©ISTOCKPHOTO.COM/HH5800

Visual Domain Adaptation

[A survey of recent advances]

[Vishal M. Patel, Raghuraman Gopalan, Ruonan Li, and Rama Chellappa]

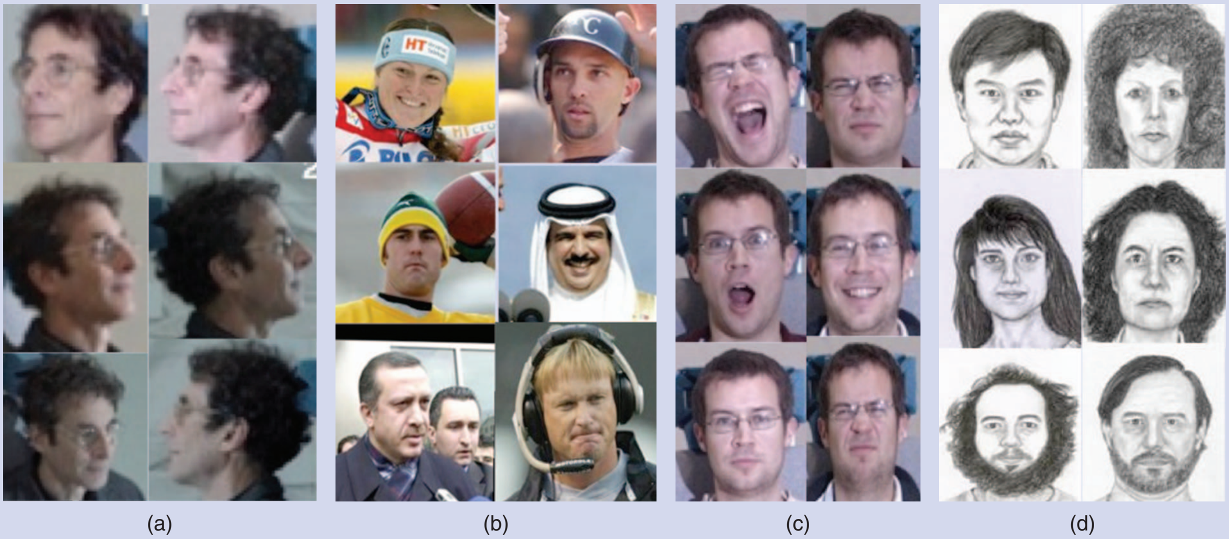
In pattern recognition and computer vision, one is often faced with scenarios where the training data used to learn a model have different distribution from the data on which the model is applied. Regardless of the cause, any distributional change that occurs after learning a classifier can degrade its performance at test time. Domain adaptation tries to mitigate this degradation. In this article, we provide a survey of domain adaptation methods for visual recognition. We discuss the merits and drawbacks of existing domain adaptation approaches and identify promising avenues for research in this rapidly evolving field.

Supervised learning techniques have made tremendous contributions to machine learning and computer vision leading to the development of robust algorithms that are applicable in practical scenarios. While these algorithms have significantly advanced the state of the art, their performance is often limited by the amount of labeled training data available. Labeling is expensive and time-consuming due to the great amount of human effort involved. However, collecting unlabeled visual data is becoming considerably easier due to the availability of low-cost consumer and surveillance cameras, and large Internet databases such as Flickr and YouTube. These data often come from multiple sources and modalities. Thus, when designing a classification or retrieval algorithm using these heterogeneous data, one has to constantly deal with the changing distribution of these data samples. Examples of such cases include: recognizing objects under poor lighting conditions and poses while algorithms are trained on well-illuminated objects at frontal pose, detecting and segmenting an organ of interest from magnetic resonance imaging (MRI) images when available algorithms are instead optimized for computed tomography and X-ray images, recognizing and detecting human faces on infrared images while algorithms are optimized for color images, etc.

This challenge is commonly referred to as *covariate shift* [1] or *data set bias* [2], [3]. Any distributional change or domain shift that occurs after training can degrade the performance at test time. For instance, in the case of face recognition, to achieve useful performance in the wild, face representation and recognition methods must learn to adapt to distributions specific to each application domain shown in Figure 1. Domain adaptation tackles this problem by leveraging domain shift characteristics

Digital Object Identifier 10.1109/MSP.2014.2347059
Date of publication: 6 April 2015

1053-5888/15©2015IEEE



[FIG1] (a) Unconstrained face images. (b) Images with expression variations. (c) Images with pose variations. (d) Sketch images. Real-world object recognition algorithms, such as face recognition, must learn to adapt to distributions specific to each domain shown in (a)–(d) [91], [102], [103].

from labeled data in a related domain when learning a classifier for unseen data. Although some special kinds of domain adaptation problems have been studied under different names such as covariate shift [1], class imbalance [4], and sample selection bias [5], [6], it only started gaining significant interest very recently in computer vision. There are also some closely related but not equivalent machine-learning problems that have been studied extensively, including transfer learning or multitask learning [7], self-taught learning [8], semisupervised learning [9], and multiview analysis [10]. A review of domain adaptation methods from machine-learning and natural language processing communities can be found in [11]. Our goal in this article is to survey recent domain adaptation approaches for computer vision applications, discuss their advantages and disadvantages, and identify interesting open problems.

NOTATION AND RELATED LEARNING PROBLEMS

In this section, we introduce the notation and formulate the domain adaptation learning problem. Furthermore, we discuss the similarities and differences among the various learning problems related to domain adaptation.

NOTATION AND FORMULATION

We refer to the training data set with plenty of labeled data as the source domain and the test data set with a few labeled data or no labeled data as the target domain. Following [11], let X and Y denote the input (data) and the output (label) random variables, respectively. Let $P(X,Y)$ denote the joint probability distribution of X and Y. In domain adaptation, the target distribution is generally different than the source distribution and the true underlying joint distribution $P(X,Y)$ is unknown. We have two different distributions: one for the target domain and the other for the source domain. We denote the joint distribution in the source domain and the target domain as $P_s(X,Y)$ and $P_t(X,Y)$, respectively. The marginal distributions of X and Y in the source and the target domains are denoted by $P_s(X), P_s(Y), P_t(X), P_t(Y)$, respectively. Similarly, the conditional distributions in the two domains are denoted by $P_s(X|Y), P_s(Y|X), P_t(X|Y), P_t(Y|X)$. The joint probability of $X=x$ and $Y=y$ is denoted by $P(X=x, Y=y) = P(x, y)$. Here, $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, where $\mathcal{X}$ and $\mathcal{Y}$ denote the instance space and class label spaces, respectively.

Let $\mathcal{S} = \{(\mathrm{x}_i^s, y_i^s)\}_{i=1}^{N_s}$, where $\mathrm{x}^s \in \mathbb{R}^N$ denote the labeled data from the source domain. Here, x^s is referred to as an *observation*, and y^s is the *corresponding class label*. Labeled data from the target domain is denoted by $\mathcal{T}_l = \{(\mathrm{x}_i^{tl}, y_i^{tl})\}_{i=1}^{N_{tl}}$, where $\mathrm{x}^{tl} \in \mathbb{R}^M$. Similarly, unlabeled data in the target domain is denoted by $\mathcal{T}_u = \{\mathrm{x}_i^{tu}\}_{i=1}^{N_{tu}}$, where $\mathrm{x}^{tu} \in \mathbb{R}^M$. Unless specified otherwise, we assume $N = M$. Let $\mathcal{T} = \mathcal{T}_l \cup \mathcal{T}_u$. As a result, the total number of samples in the target domain is denoted by N_t, which is equal to $N_{tl} + N_{tu}$. Denote $\mathbf{S} = [\mathrm{x}_1^s, \cdots, \mathrm{x}_{N_s}^s]$ as the matrix of N_s data points from $\mathcal{S}$. Denote $\mathbf{T}_l = [\mathrm{x}_1^{tl}, \cdots, \mathrm{x}_{N_{tl}}^{tl}]$ as the matrix of N_{tl} data from $\mathcal{T}_l$, $\mathbf{T}_u = [x_1^{tu}, \cdots, x_{N_{tu}}^{tu}]$ as the matrix of N_{tu} data from $\mathcal{T}_u$ and $\mathbf{T} = [\mathbf{T}_l | \mathbf{T}_u] = [\mathrm{x}_1^t, \cdots, \mathrm{x}_{N_t}^t]$ as the matrix of N_t data from $\mathcal{T}$.

It is assumed that both the target and source data pertain to C classes or categories. Furthermore, it is assumed that all categories have some labeled data. 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, $N_s \gg N_{tl}$.

The goal of domain adaptation is to learn a function $f(.)$ that predicts the class label of a novel test sample from the target domain. Depending on the availability of the source and target domain data, the domain adaptation problem can be defined in many different ways.

- In semisupervised domain adaptation, the function $f(.)$ is learned using the knowledge in $\mathcal{S}$ and $\mathcal{T}_l$.

- In unsupervised domain adaptation, the function $f(.)$ is learned using the knowledge in $\mathcal{S}$ and $\mathcal{T}_u$.
- In multisource domain adaptation, $f(.)$ is learned from more than one domain in $\mathcal{S}$ accompanying each of the first two cases.
- Finally, in the heterogeneous domain adaptation, the dimensions of features in the source and target domains are assumed to be different. In other words, $N \neq M$.

RELATED APPROACHES

COVARIATE SHIFT

One variation of the domain adaptation problem is where, given an observation, the conditional distributions of Y are the same in the source and the target domains, but the marginal distributions of X differ in the two domains. In other words, $P_t(Y|X=x) = P_s(Y|X=x)$ for all $x \in \mathcal{X}$, but $P_t(X) \neq P_s(X)$. This resulting difference between the two domains is known as *covariate shift* [1] or *sample selection bias* [5], [6].

DOMAIN ADAPTATION IS A FUNDAMENTAL PROBLEM IN MACHINE LEARNING AND HAS GAINED A LOT OF TRACTION IN NATURAL LANGUAGE PROCESSING, STATISTICS, MACHINE LEARNING, AND, RECENTLY, IN COMPUTER VISION.

Instance weighting methods can be used to address this covariate shift problem in which estimated weights are incorporated into a loss function in an attempt to make the weighted training distribution look like the testing distribution [11]. To see this, let us briefly review the empirical risk minimization framework for supervised learning [12]. Let $\theta \in \Theta$ be a model family from which we want to select an optimal parameter θ^* for the inference. Let $g(x,y,\theta)$ be a loss function. We want to minimize the following objective function:

$$\theta^* = \arg\min_{\theta\in\Theta} \sum_{(x,y)\in\mathcal{X}\times\mathcal{Y}} P(x,y)g(x,y,\theta)$$

to obtain the optimal θ^* for the distribution $P(X,Y)$. Since $P(X,Y)$ is unknown, we use the empirical distribution $\tilde{P}(X,Y)$ to estimate $P(X,Y)$. A good model $\hat{\theta}$ can be found by minimizing the following empirical risk:

$$\begin{aligned}\hat{\theta} &= \arg\min_{\theta\in\Theta} \sum_{(x,y)\in\mathcal{X}\times\mathcal{Y}} \tilde{P}(x,y)g(x,y,\theta)\\ &= \arg\min_{\theta\in\Theta} \sum_{i=1}^{N} g(x_i,y_i,\theta),\end{aligned}$$

where $\{(x_i,y_i)\}_{i=1}^{N}$ is a set of training instances randomly sampled from $P(X,Y)$. This formulation can be extended to domain adaptation by minimizing the following expected loss over the target domain distribution to find the optimal model parameter for the target domain [11]:

$$\theta_t^* = \arg\min_{\theta\in\Theta} \sum_{(x,y)\in\mathcal{X}\times\mathcal{Y}} P_t(x,y)g(x,y,\theta).$$

In domain adaptation setting, the training instances $\{(x_i^s,y_i^s)\}_{i=1}^{N_s}$ are randomly sampled from the source distribution $P_s(X,Y)$. As a result, we get

$$\begin{aligned}\theta_t^* &= \arg\min_{\theta\in\Theta} \sum_{(x,y)\in\mathcal{X}\times\mathcal{Y}} \frac{P_t(x,y)}{P_s(x,y)}P_s(x,y)g(x,y,\theta)\\ &\approx \arg\min_{\theta\in\Theta} \sum_{(x,y)\in\mathcal{X}\times\mathcal{Y}} \frac{P_t(x,y)}{P_s(x,y)}\tilde{P}_s(x,y)g(x,y,\theta)\\ &= \arg\min_{\theta\in\Theta} \sum_{i=1}^{N_s} \frac{P_t(x_i^s,y_i^s)}{P_s(x_i^s,y_i^s)}g(x_i^s,y_i^s,\theta).\end{aligned} \tag{1}$$

As can be seen from (1), weighting the loss of the source samples by $(P_t(x,y)/P_s(x,y))$ provides a solution to the domain adaptation problem [11].

Under covariate shift, the ratio $(P_t(x,y)/P_s(x,y))$ can be rewritten as:

$$\frac{P_t(x,y)}{P_s(x,y)} = \frac{P_t(x)}{P_s(x)}\frac{P_t(y|x)}{P_s(y|x)} = \frac{P_t(x)}{P_s(x)}.$$

As a result, one can weigh each training instance with $(P_t(x)/P_s(x))$. Shimodaira [1] explored this approach to reweight the log likelihood of each training instance using $(P_t(x)/P_s(x))$ for covariate shift. Various methods can be used to estimate the ratio $(P_t(x)/P_s(x))$. For instance, nonparametric density estimation [1], [13] and kernel mean match-based methods [14] have been proposed in the literature to directly estimate the ratio.

CLASS IMBALANCE

Another special case of the domain adaptation formulation assumes that $P_t(X|Y=y) = P_s(X|Y=y)$ for all $y \in \mathcal{Y}$, but $P_t(Y) \neq P_s(Y)$. This difference is often known as *class imbalance* [4]. Under this assumption, the ratio in (1) can be rewritten as:

$$\frac{P_t(x,y)}{P_s(x,y)} = \frac{P_t(y)}{P_s(y)}\frac{P_t(x|y)}{P_s(x|y)} = \frac{P_t(y)}{P_s(y)}.$$

As a result, one only needs to consider $(P_t(y)/P_s(y))$ to weigh the instances [15].

Resampling can also be applied on the training instances from the source domain so that the resampled data roughly has the same class distribution as the target domain. In these methods, underrepresented classes are oversampled and overrepresented classes are undersampled [11].

TRANSFER LEARNING

Multitask learning or transfer learning is closely related to domain adaptation [7], [16]. In multitask learning, different tasks are considered, but the marginal distribution of the source and target data are similar. In other words, assuming L tasks, the joint probability of each task $\{P(X,Y_i)\}_{i=1}^{L}$ is different, but there is only a single distribution $P(X)$ of the observation. When learning the class conditional models $\{P(Y_i|X,\theta_i)\}_{i=1}^{L}$ for L tasks, it is assumed that the model parameters of the individual tasks are drawn from a common prior distribution $P_\Theta(\theta)$.

Since domain adaptation considers only a single task but different domains, it is a somewhat different problem than multitask learning. However, one can view domain adaptation as a special case of multitask learning with two tasks, one on the source domain and the other on the target domain. In fact, some domain adaptation methods are essentially solving transfer learning problems. We refer you to [16] for a comprehensive survey on various transfer learning methods.

SEMISUPERVISED LEARNING

The performance of a supervised classification algorithm is often dependent on the availability of a sufficient amount of training data. However, labeling samples is expensive and time-consuming due to the significant human effort involved. As a result, it is desirable to have methods that learn a classifier with high accuracy from only a limited amount of labeled training data. In semisupervised learning, unlabeled data are exploited to remedy the lack of labeled data. This in turn requires that the unlabeled data comes from the same distribution as the labeled data. Hence, if we ignore the domain difference, and treat the labeled source instances as labeled data and the unlabeled target domain instances as unlabeled data, then the resulting problem is that of the semisupervised learning problem. As a result, one can apply any semisupervised learning algorithm [9] to the domain adaptation problem. The subtle difference between domain adaptation and semisupervised learning comes from the following two facts [11]:

- The amount of labeled data in semisupervised learning is small but large in domain adaptation.
- The labeled data may be noisy in domain adaptation if one does not assume $P_s(Y|X=x)=P_t(Y|X=x)$ for all x, whereas, in semisupervised learning, the labeled data are assumed to be reliable.

In fact, there have been several works in the literature that extend semisupervised learning methods to domain adaptation. A naive Bayes' transfer classifier algorithm, which allows for the training and test data distributions to be different for text classification, was proposed in [17]. This algorithm first estimates the initial probabilities under a distribution of one labeled data set and then uses an expectation maximization (EM) algorithm to revise the model for a different distribution of the test data which are assumed to be unlabeled. This EM-based domain adaptation method can be shown to be equivalent to a semisupervised EM algorithm [18]. Some of the other methods that extend domain adaptation using semisupervised learning include [19] and [20].

SELF-TAUGHT LEARNING

Another problem related to domain adaptation and semisupervised learning is self-taught learning [8], [21]. In self-taught learning, we are given limited data for a classification task and also large amounts of unlabeled data that are only mildly related to the task. In particular, the unlabeled data may not arise from the same distribution or share the class labels. This assumption essentially differentiates self-taught learning from semisupervised learning. Self-taught learning is motivated by the observation that many randomly downloaded images contain basic visual features, such as edges and corners, that are similar to those in the training images. As a result, if one is able to learn to recognize such patterns from the unlabeled data, then these features can be used for the supervised learning task of interest [8].

A sparse coding-based approach was proposed in [8] for self-taught learning, where a dictionary is learned using unlabeled data. Then, higher-level features are computed by solving a convex ℓ_1-regularized least squares problem using the learned dictionary and the labeled training data. Finally, a classifier is trained by applying a supervised learning algorithm such as a support vector machine (SVM) on these higher-level labeled features. A discriminative version of this algorithm was also presented in [22]. Furthermore, an unsupervised self-taught learning algorithm called *self-taught clustering* was proposed in [23]. Self-taught clustering aims at clustering a small collection of target unlabeled data with the help of a large amount of auxiliary unlabeled data. It is assumed that the target and auxiliary data have a different distribution. It was shown that this algorithm can greatly outperform several state-of-the-art clustering methods when using irrelevant unlabeled data.

MULTIVIEW ANALYSIS

In many computer vision applications, data often come in multiple views or styles. For instance, in object recognition, one has to deal with objects in different poses (views) and lighting conditions. As a result, one is faced with the problem of classifying or retrieving objects where the source (gallery) and target (query) data belong to different views. A direct comparison of instances across different views is not meaningful since they lie in different feature spaces.

In a multiview (also known as *cross-view* or *multimodal*) learning setting, correspondences are assumed to be known between the two view samples. In other words, samples are often given in pairs corresponding to different views. This assumption essentially differentiates cross-view learning from domain adaptation, where no correspondences are assumed between the domain samples. One popular solution in multiview learning is to learn view-specific projection directions using the paired samples from different views (domains) into a common latent space [10]. Classification or retrieval can then be performed in the latent space, where both the target and source data share the same feature space. Other methods for multiview learning include [24]–[28].

VISUAL DOMAIN ADAPTATION APPROACHES

Domain adaptation is a fundamental problem in machine learning and has gained a lot of traction in natural language processing, statistics, machine learning, and, recently, in computer vision. Early visual domain adaptation methods were applied to domain shift in videos [29], [30]. In particular, Duan et al. [30] proposed to adapt video concept classifiers between news videos collected from different news channels. Since then, there have been a plethora of approaches proposed in the vision literature for object category adaptation. In what follows, we present a number of recent domain adaptation strategies for visual recognition.

FEATURE AUGMENTATION-BASED APPROACHES

One of the simplest domain adaptation approaches is the feature augmentation work of Daumé III [31]. The goal is to make a domain-specific copy of the original features for each domain. Each feature in the original domain of dimension N is mapped onto an augmented space of dimension $3N$ simply by duplicating the feature vectors. The augmented feature maps for the source and target domains are defined as

$$\Phi^s(\mathbf{x}_i^s)=\begin{bmatrix}\mathbf{x}_i^s\\ \mathbf{x}_i^s\\ \mathbf{0}_N\end{bmatrix},\qquad \Phi^t(\mathbf{x}_i^{tl})=\begin{bmatrix}\mathbf{x}_i^{tl}\\ \mathbf{0}_N\\ \mathbf{x}_i^{tl}\end{bmatrix}, \tag{2}$$

where $\mathbf{x}_i^s\in\mathcal{S}$, $\mathbf{x}_i^{tl}\in\mathcal{T}_l$, and $\mathbf{0}_N$ denotes a zero vector of dimension N. The first N-dimensional component of this augmented feature corresponds to commonality between source and target, the second N-dimensional component corresponds to the source, while the last component corresponds to the target domain. Both source and target domain features are transformed using these augmented feature maps, and the resulting feature is passed onto the underlying supervised classifier. It was shown in [31] that when linear classifiers are used, this feature augmentation method is equivalent to decomposing the model parameter θ_i for domain i into $\tilde{\theta}_i+\theta_c$, where θ_c is shared by all domains. This "frustratingly easy" feature augmentation framework can be easily extended to a multidomain case by making more copies of the original feature space. Furthermore, a kernel version of this method is also derived in [31].

A feature augmentation-based method for utilizing the heterogeneous data from the source and target domains was recently proposed in [32]. The approach taken in [32] is to introduce a common subspace for the source and target data so that the heterogeneous features from two domains can be compared. In particular, both the source and target data of dimension N and M, respectively, are projected onto a latent domain of dimension l using two projection matrices $\mathbf{W}_1\in\mathbb{R}^{l\times N}$ and $\mathbf{W}_2\in\mathbb{R}^{l\times M}$, respectively. The augmented feature maps for the source and target domains in the common space are then defined as

$$\Phi^s(\mathbf{x}_i^s)=\begin{bmatrix}\mathbf{W}_1\mathbf{x}_i^s\\ \mathbf{x}_i^s\\ \mathbf{0}_M\end{bmatrix}\in\mathbb{R}^{l+N+M}, \tag{3}$$

$$\Phi^t(\mathbf{x}_i^{tl})=\begin{bmatrix}\mathbf{W}_2\mathbf{x}_i^{tl}\\ \mathbf{0}_N\\ \mathbf{x}_i^{tl}\end{bmatrix}\in\mathbb{R}^{l+N+M}, \tag{4}$$

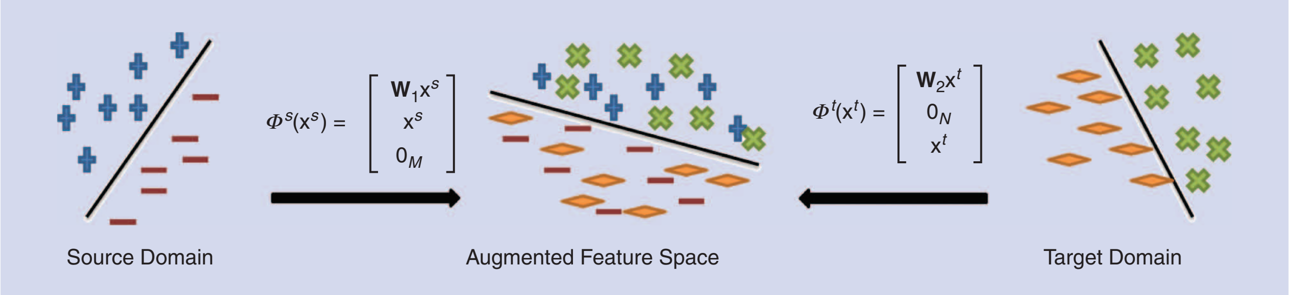
where $\mathbf{x}_i^s\in\mathcal{S}$, $\mathbf{x}_i^{tl}\in\mathcal{T}_l$, and $\mathbf{0}_M$ is an M-dimensional zero vector. Once the data from both domains are transformed onto a common space, they can be readily passed onto a supervised classifier [32]. Figure 2 illustrates an overview of this method.

The general idea behind the frustratingly easy feature augmentation method of Daumé III [31] has been extended to consider a manifold of intermediate domains [33], [34]. Manifold-based methods for unsupervised visual domain adaptation were first proposed by Gopalan et al. [33]. Rather than working with the information conveyed by the source and target domains alone, [33] proposes using incremental learning by gradually following the geodesic path between the source and target domains. Geodesic flows are used to derive intermediate subspaces that interpolate between the source and target domains. Figure 3 shows an overview of this method.

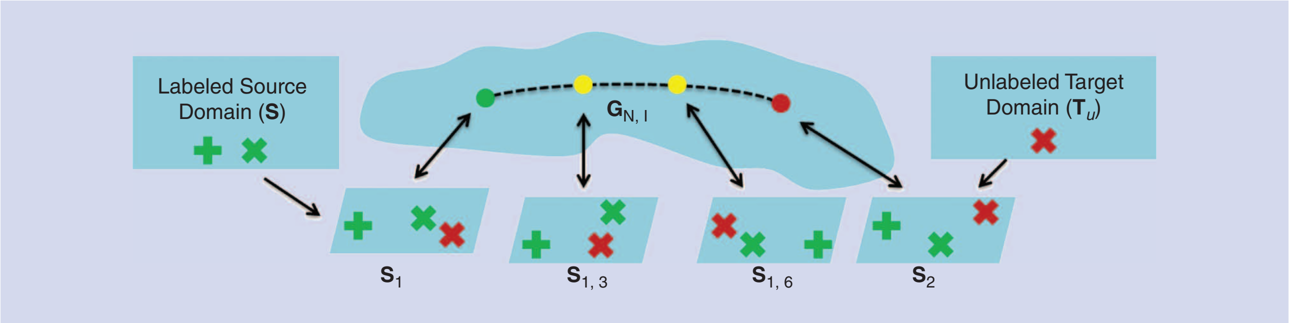
GEODESIC FLOWS ARE USED TO DERIVE INTERMEDIATE SUBSPACES THAT INTERPOLATE BETWEEN THE SOURCE AND TARGET DOMAINS.

It is assumed that the dimension of features in both the source and target domains is the same, e.g., $N=M$. First, principal component analysis (PCA) is applied on $\mathbf{S}$ and $\mathbf{T}_u$, which generates two l-dimensional subspaces dented by two matrices $\mathbf{S}_1$ and $\mathbf{S}_2$, respectively, where $l<N$. The space of l-dimensional subspaces in $\mathbb{R}^N$ containing origin can be identified with the Grassmann manifold $\mathbb{G}_{N,l}$. As a result, $\mathbf{S}_1$ and $\mathbf{S}_2$ can be viewed as points on $\mathbb{G}_{N,l}$. By viewing $\mathbb{G}_{N,l}$ as quotient space of $SO(N)$, [here, $SO(N)$ represents the special orthogonal group, which is the group of orthogonal $N\times N$ matrices with determinant 1], the geodesic path in $\mathbb{G}_{N,l}$ starting from $\mathbf{S}_1$ is given by a one-parameter exponential flow $\Psi(t')=\mathbf{Q}\exp(t'\mathbf{B})\mathbf{J}$, where exp refers to the matrix exponential, $\mathbf{Q}\in SO(N)$ such that $\mathbf{Q}^T\mathbf{S}_1=\mathbf{J}$ and

$$\mathbf{J}=\begin{bmatrix}\mathbf{I}_l\\ \mathbf{0}_{N-l,l}\end{bmatrix}.$$


$\Phi^s(x^s)=\begin{bmatrix}W_1x^s\\ x^s\\ 0_M\end{bmatrix}$
$\Phi^t(x^t)=\begin{bmatrix}W_2x^t\\ 0_N\\ x^t\end{bmatrix}$
Source Domain
Augmented Feature Space
Target Domain

[FIG2] By using two projection matrices $\mathbf{W}_1$ and $\mathbf{W}_2$, one can transform the heterogeneous samples from two domains into an augmented feature space [32].



[FIG3] An overview of the manifold-based unsupervised domain adaptation method [33]. With labeled data S from source domain corresponding to two classes + and ×, and unlabeled data T_u from target domain belonging to class ×, generative subspaces S_1 and S_2 are derived using PCA. Then, by viewing S_1 and S_2 as points on the Grassmann manifold $\mathbb{G}_{N,l}$ (green and red circles), points along the geodesic between them (dashed line) are sampled to obtain geometrically meaningful intermediate subspaces (yellow circles).

Here, I_l is a $l \times l$ identity matrix and B is a skew-symmetric, block-diagonal matrix of the form

$$B = \begin{bmatrix} 0 & A^T \\ -A & 0 \end{bmatrix},$$

$A \in \mathbb{R}^{(N-l)\times l}$, where $(.)^T$ denotes the transposition operation and the submatrix A specifies the direction and the speed of geodesic flow. The geodesic flow between S_1 and S_2 is obtained by computing the direction matrix A such that the geodesic along that direction, while starting from S_1, reaches S_2 in unit time. The matrix A is computed using the inverse exponential mapping. Once A is computed, the expression for $\Psi(t')$ is used to obtain the intermediate subspaces between S_1 and S_2 by varying the value of t' between 0 and 1.

Let S′ be the collection of subspaces $S_t, t \in \mathbb{R}, 1 \le t \le 2$, which includes S_1 and S_2 and all intermediate subspaces. Let k denote the total number of such subspaces. The intermediate cross-domain data representations U are obtained by projecting the source data S and the target data T_u onto S′. The final feature representation of dimension lk is obtained by projecting data onto k different subspaces. A model on these extended features is learned using partial least squares (PLS), and the assignment of target labels is performed using the nearest neighbor method [33]. A nonlinear version of this method, as well as an extension to semisupervised domain adaptation, has also been presented in [34]. Furthermore, assuming that the domain to which samples belong has been identified a priori [35], [36], this method has been extended to multidomain adaptation in [34].

Recently, the approach of [33] 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 S_1 and S_2, respectively [37]–[39]. Furthermore, assuming that the data lie in a union of subspaces in both the source and target domains, a framework based on the parallel transport of a union of the source subspaces on the Grassmann manifold was proposed in [40]. It was shown that this way of modeling data with a union of subspaces instead of a single subspace significantly improves the recognition performance [40].

FEATURE TRANSFORMATION-BASED APPROACHES

One of the earliest object category adaptation methods was proposed by Saenko et al. [41]. The idea behind this method is to adapt features across general image domains by learning transformations. Given feature vectors $x^s \in \mathcal{S}$ and $x^t \in \mathcal{T}$, a linear transformation $W \in \mathbb{R}^{N\times M}$ from $\mathcal{T}$ to $\mathcal{S}$ is learned. The inner product similarity function between x^s and the transformed x^t is denoted by

$$\mathrm{sim}_W = (x^s)^T W x^t. \tag{5}$$

One can view this function as an inner product between the transformed target point Wx^t and x^s. The objective is to learn the linear transformation given some form of supervision and then to use the learned similarity function in a classification algorithm [41]. A regularization function for the matrix W is introduced to avoid overfitting, which is denoted as $r(W)$. Assume that the supervision is a function of the learned similarity values sim_W, so a general optimization problem would seek to minimize the regularizer subject to supervision constraints given by functions c_i

$$\min_W r(W) \text{ s.t. } \quad c_i(S^T W T) \ge 0, \quad 1 \le i \le J. \tag{6}$$

Equation (6) can be written as an unconstrained problem

$$\min_W r(W) + \lambda \sum_i c_i(S^T W T). \tag{7}$$

The regularizer studied in [41] is

$$r(W) = \mathrm{trace}(W) - \log\det(W), \tag{8}$$

and the resulting optimization problem is solved using an information-theoretic metric-learning [42] type of algorithm. One of the limitations of this method is that it can only be applied when the dimensionalities of the two domains are the same (e.g., $N = M$).

This work was extended in [43] by Kulis et al. to the more general case where the domains are not restricted to be the same dimensionality and arbitrary asymmetric transformations can be learned. Their method can deal with more general types of domain shifts and changes in feature type and dimension. Furthermore, they show that the method in [41] is a special case of their general formulation, producing symmetric positive definite transformations [43]. It was shown that asymmetric indefinite transformations are more flexible for a variety of adaptation tasks than the symmetric transformations.

Recently, a low-rank approximation-based approach for semi-supervised domain adaptation was proposed in [44]. The basic goal of this method is to map the source data by a matrix $\mathrm{W} \in \mathbb{R}^{N \times N}$ to an intermediate representation where each transformed sample can be reconstructed by a linear combination of the target data samples

$$\mathrm{WS} = \mathrm{T}_l \mathrm{Z}, \tag{9}$$

where $\mathrm{Z} \in \mathbb{R}^{N_{tl} \times N_s}$ is the coefficient matrix. The following formulation is proposed to solve for the low-rank solution:

$$\begin{aligned}(\hat{\mathrm{W}}, \hat{\mathrm{Z}}, \hat{\mathrm{E}}) &= \min_{\mathrm{W}, \mathrm{Z}, \mathrm{E}} \operatorname{rank}(\mathrm{Z}) + \lambda \| \mathrm{E} \|_{2,1}, \\ \text{s.t.} \quad \mathrm{WS} &= \mathrm{T}_l \mathrm{Z} + \mathrm{E}, \quad \mathrm{WW}^{\mathrm{T}} = \mathrm{I},\end{aligned} \tag{10}$$

where $\operatorname{rank}(.)$ denotes the rank of a matrix, λ is a parameter, $\mathrm{E} \in \mathbb{R}^{N \times N_s}$ is the error term, and the $\ell_{2,1}$-norm is defined as $\| \mathrm{E} \|_{2,1} = \sum_{i=1}^{N} \sqrt{\sum_{j=1}^{N_s} E_{ij}^2}$. As a common practice in rank minimization problems, the rank of Z is replaced by its nuclear norm in (10) [44]. The augmented Lagrange multiplier method is proposed to solve the optimization problem.

Once the solution $(\hat{\mathrm{W}}, \hat{\mathrm{Z}}, \hat{\mathrm{E}})$ is obtained, the source data are transformed to the target domain as

$$\hat{\mathrm{W}}\mathrm{S} - \hat{\mathrm{E}}. \tag{11}$$

The transformed source data are mixed with the target samples as the augmented training samples for training the classifiers. The trained classifier is then used to perform recognition on the unseen test samples in the target domain [44]. An extension of this method for the multiple source domain adaptation problem has also been proposed in [44]. Other recent transformation-based visual domain adaptation methods include [45] and [46].

PARAMETER ADAPTATION METHODS

Several algorithms have been proposed in the literature that investigate modifying the SVM algorithms for the domain adaptation problem. In particular, Yang et al. proposed an adaptive SVM (A-SVM) [29] method in which the source classifier $f_S(\mathrm{x})$ trained on the source data $\mathcal{S} = \{(x_i^s, y_i^s)\}_{i=1}^{N_s}$ is adapted to a new classifier $f_T(\mathrm{x})$ for the unseen target data $\mathcal{T}_u = \{x_i^{tu}\}_{i=1}^{N_{tu}}$. The decision function is formulated as

$$f_T(\mathrm{x}) = f_S(\mathrm{x}) + \delta f(\mathrm{x}), \tag{12}$$

where $\delta f(\mathrm{x})$ is the perturbation function. It was shown in [29] that the perturbation function can be formulated as $\delta f(\mathrm{x}) = \theta^T \phi(\mathrm{x})$, where a feature map ϕ is used to project x into a high-dimensional feature vector $\phi(\mathrm{x})$. The perturbation function $\delta f(\mathrm{x})$ is learned using the labeled data $\mathcal{T}_l = \{(\mathrm{x}_i^{tl}, y_i^{tl})\}_{i=1}^{N_{tl}}$ from the target domain. To learn the parameter w of the perturbation function $\delta f(\mathrm{x})$, the following optimization problem is solved:

$$\begin{gathered}\min_{\theta} \frac{1}{2} \| \theta \|^2 + \alpha \sum_{i=1}^{N_{tl}} \xi_i \\ \text{s.t. } \xi_i \geq 0, \\ y_i^{tl} f_S(\mathrm{x}_i^{tl}) + y_i^{tl} \theta^T \phi(\mathrm{x}_i^{tl}) \geq 1 - \xi_i, \forall (\mathrm{x}_i^{tl}, y_i^{tl}) \in \mathcal{T}_l,\end{gathered} \tag{13}$$

where ξ_i is the penalizing variable and α is a parameter that determines how much error an SVM can tolerate. The first term in (13) tries to minimize the deviation between the new decision boundary and the old one, and the second term controls the penalty of the classification error over the training data in the target domain.

This work was improved in [47] for object category detection and in [48] for visual concept classification. Domain transfer SVM [49] attempts to reduce the mismatch in the domain distributions, measured by the maximum mean discrepancy (MMD) while also learning a target decision function. Other SVM-based domain adaptation methods include [50]–[54].

As discussed previously, several domain adaptation methods make use of the kernel methods. The classification performance of these kernel-based methods is highly dependent on the choice of the kernel. Multiple kernel learning (MKL) can be used to combine multiple kernel functions to obtained a better solution [55]. MKL has been shown to work well in many computer vision applications. However, these methods assume that both training and test data come from the same domain. As a result, MKL methods cannot learn the optimal kernel with the combined data from the source and target domains for the domain adaptation problem. Hence, training data from the auxiliary domain may degrade the performance of MKL algorithms in the target domain. To deal with this, several cross-domain kernel learning methods have been proposed in the literature [56]–[58].

In [56], adaptive MKL is used to learn a kernel function based on multiple base kernels. In [57], a kernel function and a classifier are simultaneously learned by minimizing both the structural risk functional and the distribution mismatch between the labeled and unlabeled samples from the auxiliary and target domains. It was shown in [56] and [57] that these domain-adaptive MKL methods can significantly outperform traditional MKL and cross-domain learning methods.

There are some limitations of the feature-based and parameter transfer-based visual domain adaptation methods reviewed in this survey. For instance, the transform-based approaches discussed in [41], [43], [45], and [46] are based on some notion of closeness between the transformed source samples and target samples. They do not optimize the objective function of a discriminative classifier directly. Also, the computational complexity of these methods is highly dependent on the total number of samples used for training. On the other hand, parameter adaptation-based methods such as [29] and [48] optimize the classifier directly but they are not able to transfer the adapted function to novel categories. To deal with this problem, several methods have been developed in the literature that attempt to optimize both the transformation and classifier parameters jointly [59]–[61].

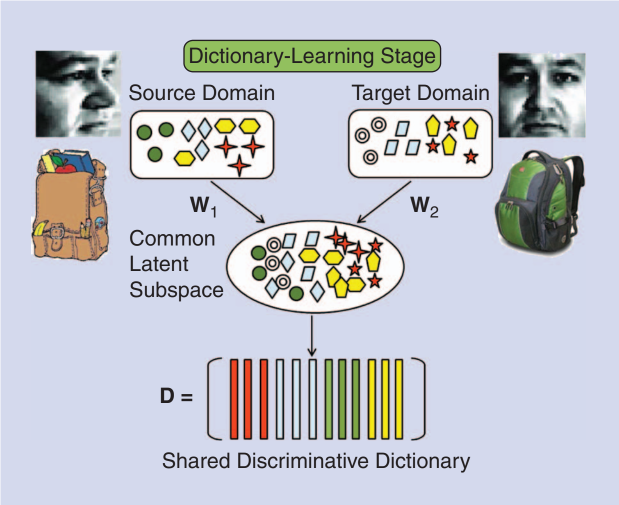
In particular, the max-margin domain transfer method was recently proposed by Hoffman et al. in [60], which uses an asymmetric transform W to map target features to a new representation where they are maximally aligned with the source and learns the transform jointly on all categories for which target labels are available. It provides a way to adapt max-margin classifiers in a multiclass setting by learning a common component of the domain shift as captured by W.

The goal of this method is to jointly learn affine hyperplanes that separate the classes in the source domain and a transformation from the points in the target domain into the source domain such that the transformed target data lie on the correct side of the learned source hyperplanes. For simplicity, let us consider the optimization for the binary problem [60]

$$\begin{aligned}&\min_{\mathrm{W},\theta,b}\frac{1}{2}\|\mathrm{W}\|_F^2+\frac{1}{2}\|\theta\|_F^2\\&\text{s.t } y_i^s\left(\begin{bmatrix}\mathrm{x}_i^s\\1\end{bmatrix}^T\begin{bmatrix}\theta\\b\end{bmatrix}\right)\geq 1\forall i\in\{1,\cdots,N_s\}\\&y_i^{tl}\left(\begin{bmatrix}\mathrm{x}_i^{tl}\\1\end{bmatrix}^T\mathrm{W}^T\begin{bmatrix}\theta\\b\end{bmatrix}\right)\geq 1\forall i\in\{1,\cdots,N_{tl}\},\end{aligned}\qquad(14)$$

where $\boldsymbol{\theta}$ denotes the normal of the affine hyperplane and b is the bias term. This formulation can be easily extended to the multiclass case by adding a sum over the regularizers on all class-specifics parameters and adding the constraints for all categories. The resulting optimization problem is not convex. As a result, it is solved by alternating minimization on W and (θ, b) [60]. This work was extended in [61] to include Laplacian regularization using instance constraints that are encoded by an arbitrary graph.

Another approach to simultaneous learning of domain-invariant features and classifiers was proposed by Shi and Sha in [59]. Their framework is based on the notion of discriminative clustering in which both the source and target domains are assumed to be tightly clustered and clusters are assumed to correspond to class boundaries. It is assumed that for the same class, the clusters from the two domains are geometrically close to each other. Their formulation of learning the optimal feature space is based on maximizing the domain similarity that makes the source and the target domains look alike and minimizing the expected classification error on the target domain. An information-theoretic framework is proposed for solving their formulation [59].



[FIG4] An overview of the domain-adaptive latent space dictionary-learning framework [66].

DICTIONARY-BASED APPROACHES

The study of sparse representation of signals and images has attracted tremendous interest over the last few years. This is partly because signals or images of interest, although high dimensional, can often be coded using few representative atoms in some dictionary. In their seminal work, Olshausen and Field [62] introduced the idea of learning a dictionary from data instead of using off-the-shelf bases. Since then, data-driven dictionaries have been shown to work well for both image restoration and classification tasks [63], [64]. The efficiency of dictionaries in these wide range of applications can be attributed to the robust discriminant representations that they provide by adapting to particular data samples. However, the learned dictionary may not be optimal if the target data have a different distribution than the data used for training. Several dictionary-learning-based methods have been proposed in the literature to deal with this domain shift problem [65]–[68].

A function learning framework for the task of transforming a dictionary learned from one visual domain to the other while maintaining a domain-invariant sparse representation of a signal was proposed in [65]. Domain dictionaries are modeled by a linear or nonlinear parametric function. The dictionary function parameters and domain-invariant sparse codes are then jointly learned by solving an optimization problem. Motivated by the manifold-based incremental learning work of Gopalan et al. [33], [34], Ni et al. [67] proposed an unsupervised domain-adaptive dictionary-learning framework by generating a set of intermediate dictionaries, which smoothly connect the source and target domains. One of the important properties of this approach is that it allows the synthesis of data associated with the intermediate domains while exploiting the discriminative power of generative dictionaries. The intermediate data can then be used to build a classifier for recognition under domain shifts.

In [66], Shekhar et al. proposed a semisupervised domain-adaptive dictionary-learning framework for learning a single dictionary to optimally represent both source and target data. As the features may not be correlated well in the original space, they propose to project data from both the domains onto a common low-dimensional space while maintaining the manifold structure of the data. They argue that learning the dictionary on a low-dimensional space makes the algorithm faster and that irrelevant information in the original features can be discarded. Moreover, joint learning of dictionary and projections ensures that the common internal structure of data in both domains is extracted, which can be represented well by sparse linear combinations of dictionary atoms. Figure 4 shows an overview of this method [66].

Given source and target domain data $\mathbf{S} \in \mathbb{R}^{N \times N_s}$ and $\mathbf{T}_l \in \mathbb{R}^{M \times N_{tl}}$, respectively, Shekhar et al. learn a shared K atom dictionary, $\mathbf{D} \in \mathbb{R}^{l \times K}$, and mappings $\mathbf{W}_1 \in \mathbb{R}^{l \times N}$ and $\mathbf{W}_2 \in \mathbb{R}^{l \times M}$ onto a common low-dimensional space, which will minimize the representation error in the projected space. Formally, the following cost is minimized:

$$C_1(\mathbf{D}, \mathbf{W}_1, \mathbf{W}_2, \mathbf{X}_1, \mathbf{X}_2) = \| \mathbf{W}_1 \mathbf{S} - \mathbf{D}\mathbf{X}_1 \|_F^2 + \| \mathbf{W}_2 \mathbf{T}_l - \mathbf{D}\mathbf{X}_2 \|_F^2$$

subject to sparsity constraints on $\mathbf{X}_1 \in \mathbb{R}^{K \times N_s}$ and $\mathbf{X}_2 \in \mathbb{R}^{K \times N_{tl}}$. It is assumed that rows of the projection matrices, $\mathbf{W}_1$ and $\mathbf{W}_2$, are orthogonal and normalized to unit norm. This prevents the solution from becoming degenerate, leads to an efficient scheme for optimization, and makes the kernelization of the algorithm possible. Note that this method does not require the data to be of the same dimension in the source and target domains. As a result, this method is applicable to heterogeneous domain adaptation problems [32].

To make sure that the projections do not lose too much information available in the original domains after projecting onto the latent space, a PCA-like regularization term is added, which preserves energy in the original signal, given as

$$C_2(\mathbf{W}_1, \mathbf{W}_2) = \| \mathbf{S} - \mathbf{W}_1^{\mathrm{T}} \mathbf{W}_1 \mathbf{S} \|_F^2 + \| \mathbf{T}_l - \mathbf{W}_2^{\mathrm{T}} \mathbf{W}_2 \mathbf{T}_l \|_F^2.$$

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

$$C_1(\mathbf{D}, \tilde{\mathbf{W}}, \tilde{\mathbf{X}}) = \| \tilde{\mathbf{W}} \tilde{\mathbf{Y}} - \mathbf{D} \tilde{\mathbf{X}} \|_F^2, \qquad (15)$$

$$C_2(\tilde{\mathbf{W}}) = - \operatorname{trace}((\tilde{\mathbf{W}}\tilde{\mathbf{Y}})(\tilde{\mathbf{W}}\tilde{\mathbf{Y}})^T), \qquad (16)$$

where

$$\tilde{\mathbf{W}} = [\mathbf{W}_1 \ \mathbf{W}_2], \ \tilde{\mathbf{Y}} = \begin{pmatrix} \mathbf{S} & 0 \\ 0 & \mathbf{T}_l \end{pmatrix}, \text{ and } \tilde{\mathbf{X}} = [\mathbf{X}_1 \ \mathbf{X}_2].$$

Hence, the overall optimization is given as

$$\{\mathbf{D}^*, \tilde{\mathbf{W}}^*, \tilde{\mathbf{X}}^*\} = \arg \min_{\mathbf{D}, \tilde{\mathbf{W}}, \tilde{\mathbf{X}}} C_1(\mathbf{D}, \tilde{\mathbf{W}}, \tilde{\mathbf{X}}) + \lambda C_2(\tilde{\mathbf{W}})$$
$$\text{s.t. } \mathbf{W}_i \mathbf{W}_i^{\mathrm{T}} = \mathbf{I}, \ i = 1,2 \text{ and } \| \tilde{\mathbf{x}}_j \|_0 \le T_0, \forall j, \qquad (17)$$

where λ is a positive constant. An efficient two-step procedure is proposed for solving this optimization problem in [66]. Furthermore, this method has been extended to multiple domains and kernelized in [66]. Once the projection matrices and the dictionary are learned, given a novel test sample form the target domain, it is first projected onto the latent domain using $\mathbf{W}_2$ and classified using a variation of the latent sparse embedding residual classifier (LASERC) algorithm proposed in [69].

DOMAIN RESAMPLING

An unsupervised domain adaptation method was recently proposed in [70] and [71] based on the notion of landmarks. Landmarks are a subset of labeled data instances in the source domain that are distributed most similarly to the target domain [70]. The key insight of their method is that not all instances are created equally for adaptation. As a result, they pick out and exploit the most desirable instances to facilitate adaptation. An overview of this method is shown in Figure 5.

A variant of MMD is used to select samples from the source domain to match the distribution of the target domain. To identify landmarks, N_s indicator variables $\alpha = \{\alpha_i \in \{0,1\}\}$ are used, one for each data point in the source domain. If $\alpha_i = 1$, then $\mathbf{x}_i^s$ is regarded as a landmark. The vector α is identified by minimizing the MMD metric, defined with a kernel mapping function $\phi(\mathbf{x})$,

$$\min_{\alpha} \left\| \frac{1}{\sum_i \alpha_i} \sum_i \alpha_i \phi(\mathbf{x}_i^s) - \frac{1}{N_{tu}} \sum_j \phi(\mathbf{x}_j^{tu}) \right\|_{\mathcal{H}}^2$$
$$\text{s.t.} \ \frac{1}{\sum_i \alpha_i} \sum_i \alpha_i y_{ic} = \frac{1}{N_s} \sum_i y_{ic}, \qquad (18)$$

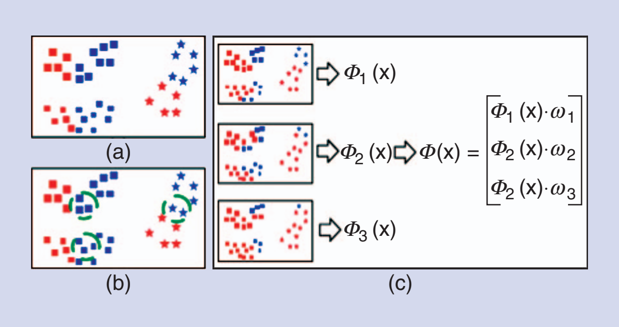
where y_{ic} is the indicator variable for $y_{ic} = c$. The right-hand side of the constraint is simply the prior probability of the class c, estimated from the source domain.

The geodesic flow kernel computed between the source $\mathcal{S}$ and the target $\mathcal{T}_u$ is used to compose the kernel mapping function $\phi(\mathbf{x})$ [70]

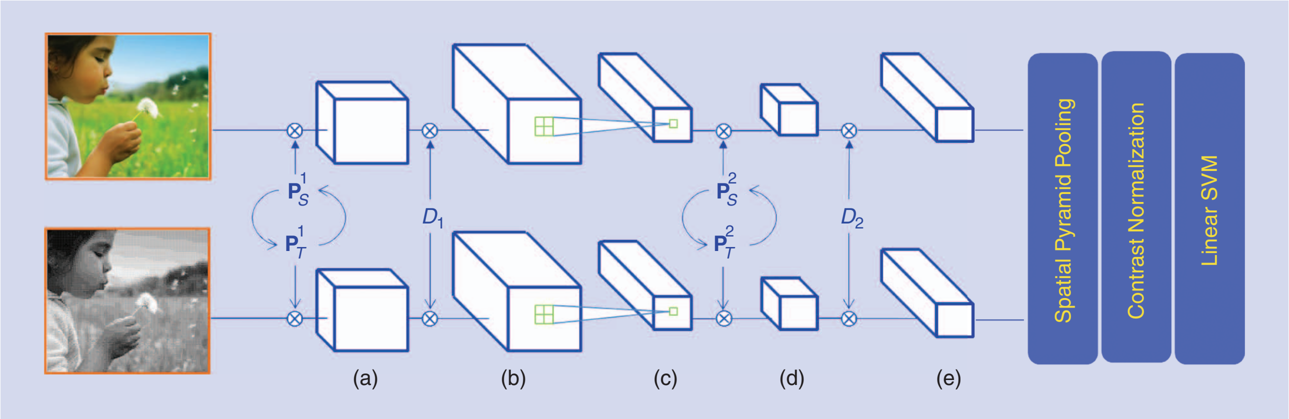
$$\phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j) = K(\mathbf{x}_i, \mathbf{x}_j)$$
$$= \exp\{-(\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{G} (\mathbf{x}_i - \mathbf{x}_j)/\sigma^2\}, \qquad (19)$$

where $\mathbf{G}$ is computed using the singular value decomposition of $\mathbf{S}_1^T \mathbf{S}_2$. Here, $\mathbf{S}_1$ and $\mathbf{S}_2$ are the matrices obtained by applying PCA on $\mathbf{S}$ and $\mathbf{T}_u$, respectively [37].

A set of factors $\{\sigma_i \in [\sigma_{\min}, \sigma_{\max}]\}_{i=1}^Q$ is used to select the scale factor σ in (19). For each σ_i, (18) is solved to obtain the corresponding landmarks $\mathcal{L}^i$ whose α_i is equal to one. For each set of landmarks, a new domain pair is constructed by moving the landmarks from the original source to the target domains. It was argued that each auxiliary task is easier to adapt than the original pair $\mathcal{S}$ and $\mathcal{T}_u$ [70].



[FIG5] An overview of the landmark-based method proposed in [70]. (a) The original domain adaptation problem where the instances in red are from the target and those in blue are from the source. (b) Landmarks, shown inside the green circles, are data instances from the source that can be regarded as samples from the target. (c) Multiple auxiliary tasks are created by augmenting the original target with landmarks, which switches their color from blue to red. Each task gives rise to a new feature representation. These representations are combined discriminatively to form domain-invariant features for the original domain adaptation problem [70].



[FIG6] **An illustration of domain adaptation using a sparse and hierarchical network (DASH-N) algorithm [77]. The source domain is RGB images, and the target domain is half-tone images. First, the images are divided into small overlapping patches. These patches are vectorized while maintaining their spatial arrangements. (a) Performing contrast normalization and dimensionality reduction using P_S for source images and P_T for target images. The circular feedbacks between P_S and P_T indicate that these two transformations are learned jointly. (b) Obtaining sparse codes using the common dictionary D_1. (c) Performing max pooling. The process then repeats for (d) and (e) layer 2, except that the input is the sparse codes from layer 1 instead of pixel intensities. At the final stage, spatial pyramids with max pooling are used to create image descriptors. Classification is done using a linear SVM.**

The final kernel is then learned as a convex combination of all of the kernels from the auxiliary tasks

$$\mathrm{F} = \sum_i \beta_i \mathrm{G}_i \quad \text{s.t. } \beta_i \geq 0 \text{ and } \sum_i \beta_i = 1. \tag{20}$$

The coefficients β_i are optimized on a labeled training set $\sum_i \mathcal{L}^i$ composed of all landmarks selected at different granularities. Finally, F is used in an SVM classifier whose accuracy is optimized with the standard MKL algorithm to learn β_i [70], [71]. Since $\sum_i \mathcal{L}^i$ consists of landmarks that are distributed similarly to the target, it is expected that the classification error on $\sum_i \mathcal{L}^i$ will be a good proxy to that of the target domain [70].

OTHER METHODS

Deep neural networks have had tremendous success achieving a state-of-the-art performance on a number of machine-learning and computer vision tasks [72]. This is due in part to the fact that deep networks are able to learn extremely powerful hierarchical nonlinear representations of the inputs [73], [74]. Motivated by recent works on deep learning, several hierarchical domain adaptation approaches have been proposed in the literature [75]–[79].

In [78], multiple intermediate representations are explored along an interpolating path between the target and source domains. Starting with all the source data samples $\mathcal{S}$, intermediate sampled data sets are generated. For each successive data set, the proportion of samples randomly drawn from $\mathcal{T}$ is increased and the proportion of samples drawn from $\mathcal{S}$ is decreased. Let $i \in [1, \cdots, k]$ be an index set over k intermediate data sets. Then, $\mathcal{S}_i = \mathcal{S}$ for $i = 1$, $\mathcal{S}_i = \mathcal{T}$, for $i = k$. For $i \in [2, \cdots, k-1]$, data sets $\mathcal{S}_i$ and $\mathcal{S}_{i+1}$ are created in a way so that the proportion of samples from $\mathcal{T}$ in $\mathcal{S}_i$ is less than in $\mathcal{S}_{i+1}$. Each of these data sets can be thought of as a single point on a particular kind of interpolating path between $\mathcal{S}$ and $\mathcal{T}$.

For each intermediate data set $\mathcal{S}_i$, a deep nonlinear feature extractor is trained. Once feature extractors corresponding to all points on the path are trained, any input sample can be represented by concatenating all of the outputs from the feature extractors together to create path features for the input. The hope is that this path representation will be more effective at domain adaptation because it is constructed to capture information about incremental changes between the source and target domains similar to [33] and [37]. After creating the path representation of the inputs, a classifier is trained on the data generated from the source domain data by minimizing an appropriate loss function [78].

Another recent work for visual domain adaptation using hierarchical networks was recently proposed by Nguyen et al. [77]. Their method jointly learns a hierarchy of features together with transformations that address the mismatch between different domains. This method was motivated by [80] in which multilayer sparse coding networks are proposed for building feature hierarchies layer by layer using sparse codes and spatial pooling. Figure 6 shows an overview of the sparse hierarchical domain adaptation method [77]. The network contains multiple layers, each of which contains three sublayers. The first sublayer performs contrast normalization and dimensionality reduction on the input data. Sparse coding is carried out in the second sublayer. In the final sublayer, adjacent features are max-pooled together to produce a new feature. The output from one layer becomes the input to the next layer. This method can be viewed as a generalization of the domain-adaptive dictionary learning framework [66] using hierarchical networks. An extension of this method to multiple source domains has also been presented in [77].

Visual attributes are human understandable properties to describe images such as blue, dark, and two-legged. They are valuable as a semantic cue in various vision problems. Recent research explores a variety of applications for visual attributes including face verification [81], object recognition [82]–[84], and facilitating transfer learning [85]. The existing methods [82], [84], [85] assume that one model of an attribute is sufficient to capture all user perceptions. However, there are some real perceptual differences between annotators. Consider the example shown in Figure 7: five users confidently declared that the shoe on the left is formal, while five confidently declared the opposite. These differences stem from several factors such as the words for attributes are imprecise, their meaning often depends on context and culture, and they often stretch to refer to quite distinct object categories [86].

VISUAL ATTRIBUTES ARE HUMAN UNDERSTANDABLE PROPERTIES TO DESCRIBE IMAGES SUCH AS BLUE, DARK, AND TWO-LEGGED.

To capture the inherent differences in perception, [86] proposes to model attributes in a user-specific way. In particular, attribute learning is posed as an adaptation problem. First, they leverage any commonalities in perception to learn a generic prediction function using a large margin learning algorithm and data labeled with a majority vote from multiple annotators. Then, they use a small number of user-labeled examples to adapt the parameters of the generic model into a user-specific prediction function while not straying too far from the prior generic model. Essentially, this amounts to imposing regularizers on the learning objective favoring user-specific model parameters that are similar to the generic ones while still satisfying the user-specific label constraints [86]. The impact of this attribute adaptation work is that one can capture a user's perception with minimal annotation effort. It was shown that the resulting personalization can make attribute-based image searches more accurate [86].

Tommasi and Caputo [87] very recently proposed a naive Bayes' nearest neighbor-based domain adaptation method that iteratively learns a Mahalanobis class-specific metric while inducing for each sample a large margin separation among classes. Both semisupervised and unsupervised domain adaptation scenarios are presented.

In [88], Jain and Farfade proposed an approach for adapting a cascade of classifiers to perform classification in a similar domain for which only a few positive examples are available. A cascade of classifiers is a classifier f that is composed of m stage classifiers $\{f_1, \cdots, f_m\}$ that are applied in a sequential manner. They are commonly used for anomaly detection and one-class classification. It was shown that, by adapting classification cascades to new domains, one can obtain huge gains in performance in detecting faces of human babies and human-like characters from movies.

APPLICATIONS

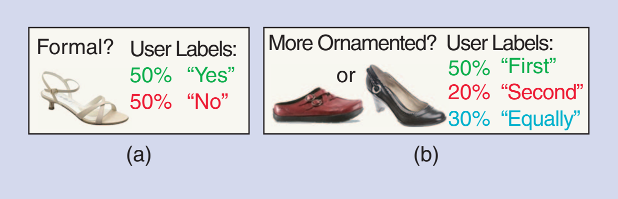
In this section, we illustrate through different application examples the uses and capabilities of various visual domain adaptation methods. In particular, we focus on object recognition and face recognition applications.

FACE RECOGNITION

Face recognition is a challenging problem that has been actively researched for more than two decades [89]. The current systems work very well when training, and test images are captured under controlled conditions. However, their performance degrades significantly when the test images contain variations that are not present in the training images. One of these variations is change in pose. Along with the frontal images with different illumination (source images), if we are also given a few images at different poses (target images), then the resulting face recognition problem can be viewed a domain adaptation problem [65], [66], [90].

Face recognition experiments were conducted on the Carnegie Mellon University (CMU) multipose, illumination, and expression (PIE) data set [91] with images of 129 subjects in a frontal pose as the source domain, and five other off-frontal poses as the target domain. Images under five illumination conditions across source and target domains were used for training with which images from the remaining 15 illumination conditions in the target domain were recognized. The results provided in Table 1 show that the dictionary-based adaptation method [66] compares favorably with some of the recently proposed multiview recognition algorithms [10] as well as many other nonadaptation techniques and gives the best performance on average. Note that the discriminative dictionary-learning algorithm, Fisher discrimination dictionary learning (FDDL) [92], does not provide the best results here as it is not able to efficiently represent the nonlinear changes introduced by the pose variation.

Furthermore, the learned dictionaries were also used for pose alignment where the goal is to align faces from one pose to a different pose. This is a challenging problem since actual pose variations

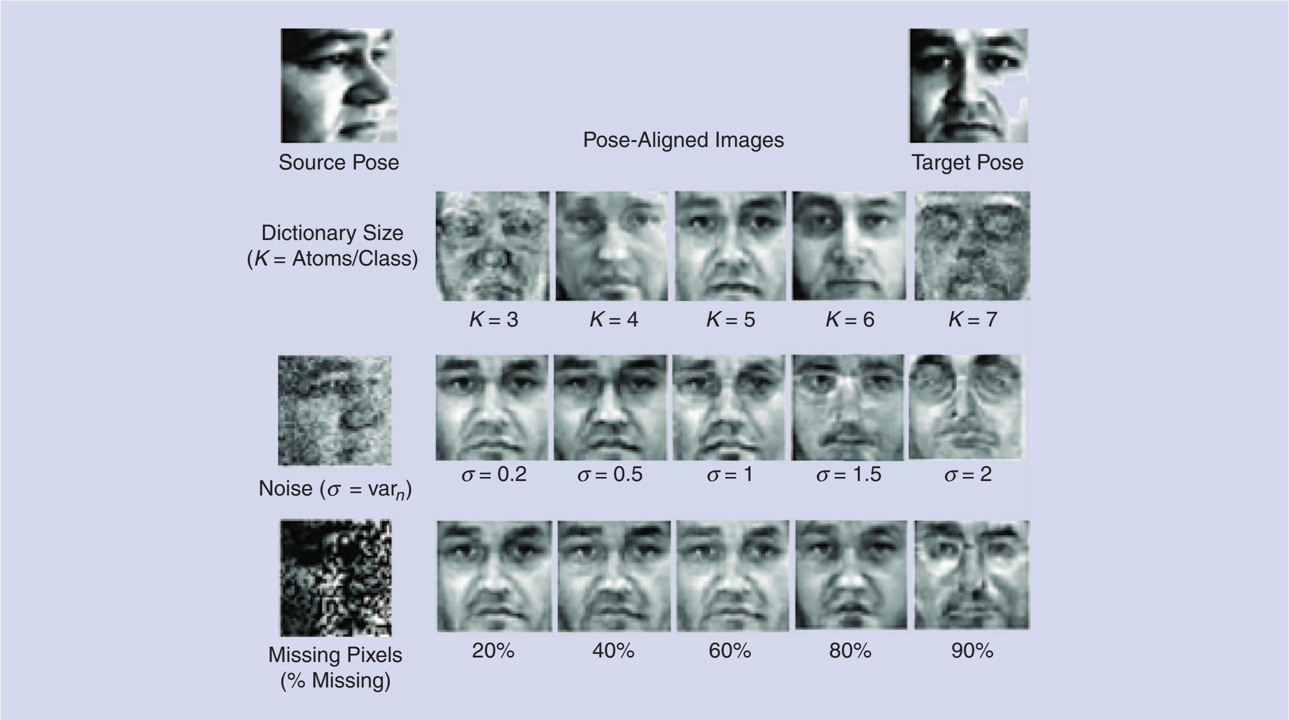


[FIG7] Virtual attribute interpretations vary slightly from viewer to viewer. For instance, five viewers confidently declare the shoe as (a) formal or (b) more ornamented, while five others confidently declare the opposite. Attribute adaptation models are proposed to take these differences in perception into account [86].

[TABLE 1] A COMPARISON OF VARIOUS ALGORITHMS FOR FACE RECOGNITION ACROSS POSE [66].

METHOD	PROBE POSE					AVERAGE
	15°	30°	45°	60°	75°	
PCA	15.3	5.3	6.5	3.6	2.6	6.7
PLS [27]	39.3	40.5	41.6	41.1	38.7	40.2
LDA	98	94.2	91.7	84.9	79	89.5
CCA [27]	92.1	89.7	88	86.1	83	83.5
GMLDA [10]	**99.7**	**99.2**	98.6	94.9	95.4	97.6
FDDL [92]	96.8	90.6	94.4	91.4	90.5	92.7
SDDL [66]	98.4	98.2	**98.9**	**99.1**	**98.8**	**98.7**

Boldface indicates the top performing algorithm in each experiment.
LDA: Linear discriminant analysis; GMLDA: generalized multiview linear discriminant analysis; and SDDL: shared domain-adapted dictionary learning.



[FIG8] Examples of pose-aligned images. Synthesis in various conditions demonstrate the robustness of the domain-adaptive dictionary-learning method [66].

are three dimensional (3-D), whereas the image evidence one has is two dimensional (2-D). Sample results are shown in Figure 8. One of the interesting features of the dictionary-based adaptation methods is that they allow the synthesis of data associated with different domains while exploiting the generative power of dictionary-based representations. This is essentially what is highlighted in the last two rows of Figure 8. The dictionary-based method is robust at high levels of noise and missing pixels. It produces denoised and inpainted synthesized images. Additional results on various face recognition tasks using domain adaptation can be found in [65] and [67].

OBJECT RECOGNITION

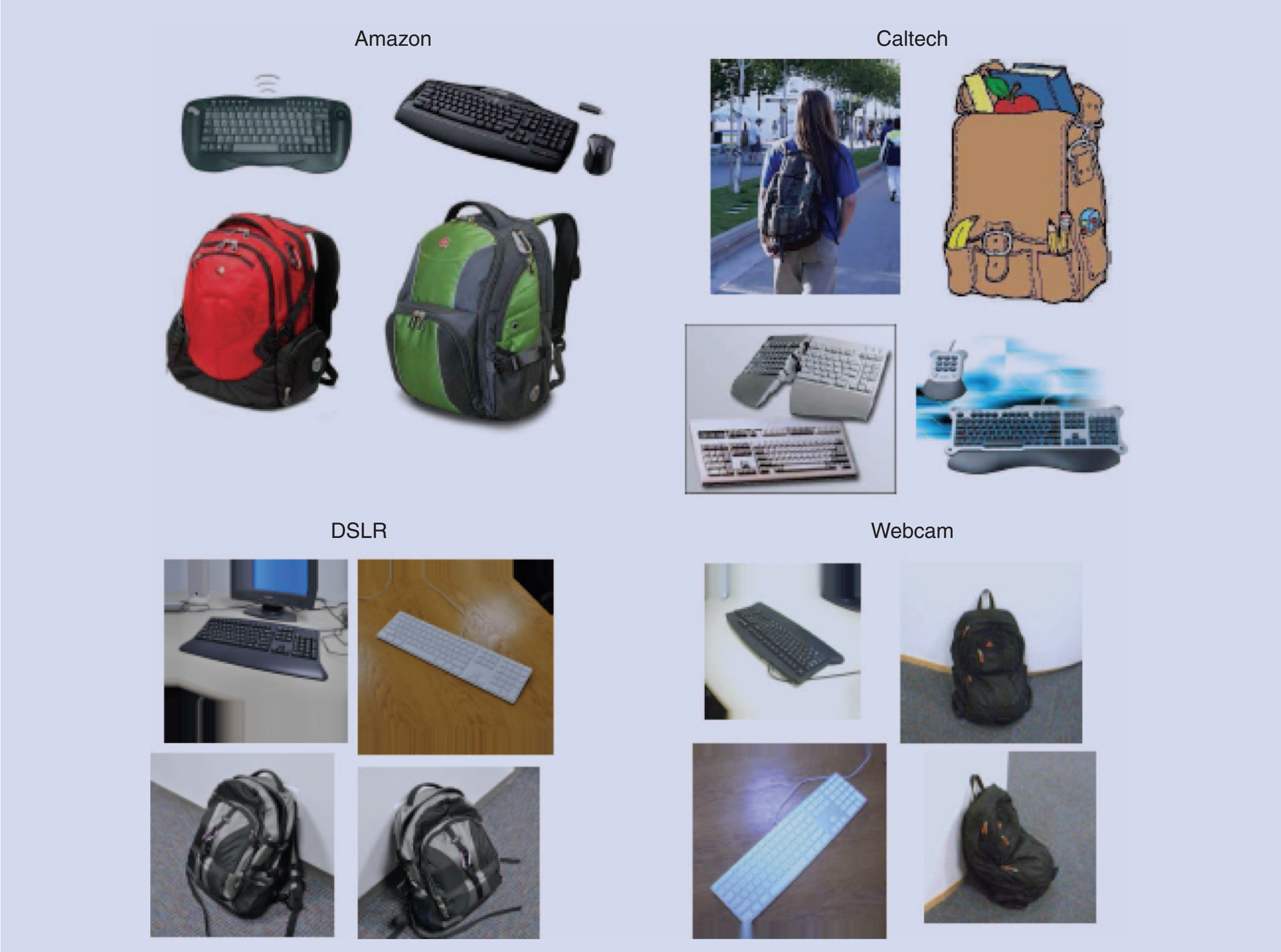
In this section, we compare the performance of various visual domain adaptation methods on a benchmark object recognition data set that was introduced in [41]. The data set consists of images from three sources: Amazon (consumer images from online merchant sites), digital single-lens reflex [(DSLR) images from a DSLR camera], and Webcam (low-quality images from Webcams). In addition, algorithms are tested on the Caltech-256 data set [93], taking it as the fourth domain. Figure 9 shows sample images from these data sets and clearly highlights the differences between them.

Three setups are followed for comparing the performance of various algorithms. In the first setup, ten classes: "Backpack," "Touring Bike," "Calculator," "Headphones," "Computer Keyboard," "Laptop 101," "Computer Monitor," "Computer Mouse," "Coffee Mug," and "Video Projector" common to all the four data sets are used. In this case, there are a total of 2,533 images. Each category has eight to 151 images in a data set. In the second setup, all 31 classes from Amazon, Webcam, and DSLR are used to evaluate various algorithms. Finally, in the third setup, methods for adaptation are evaluated using multiple domains. In this case, the first data set is used, and the methods are tested on all 31 classes in it. For both cases, we use 20 training samples per class for Amazon/Caltech, eight samples per class for DSLR/Webcam when used as source, and three training samples for all of them when used for the target domain. The rest of the data in the target domain is used for testing. The experiment is run multiple times for random train/test splits, and the result is averaged over all the runs. For the unsupervised case, the same setting as semisupervised adaptation described earlier is followed but without using any labeled data from the target domain. (Several recent methods explore both source and target data at once in a transductive manner rather than splitting the data sets into multiple training/testing partitions; see [70] for details on the evaluation protocol using this setting.)

SEMISUPERVISED ADAPTATION RESULTS USING A SINGLE SOURCE

The semisupervised adaptation recognition results of different algorithms on eight pairs of source–target domains and on all 31 classes are shown in Tables 2 and 3, respectively. The baseline results obtained using the hierarchical matching pursuit (HMP) method [80] as well as the FDDL method [92], which learn the dictionaries separately for the source and target domains without performing domain adaptation, are also included.

Compared to the metric-learning-based approach [41], manifold-based feature concatenation methods [33], [37] provide better results. This makes sense because, by finding intermediate domain



[FIG9] Some example images from the "Keyboard" and "Backpack" categories in Caltech-256, Amazon, Webcam, and DSLR. The Caltech-256 and Amazon data sets have diverse images, and Webcam and DSLR are similar data sets with images mostly from offices [66].

[TABLE 2] THE SEMISUPERVISED DOMAIN ADAPTATION RESULTS OF DIFFERENT APPROACHES ON FOUR DOMAINS WITH TEN COMMON CLASSES (C: CALTECH, A: AMAZON, D: DSLR, W: WEBCAM).

METHODS	C → A	C → D	A → C	A → W	W → C	W → A	D → A	D → W
METRIC [41]	33.7 ± 0.8	35 ± 1.1	27.3 ± 0.7	36 ± 1	21.7 ± 0.5	32.3 ± 0.8	30.3 ± 0.8	55.6 ± 0.7
SGF [33]	40.2 ± 0.7	36.6 ± 0.8	37.7 ± 0.5	37.9 ± 0.7	29.2 ± 0.7	38.2 ± 0.6	39.2 ± 0.7	69.5 ± 0.9
GFK [37]	46.1 ± 0.6	55 ± 0.9	39.6 ± 0.4	56.9 ± 1	32.8 ± 0.1	46.2 ± 0.6	46.2 ± 0.6	**80.2 ± 0.4**
FDDL [92]	39.3 ± 2.9	55 ± 2.8	24.3 ± 2.2	50.4 ± 3.5	22.9 ± 2.6	41.1 ± 2.6	36.7 ± 2.5	65.9 ± 4.9
HMP [80]	67.7 ± 2.3	70.2 ± 5.1	51.7 ± 4.3	70 ± 4.2	46.8 ± 2.1	61.5 ± 3.8	64.7 ± 2	76.0 ± 4
SDDL [66]	49.5 ± 2.6	76.7 ± 3.9	27.4 ± 2.4	72 ± 4.8	29.7 ± 1.9	49.4 ± 2.1	48.9 ± 3.8	72.6 ± 2.1
DASH-N [77]	**71.6 ± 2.2**	**81.4 ± 3.5**	**54.9 ± 1.8**	**75.5 ± 4.2**	**50.2 ± 3.3**	**70.4 ± 3.2**	**68.9 ± 2.9**	77.1 ± 2.8

Boldface indicates the top-performing algorithm in each experiment.
SGF: Subspaces by sampling geodesic flow; GFK: geodesic flow kernel.

representations, one is able to learn a feature vector that is more robust than a feature vector that results by learning a single transformation that minimizes the effect of the domain shift. The SDDL method can be viewed as an extension of the FDDL method, which simultaneously learns discriminative dictionaries on a latent space where both the source and the target data are forced to have similar sparse representation. As a result, one can clearly see the performance gain of the SDDL method over the FDDL method as well as the manifold-based methods in Tables 2 and 3.

The HMP method [80] builds a feature hierarchy layer by layer using an efficient matching pursuit encoder. It consists of three

[TABLE 3] SINGLE-SOURCE SEMISUPERVISED DOMAIN ADAPTATION RESULTS ON ALL 31 CLASSES.

METHOD	A → W	D → W	W → D
METRIC [41]	44	31	27
RDALR [44]	50.7 ± 0.8	36.9 ± 19.9	32.9 ± 1.2
SGF [33]	57 ± 3.5	36 ± 1.1	37 ± 2.3
GFK [37]	46.4 ± 0.5	61.3 ± 0.4	66.3 ± 0.4
HMP [80]	55.7 ± 2.5	50.5 ± 2.7	56.8 ± 2.6
SDDL [66]	50.1 ± 2.5	51.2 ± 2.1	50.6 ± 2.6
DASH-N [77]	**60.6 ± 3.5**	**67.9 ± 1.1**	**71.1 ± 1.7**

Boldface indicates the top performing algorithm in each experiment.
RDALR: Robust domain adaptation with low-rank reconstruction.

[TABLE 4] MULTIPLE-SOURCE DOMAIN ADAPTATION RESULTS OF VARIOUS METHODS ON THE AMAZON, WEBCAM, AND DSLR DATA SETS.

SOURCE	TARGET	SGF [34]	SGF [33]	RDALR [44]	FDDL [92]	SDDL [66]	A-SVM [29]	HMP [80]	DASH-N [77]
DSLR, AMAZON	WEBCAM	**64.5 ± 0.3**	52 ± 2.5	36.9 ± 1.1	41 ± 2.4	57.8 ± 2.4	30.4 ± 0.6	47.2 ± 1.9	**64.5 ± 2.3**
AMAZON, WEBCAM	DSLR	51.3 ± 0.7	39 ± 1.1	31.2 ± 1.3	38.4 ± 3.4	56.7 ± 2.3	25.3 ± 1.1	51.3 ± 1.4	**68.6 ± 3.7**
WEBCAM, DSLR	AMAZON	38.4 ± 1.0	28 ± 0.8	20.9 ± 0.9	19 ± 1.2	24.1 ± 1.6	17.3 ± 0.9	37.3 ± 1.4	**41.8 ± 1.1**

main components: batch tree orthogonal matching pursuit, spatial pyramid matching, and contrast normalization. As a result, it is robust to some of the variations present in the images such as illumination changes, pose variations, and resolution variations. The DASH-N method essentially extends the SDDL and HMP methods by learning features directly from data for domain adaptation. As a result, it provides a more robust and discriminative representation of the data and performs the best on this data set on both settings. The dictionary learning-based methods [92], [80] essentially find the common internal structure of the data. They inherently have the denoising capability and provide robust representation of the data. This is one of the reasons why in some cases the FDDL and the HMP methods provide better results than metric-learning- and manifold-based methods.

SEMISUPERVISED ADAPTATION RESULTS USING MULTIPLE SOURCES

As some of the methods reviewed in this paper can also handle multiple domains, we report results of different algorithms on multiple-source adaption. Table 4 shows the results for three possible combinations. Again, the sparse hierarchical network-based adaptation method [77] performs the best. The incremental learning motivated manifold method [34] also provides good results on multidomain adaptation using this data set. It is interesting to see that increasing the number of domains can be helpful, especially when compared to a single source and single target. Many multidomain adaptation methods in Table 4 outperform a single source and a single target in many cases, although, in a small number of cases, they do not outperform a single source and a single target. As a result, a better strategy to deal with multiple domains is required in these cases.

UNSUPERVISED DOMAIN ADAPTATION RESULTS

The results of three source–target combinations of the Amazon, DSLR, and Webcam data sets are shown in Table 5. The manifold-based approach [34] outperforms the existing unsupervised domain adaptation methods in two of the three source–target combinations. The information–theoretic learning method [59] for unsupervised domain adaptation also performs well on this data set. By comparing the results in Tables 2 and 3 with the results in Table 5, we see that the semisupervised adaptation results are generally better than in the unsupervised case. Using labels in both the intermediate data generation and classification stage generally produces better results than using labels only during classification [34]. Also, it is interesting to see that, since the introduction of this data set in [41], the recognition performance has significantly improved in the last few years.

COMPUTATIONAL COMPLEXITY

The main processing steps involved in manifold-based adaptation techniques [33], [34], [37] are computing the geodesic between the source and target domains, and then sampling points along the geodesic to infer intermediate domains that account for the domain shift. This involves mapping entities on the manifold to the locally Euclidean tangent plane and warping the results from the tangent plane back onto the manifold. Computationally efficient algorithms for these steps have been discussed in the literature for Grassmann manifolds [94]. For orthogonal matrices of dimensions $N_1 \times N_2$, the geodesic computation has a complexity of $O(N_1^2 N_2)$ along with an $O(N_1 N_2)$ cost for sampling each point along the geodesic.

For deep learning approaches [77], [78], the complexity depends, among others, on the number of layers used in the hierarchy to learn feature correlation for adaptation. While the deep network circuits can have different architectures, such as autoencoders and restricted Boltzmann machines, there is an active stream of work in making the training procedure of these circuits computationally tractable. See [72] for a more detailed discussion on the complexity of deep architectures.

A major computationally heavy step of dictionary-based domain adaptation methods is dominated by sparse coding. Efficient batch methods have been proposed to learn dictionaries for large-scale problems. For instance, a batch orthogonal matching pursuit-based KSVD algorithm for learning dictionaries was proposed in [95]. It was shown that the operation count per training iteration for learning a dictionary of size $l \times K$ with R number of training signals are $R(T_0^2 K + 2lK)$, where T_0 is the target sparsity. One can also adapt fast ℓ_1 solvers for sparse coding [96], [97] rather than using greedy orthogonal matching pursuit algorithms.

For the low-rank approximation-based methods, the major computation is in finding the SVD of a matrix. As a result, these methods tend to be time-consuming if the matrix is large. However, efficient methods do exist for finding the low-rank approximation of large matrices [98]–[100].

[TABLE 5] UNSUPERVISED DOMAIN ADAPTATION RESULTS OF VARIOUS METHODS ON THE AMAZON, WEBCAM, AND DSLR DATA SETS.

SOURCE	TARGET	SGF [34]	SGF [33]	RDALR [44]	GFK [37]	ITLUDA [59]
WEBCAM	DSLR	**71.2**	19 ± 1.2	32.89 ± 1.2	49.7 ± 0.5	—
DSLR	WEBCAM	68.8	26 ± 0.8	36.85 ± 1.9	44.6 ± 0.3	**83.6 ± 0.5**
AMAZON	WEBCAM	**55.6**	39 ± 2	50.71 ± 0.8	15 ± 0.4	38.5 ± 1.3

Many parameter adaptation methods such as A-SVM [29] are large-scale quadratic programming problems for which efficient implementations do exist in the literature (see [101] for more details).

CONCLUSIONS AND FUTURE DIRECTIONS

This article attempted to provide an overview of recent developments in domain adaptation for computer vision, with an emphasis on applications to the problems of face and object recognition. We believe that the availability of massive data has brought substantial opportunities and challenges to the analysis of data sets bias or covariant shifts and domain adaptation problems. We hope that the survey has helped to guide interested readers among the extensive literature to some degree, but obviously it cannot cover all of the literature on domain adaptation, and we have chosen to focus on a representative subset of the latest progress made in computer vision.

DOMAIN ADAPTATION PROMISES TO BE AN ACTIVE AREA OF RESEARCH, ESPECIALLY AS ONE OF THE POSSIBLE WAYS TO QUICKLY PROPAGATE SEMANTIC ANNOTATIONS TO THE LARGE-SCALE VISUAL DATA BEING ACQUIRED EVERY MINUTE.

Domain adaptation promises to be an active area of research, especially as one of the possible ways to quickly propagate semantic annotations to the large-scale visual data being acquired every minute. In computer vision, researchers have identified specific challenges that do not belong to machine learning: a major question among them that is rarely addressed in traditional domain adaptation research is one of adapting structured (nonvector) data representations. In machine learning or natural language processing, an input sample is usually represented as a vector in Euclidean space, different samples are treated as independent observations, and the task is typically classification. This is, however, not the case in computer vision where the representations to be potentially adapted include shapes and contours, deformable and articulated 2-D or 3-D objects, graphs, and random fields, intrinsic images, as well as visual dynamics, none of which is directly supported by vectorial domain adaptation techniques. In addition to recognition and detection, models and algorithms for segmentation, reconstruction, and tracking are awaiting mechanisms that do not yet exist to be adapted toward emerging new domains. All of these challenges necessitate continuous efforts on characterizing visual domain shift and a paradigm of effective and efficient adaptation methods that are dedicated to visual data.

In the meantime, it is generally accepted that domain shifts in computer vision are usually due to causes from the imaging process that can be explained physically, such as illumination changes, sensor changes, and viewpoint changes. We believe that incorporating these physical priors into strong statistical adaptation approaches will not only lead to a performance increase but also to other insights in understanding the imaging process. This calls for a physically informed adaptation paradigm that better exploits knowledge about image formation and better integrates other domain-specific knowledge implied by the diverse set of partial, noisy, and multimodal side information accompanying the visual data, such as imagery obtained from online social media. We hope that by appropriately incorporating a physically informed adaptation paradigm, distributional changes across different sensors (electro-optical/synthetic aperture radar, infrared/synthetic aperture radar, electro-optical/infrared, etc.) can be handled.

Finally, we expect that studies on data characteristics and adaptations will produce stronger guidance to developing more desirable data sets for evaluating research in a wider spectrum of computer vision problems.

ACKNOWLEDGMENT

This work was partially supported by a multidisciplinary university research initiative from the Office of Naval Research under grant 1141221258513.

AUTHORS

Vishal M. Patel (pvishalm@umd.edu) received his B.S. degrees in electrical engineering and applied mathematics (with honors) and his M.S. degree in applied mathematics from North Carolina State University, Raleigh, in 2004 and 2005, respectively. He received his Ph.D. degree in electrical engineering from the University of Maryland, College Park, in 2010. He is a member of the research faculty at the University of Maryland Institute for Advanced Computer Studies. His research interests are in signal processing, computer vision, and machine learning with applications to imaging and biometrics. He was a recipient of the Oak Ridge Associated Universities postdoctoral fellowship in 2010. He is also a member of Eta Kappa Nu, Pi Mu Epsilon, and Phi Beta Kappa. He is a Member of the IEEE.

Raghuraman Gopalan (raghuram@research.att.com) received his Ph.D. degree in electrical and computer engineering from the University of Maryland, College Park, in 2011. He is a senior member of technical staff at AT&T Labs–Research. His research interests are in computer vision and machine learning with a focus on object recognition problems. He is a Member of the IEEE.

Ruonan Li (ruonanli@seas.harvard.edu) received his B.E. and M.E. degrees from Tsinghua University, Beijing, China. He received the Ph.D. degree in electrical and computer engineering from the University of Maryland, College Park, in 2011. He is currently a research associate at Harvard University, Cambridge, Massachusetts. His research interests include general problems in computer vision, image processing, pattern recognition, and machine learning with recent focuses on video analysis and video-based recognition, socialized visual analytics, cross-domain model adaptation, and the application of differential geometric methods to the related problems.

Rama Chellappa (rama@umiacs.umd.edu) is a Minta Martin Professor of Engineering and the chair of the Electronics and Communication Engineering Department at the University of Maryland (UMD). He is a recipient of the K.S. Fu Prize from the International Association of Pattern Recognition, the Society, Technical Achievement and Meritorious Service Awards from the

IEEE Signal Processing Society, and the Technical Achievement and Meritorious Service Awards from the IEEE Computer Society. At UMD, he received college- and university-level recognition for research, teaching, innovation, and mentoring of undergraduate students. He is a fellow of the International Association of Pattern Recognition, the Optical Society of America, the American Association for the Advancement of Science, and the Association for Computing Machinery and holds four patents. He is a Fellow of the IEEE.

REFERENCES

[1] H. Shimodaira, "Improving predictive inference under covariate shift by weighting the log-likelihood function," *J. Stat. Plan. Inference*, vol. 90, no. 2, pp. 227–244, 2000.

[2] A. Torralba and A. A. Efros, "Unbiased look at data set bias," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2011, pp. 1521–1528.

[3] A. Khosla, T. Zhou, T. Malisiewicz, A. A. Efros, and A. Torralba, "Undoing the damage of data set bias," in *Proc. European Conf. Computer Vision*, 2012, pp. 158–171.

[4] N. Japkowicz and S. Stephen, "The class imbalance problem: A systematic study," *Intell. Data Anal.*, vol. 6, no. 5, pp. 429–450, 2002.

[5] J. J. Heckman, "Sample selection bias as a specification error," *Econometric*, vol. 47, no. 1, pp. 153–161, 1979.

[6] B. Zadrozny, "Learning and evaluating classifiers under sample selection bias," in *Proc. Int. Conf. Machine Learning*, 2004, pp. 114–121.

[7] R. Caruana, "Multitask learning," *Machine Learning*, vol. 28, no. 1, pp. 41–75, 1997.

[8] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng, "Self-taught learning: Transfer learning from unlabeled data," in *Proc. Int. Conf. Machine Learning*, 2007, pp. 759–766.

[9] O. Chapelle, B. Schölkopf, and A. Zien, *Semi-Supervised Learning*. Cambridge, MA: MIT Press, 2006.

[10] A. Sharma, A. Kumar, H. Daume, and D. Jacobs, "Generalized multiview analysis: A discriminative latent space," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2012, pp. 2160–2167.

[11] J. Jiang, "Domain adaptation in natural language processing," Ph.D. dissertation, University of Illinois at Urbana-Champaign, 2008.

[12] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York, NY: Springer-Verlag, 1995.

[13] M. Sugiyama and K. R. Muller, "Input-dependent estimation of generalization error under covariate shift," *Stat. Decisions*, vol. 23, no. 4, pp. 249–279, 2005.

[14] J. Huang, A. J. Smola, A. Gretton, K. M. Borgwardt, and B. Schlkopf, "Correcting sample selection bias by unlabeled data," in *Advances in Neural Information Processing Systems 19*, B. Schölkopf, J. Platt, and T. Hoffman, Eds. Cambridge, MA: MIT Press, 2007, pp. 601–608.

[15] Y. Lin, Y. Lee, and G. Wahba, "Support vector machines for classification in nonstandard situations," *Mach. Learn.*, vol. 46, nos. 1–3, pp. 191–202, 2002.

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

[17] W. Dai, G. Rong Xue, Q. Yang, and Y. Yu, "Transferring naive Bayes classifiers for text classification," in *Proc. AAAI Conf. Artificial Intelligence*, 2007, pp. 540–545.

[18] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell, "Text classification from labeled and unlabeled documents using EM," *Mach. Learn.*, vol. 39, nos. 2–3, pp. 103–134, May 2000.

[19] D. Xing, W. Dai, G.-R. Xue, and Y. Yu, "Bridged refinement for transfer learning," in *Proc. European Conf. Principles and Practice of Knowledge Discovery in Databases*, 2007, pp. 324–335.

[20] J. Jiang and C. Zhai, "Instance weighting for domain adaptation in NLP," in *Proc. Annu. Meeting of the Association for Computational Linguistics*, 2007, pp. 264–271.

[21] R. Raina, "Self-taught learning," Ph.D. dissertation, Stanford University, 2009.

[22] H. Wang, F. Nie, and H. Huang, "Robust and discriminative self-taught learning," in *Proc. Int. Conf. Machine Learning*, vol. 28, no. 3, 2013, pp. 298–306.

[23] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, "Self-taught clustering," in *Proc. Int. Conf. Machine Learning, ACM*, 2008, pp. 200–207.

[24] Y. Jia, M. Salzmann, and T. Darrell, "Factorized latent spaces with structured sparsity," in *Advances in Neural Information Processing Systems*, J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, Eds. Curran Associates, Inc., 2010, pp. 982–990.

[25] N. Rasiwasia, J. Costa Pereira, E. Coviello, G. Doyle, G. R. Lanckriet, R. Levy, and N. Vasconcelos, "A new approach to cross-modal multimedia retrieval," in *Proc. Int. Conf. Multimedia*, ACM, 2010, pp. 251–260.

[26] K. Wang, R. He, W. Wang, L. Wang, and T. Tan, "Learning coupled feature spaces for cross-modal matching," in *Proc. Int. Conf. Computer Vision*, 2013, pp. 2088–2095.

[27] A. Sharma and D. Jacobs, "Bypassing synthesis: PLS for face recognition with pose, low-resolution and sketch," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2011, pp. 593–600.

[28] Y. Jia, M. Salzmann, and T. Darrell, "Learning cross-modality similarity for multinomial data," in *Proc. IEEE Int. Conf. Computer Vision*, 2011, pp. 2407–2414.

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

[30] L. Duan, I. W. Tsang, D. Xu, and T.-S. Chua, "Domain adaptation from multiple sources via auxiliary classifiers," in *Proc. Int. Conf. Machine Learning*, 2009, pp. 289–296.

[31] H. Daumé III, "Frustratingly easy domain adaptation," in *Proc. Conf. Association for Computational Linguistics*, 2007, p. 256.

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

[33] R. Gopalan, R. Li, and R. Chellappa, "Domain adaptation for object recognition: An unsupervised approach," in *Proc. IEEE Int. Conf. Computer Vision*, 2011, pp. 999–1006.

[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] J. Hoffman, B. Kulis, T. Darrell, and K. Saenko, "Discovering latent domains for multisource domain adaptation," in *Proc. European Conf. Computer Vision*, 2012, pp. 702–715.

[36] B. Gong, K. Grauman, and F. Sha, "Reshaping visual data sets for domain adaptation," in *Proc. Neural Information Processing Systems*, 2013, pp. 1286–1294.

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

[38] B. Gong, K. Grauman, and F. Sha, "Learning kernels for unsupervised domain adaptation with applications to visual object recognition," *Int. J. Comput. Vis.*, vol. 109, no. 1–2, pp. 3–27, Aug. 2014.

[39] J. Zheng, M.-Y. Liu, R. Chellappa, and P. Phillips, "A Grassmann manifold-based domain adaptation approach," in *Proc. Int. Conf. Pattern Recognition*, 2012, pp. 2095–2099.

[40] A. Shrivastava, S. Shekhar, and V. M. Patel, "Unsupervised domain adaptation using parallel transport on Grasmann manifold," in *Proc. IEEE Winter Conf. Applications of Computer Vision*, 2014, pp. 277–284.

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

[42] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon, "Information-theoretic metric learning," in *Proc. Int. Conf. Machine Learning*, 2007, pp. 209–216.

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

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

[45] M. Baktashmotlagh, M. T. Harandi, B. C. Lovell, and M. Salzmann, "Unsupervised domain adaptation by domain invariant projection," in *Proc. IEEE Int. Conf. Computer Vision*, 2013, pp. 769–776.

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

[47] Y. Aytar and A. Zisserman, "Tabula rasa: Model transfer for object category detection," in *Proc. IEEE Int. Conf. Computer Vision*, 2011, pp. 2252–2259.

[48] W. Jiang, E. Zavesky, S.-F. Chang, and A. Loui, "Cross-domain learning methods for high-level visual concept classification," in *Proc. IEEE Int. Conf. Image Processing*, 2008, pp. 161–164.

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

[50] A. Bergamo and L. Torresani, "Exploiting weakly-labeled web images to improve object classification: A domain adaptation approach," in *Advances in Neural Information Processing Systems*, J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, Eds. Curran Associates, Inc., 2010, pp. 181–189.

[51] L. Bruzzone and M. Marconcini, "Domain adaptation problems: A DASVM classification technique and a circular validation strategy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 5, pp. 770–787, 2010.

[52] L. Duan, D. Xu, and I. Tsang, "Domain adaptation from multiple sources: A domain-dependent regularization approach," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 3, pp. 504–518, 2012.

[53] A. J. Ma, P. C. Yuen, and J. Li, "Domain transfer support vector ranking for person re-identification without target camera label information," in *Proc. IEEE Int. Conf. Computer Vision*, 2013, pp. 3567–3574.

[54] L. Duan, D. Xu, and S.-F. Chang, "Exploiting web images for event recognition in consumer videos: A multiple source domain adaptation approach," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2012, pp. 1338–1345.

[55] M. Gönen and E. Alpaydin, "Multiple kernel learning algorithms," *J. Mach. Learn. Res.*, vol. 12, pp. 2211–2268, July 2011.

[56] L. Duan, D. Xu, I.-H. Tsang, and J. Luo, "Visual event recognition in videos by learning from web data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1667–1680, 2012.

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

[58] Z. Guo and Z. J. Wang, "Cross-domain object recognition via input-output kernel analysis," *IEEE Trans. Image Processing*, vol. 22, no. 8, pp. 3108–3119, Aug. 2013.

[59] Y. Shi and F. Sha, "Information-theoretical learning of discriminative clusters for unsupervised domain adaptation," in *Proc. Int. Conf. Machine Learning*, 2012, pp. 1079–1086.

[60] J. Hoffman, E. Rodner, J. Donahue, T. Darrell, and K. Saenko, "Efficient learning of domain-invariant image representations," in *Proc. Int. Conf. Learning Representations*, 2013, pp. 1–9.

[61] J. Donahue, J. Hoffman, E. Rodner, K. Saenko, and T. Darrell, "Semi-supervised domain adaptation with instance constraints," in *Proc. IEEE Int. Conf. Computer Vision*, 2013, pp. 668–675.

[62] B. A. Olshausen and D. J. Fieldt, "Sparse coding with an overcomplete basis set: A strategy employed by v1," *Vis. Res.*, vol. 37, no. 23, pp. 3311–3325, Dec. 1997.

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

[64] R. Rubinstein, A. M. Bruckstein, and M. Elad, "Dictionaries for sparse representation modeling," *Proc. IEEE*, vol. 98, no. 6, pp. 1045–1057, June 2010.

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

[66] S. Shekhar, V. M. Patel, H. V. Nguyen, and R. Chellappa, "Generalized domain-adaptive dictionaries," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2013, pp. 361–368.

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

[68] J. Zheng, R. Chellappa, and P. J. Phillips, "Sparse embedding-based domain adaptation for object recognition," in *Proc. IEEE Int. Conf. Computer Vision Workshop on Visual Domain Adaptation and Dataset Bias*, 2013, pp. 1–2.

[69] H. V. Nguyen, V. M. Patel, N. M. Nasrabadi, and R. Chellappa, "Sparse embedding: A framework for sparsity promoting dimensionality reduction," in *Proc. European Conf. Computer Vision*, 2012, pp. 414–427.

[70] 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. Machine Learning*, 2013, pp. 222–230.

[71] B. Gong, F. Sha, and K. Grauman, "Overcoming data set bias: An unsupervised domain adaptation approach," in *Proc. Neural Information Processing Systems Workshop on Large Scale Visual Recognition and Retrieval*, 2012, pp. 1–5.

[72] Y. Bengio, "Learning deep architectures for AI," *Found. Trends Mach. Learn.*, vol. 2, no. 1, pp. 1–127, 2009.

[73] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, July 2006.

[74] G. Hinton and R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[75] M. Chen, Z. Xu, and K. Q. Weinberger, "Marginalized denoising autoencoders for domain adaptation," in *Proc. Int. Conf. Machine Learning*, 2012, pp. 767–774.

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

[77] H. Van Nguyen, "Non-linear and sparse representations for multi-modal recognition," Ph.D. dissertation, Univ. Maryland, 2013.

[78] S. Chopra, S. Balakrishnan, and R. Gopalan, "DLID: Deep learning for domain adaptation by interpolating between domains," in *Proc. ICML Workshop on Challenges in Representation Learning*, 2013, pp. 1–8.

[79] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," in *Proc. Int. Conf. Machine Learning*, 2013, pp. 647–655.

[80] L. Bo, X. Ren, and D. Fox, "Hierarchical matching pursuit for image classification: Architecture and fast algorithms," in *Proc. Neural Information Processing Systems*, 2011, pp. 2115–2123.

[81] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar, "Attribute and simile classifiers for face verification," in *Proc. IEEE Int. Conf. Computer Vision*, Oct. 2009, pp. 365–372.

[82] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth, "Describing objects by their attributes," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2009, pp. 1778–1785.

[83] Y. Wang and G. Mori, "A discriminative latent model of object classes and attributes," in *Proc. European Conf. Computer Vision*, 2010, pp. 155–168.

[84] S. Branson, C. Wah, B. Babenko, F. Schroff, P. Welinder, P. Perona, and S. Belongie, "Visual recognition with humans in the loop," in *Proc. European Conf. Computer Vision*, pp. 438–451.

[85] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2009, pp. 951–958.

[86] A. Kovashka and K. Grauman, "Attribute adaptation for personalized image search," in *Proc. IEEE Int. Conf. Computer Vision*, 2013, pp. 3432–3439.

[87] T. Tommasi and B. Caputo, "Frustratingly easy NBNN domain adaptation," in *Proc. IEEE Int. Conf. Computer Vision*, 2013, pp. 897–904.

[88] S. S. F. V. Jain, "Adapting classification cascades to new domains," in *Proc. IEEE Int. Conf. Computer Vision*, 2013, pp. 105–112.

[89] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: A literature survey," in *ACM Computing Surveys*, Dec. 2003, pp. 399–458.

[90] H. Ho and R. Gopalan, "Model-driven domain adaptation on product manifolds for unconstrained face recognition," *Int. J. Comput. Vis.,* vol. 109, no. 1–2, pp. 110–125, Aug. 2014.

[91] R. Gross, I. Matthews, J. F. Cohn, T. Kanade, and S. Baker, "Multi-pie," *Image Vis. Comput.*, vol. 28, no. 5, pp. 807–813, 2010.

[92] M. Yang, L. Zhang, X. Feng, and D. Zhang, "Fisher discrimination dictionary learning for sparse representation," in *Proc. Int. Conf. Computer Vision*, 2011, pp. 543–550.

[93] G. Griffin, A. Holub, and P. Perona. (2007). Caltech-256 object category data set. California Institute of Technology, Tech. Rep. 7694. [Online]. Available: http://authors.library.caltech.edu/7694

[94] K. Gallivan, A. Srivastava, X. Liu, and P. Van Dooren, "Efficient algorithms for inferences on Grassmann manifolds," in *Proc. IEEE Workshop Statistical Signal Processing,* Sept. 2003, pp. 315–318.

[95] R. Rubinstein, M. Zibulevsky, and M. Elad, "Efficient implementation of the k-svd algorithm using batch orthogonal matching pursuit," Technion–Computer Science Department, Tech. Rep. CS-2008-08, Apr. 2008.

[96] M. Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. New York: Springer, 2010.

[97] A. Yang, Z. Zhou, A. Balasubramanian, S. Sastry, and Y. Ma, "Fast l1-minimization algorithms for robust face recognition," *IEEE Trans. Image Processing*, vol. 22, no. 8, pp. 3234–3246, Aug. 2013.

[98] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 171–184, Jan. 2013.

[99] Z. Lin, A. Ganesh, J. Wright, L. Wu, M. Chen, and Y. Ma, "Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix," UIUC Technical Report, Tech. Rep. UILU-ENG-09-2214, July 2009.

[100] Z. Lin, M. Chen, L. Wu, and Y. Ma, "The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices," UIUC Tech. Rep. UILU-ENG-09-2215, Oct 2009.

[101] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York: Cambridge Univ. Press, 2004.

[102] X. Wang and X. Tang, "Face photo-sketch synthesis and recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 11, pp. 1955–1967, Nov. 2009.

[103] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: a database for studying face recognition in unconstrained environments," Univ. Massachusetts, Amherst, LFWTech. Rep. 07-49, Oct. 2007.

[SP]