, X. Wu, and S. J. Maybank, "General tensor discriminant analysis and Gabor features for gait recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 10, pp. 1700–1715, Oct. 2007.
[22] N. Wang, D. Tao, X. Gao, X. Li, and J. Li, "Transductive face sketch-photo synthesis," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 9, pp. 1364–1376, Sep. 2013.
[23] J. Yu, Y. Rui, and D. Tao, "Click prediction for Web image reranking using multimodal sparse coding," *IEEE Trans. Image Process.*, vol. 23, no. 5, pp. 2019–2032, May 2014.
[24] J. Yu, D. Tao, M. Wang, and Y. Rui, "Learning to rank using user clicks and visual features for image retrieval," *IEEE Trans. Cybern.*, vol. 45, no. 4, pp. 767–779, Apr. 2015.
[25] H. Daumé, III, "Frustratingly easy domain adaptation," in *Proc. Conf. Assoc. Comput. Linguist.*, 2007, pp. 256–263.
[26] 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.
[27] A. Kumar, A. Saha, and H. Daume, "Co-regularization based semi-supervised domain adaptation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 478–486.
[28] M. Chen, K. Q. Weinberger, and J. C. Blitzer, "Co-training for domain adaptation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 2456–2464.
[29] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Trans. Neural Netw.*, vol. 22, no. 2, pp. 199–210, Feb. 2011.
[30] 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.
[31] J. Donahue, J. Hoffman, E. Rodner, K. Saenko, and T. Darrell, "Semi-supervised domain adaptation with instance constraints," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Portland, OR, USA, 2013, pp. 668–675.
[32] J. Donahue *et al.*, "DeCAF: A deep convolutional activation feature for generic visual recognition," in *Proc. Int. Conf. Mach. Learn.*, Beijing, China, 2014, pp. 647–655.
[33] M. Xiao and Y. Guo, "Feature space independent semi-supervised domain adaptation via kernel matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 1, pp. 54–66, Jan. 2015.
[34] 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, Nov. 2014.
[35] R. Caseiro, J. F. Henriques, P. Martins, and J. Batista, "Beyond the shortest path: Unsupervised domain adaptation by sampling subspaces along the spline flow," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Boston, MA, USA, 2015, pp. 3846–3854.
[36] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars, "Unsupervised visual domain adaptation using subspace alignment," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sydney, NSW, Australia, 2013, pp. 2960–2967.
[37] T. Tommasi and B. Caputo, "Frustratingly easy NBNN domain adaptation," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sydney, NSW, Australia, 2013, pp. 897–904.
[38] M. Baktashmotlagh, M. T. Harandi, B. C. Lovell, and M. Salzmann, "Domain adaptation on the statistical manifold," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Columbus, OH, USA, 2014, pp. 2481–2488.
[39] V. Zografos, L. Ellisy, and R. Mester, "Discriminative subspace clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Portland, OR, USA, 2013, pp. 2107–2114.
[40] J. Hamm and D. D. Lee, "Grassmann discriminant analysis: A unifying view on subspace-based learning," in *Proc. Int. Conf. Mach. Learn.*, Helsinki, Finland, 2008, pp. 376–383.
[41] D. Zhai *et al.*, "Parametric local multimodal hashing for cross-view similarity search," in *Proc. Int. Joint Conf. Artif. Intell.*, Beijing, China, 2013, pp. 2754–2760.
[42] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Comput.*, vol. 15, no. 6, pp. 1373–1396, 2003.
[43] X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-supervised learning using Gaussian fields and harmonic functions," in *Proc. Int. Conf. Mach. Learn.*, Washington, DC, USA, 2003, pp. 912–919.
[44] K. B. Petersen and M. S. Pedersen, *The Matrix Cookbook*, Tech. Univ. Denmark, Kongens Lyngby, Denmark, 2012.
[45] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," California Inst. Technol., Pasadena, CA, USA, Tech. Rep. 7694, 2007.
[46] J. Blitzer, M. Dredze, and F. Pereira, "Biographies, bollywood, boomboxes and blenders: Domain adaptation for sentiment classification," in *Proc. Conf. Assoc. Comput. Linguist.*, Prague, Czech Republic, 2007, pp. 440–447.
[47] 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.
[48] Y. Shi and F. Sha, "Information-theoretical learning of discriminative clusters for unsupervised domain adaptation," in *Proc. Int. Conf. Mach. Learn.*, Edinburgh, U.K., 2012, pp. 1–8.
[49] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded-up robust features (SURF)," *Comput. Vis. Image Understand.*, vol. 110, no. 3, pp. 346–359, 2008.
[50] M. Yang, L. Zhang, X. Feng, and D. Zhang, "Fisher discrimination dictionary learning for sparse representation," in *Proc. IEEE Int. Conf. Comput. Vis.*, Barcelona, Spain, 2011, pp. 543–550.
[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] B. Gong, K. Grauman, and F. Sha, "Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation," in *Proc. Int. Conf. Mach. Learn.*, Atlanta, GA, USA, 2013, pp. 222–230.
[53] J. Blitzer, R. McDonald, and F. Pereira, "Domain adaptation with structural correspondence learning," in *Proc. Conf. Empir. Methods Nat. Lang. Process.*, Sydney, NSW, Australia, 2006, pp. 120–128.
[54] J. Huang, A. J. Smola, A. Gretton, K. M. Borgwardt, and B. Scholkopf, "Correcting sample selection bias by unlabeled data," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, 2007, pp. 601–608.

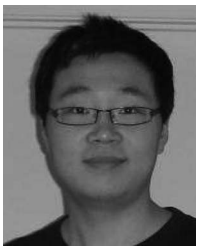
Yuewei Lin (S'09) received the B.S. degree in optical information science and technology from Sichuan University, Chengdu, China, and the M.E. degree in optical engineering from Chongqing University, Chongqing, China. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, University of South Carolina, Columbia, SC, USA.

His current research interests include computer vision, machine learning, and image/video processing.



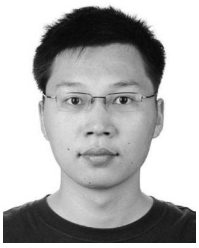
Jing Chen received the B.S. degree in computational mathematics from Shandong University, Jinan, China, and the M.S. and Ph.D. degrees in computer science and technology from Chongqing University, Chongqing, China.

He is currently a Post-Doctoral Fellow with the Faculty of Science and Technology, University of Macau, Macau, China. His current research interests include machine learning, pattern recognition, and biomedical information processing and analysis.



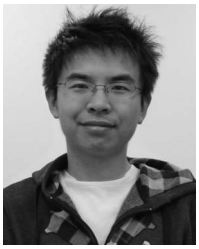
Yu Cao (M'14) received the B.S. degree in information and computation science and the M.S. degree in applied mathematics from Northeastern University, Shenyang, China, in 2003 and 2007, respectively, and the Ph.D. degree in computer science and engineering from the University of South Carolina, Columbia, SC, USA, in 2013.

He is currently a Post-Doctoral Researcher with IBM Almaden Research Center, San Jose, CA, USA. His current research interests include computer vision, machine learning, pattern recognition, and medical image processing.



Youjie Zhou received the B.S. degree in software engineering from East China Normal University (ECNU), Shanghai, China, in 2010. He is currently pursuing the Ph.D. degree in computer science and engineering with the University of South Carolina, Columbia, SC, USA.

He is a Research Assistant with the Computer Vision Laboratory, University of South Carolina. From 2007 to 2010, he was a Research Assistant with the Institute of Massive Computing, ECNU, where he researched on multimedia news exploration and retrieval. His current research interests include computer vision, machine learning, and large-scale multimedia analysis.



Lingfeng Zhang (S'15) received the B.S. degree in mathematics and the M.S. degree in computer science from Chongqing University, Chongqing, China, in 2009 and 2012, respectively. He is currently pursuing the Ph.D. degree with the University of Houston, Houston, TX, USA.

His current research interests include machine learning and big data analysis.

Yuan Yan Tang (F'04) received the Ph.D. degree in computer science from Concordia University, Montreal, QC, Canada, in 1990.

He is a Chair Professor with the Faculty of Science and Technology, University of Macau, Macau, China, and a Professor/Adjunct Professor/Honorary Professor at several institutes in China, USA, Canada, France, and Hong Kong. He has published over 400 academic papers and authored/co-authored over 25 monographs/books/bookchapters. His current research interests include wavelets, pattern recognition, image processing, and artificial intelligence.

Dr. Tang is the Founder and the Editor-in-Chief of the *International Journal on Wavelets, Multiresolution, and Information Processing* and an Associate Editor of several international journals. He is the Founder and the Chair of Pattern Recognition Committee in the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS and the Macau Branch of International Associate of Pattern Recognition (IAPR). He is the Founder and the General Chair of the series International Conferences on Wavelets Analysis and Pattern Recognition. He has served as the General Chair, the Program Chair, and a Committee Member for several international conferences. He is a fellow of IAPR.

Song Wang (SM'13) received the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana–Champaign (UIUC), Urbana, IL, USA, in 2002.

From 1998 to 2002, he was a Research Assistant with the Image Formation and Processing Group, Beckman Institute, UIUC. In 2002, he joined the Department of Computer Science and Engineering, University of South Carolina, Columbia, SC, USA, where he is currently a Professor. His current research interests include computer vision, medical image processing, and machine learning.

Prof. Wang currently serves as the Publicity/Web Portal Chair of the Technical Committee of Pattern Analysis and Machine Intelligence, the IEEE Computer Society, and an Associate Editor of *Pattern Recognition Letters*.