Adaptive RGB Image Recognition by Visual-Depth Embedding

Ziyun Cai, Yang Long, and Ling Shao

***Abstract*— Recognizing RGB images from RGB-D data is a promising application, which significantly reduces the cost while can still retain high recognition rates. However, existing methods still suffer from the domain shifting problem due to conventional surveillance cameras and depth sensors are using different mechanisms. In this paper, we aim to simultaneously solve the above two challenges: 1) how to take advantage of the additional depth information in the source domain? 2) how to reduce the data distribution mismatch between the source and target domains? We propose a novel method called adaptive visual-depth embedding (aVDE), which learns the compact shared latent space between two representations of labeled RGB and depth modalities in the source domain first. Then the shared latent space can help the transfer of the depth information to the unlabeled target dataset. At last, aVDE models two separate learning strategies for domain adaptation (feature matching and instance reweighting) in a unified optimization problem, which matches features and reweights instances jointly across the shared latent space and the projected target domain for an adaptive classifier. We test our method on five pairs of data sets for object recognition and scene classification, the results of which demonstrates the effectiveness of our proposed method.**

***Index Terms*— RGB-D data, domain adaptation, visual categorization.**

I. Introduction

DUE to the recent developments in low-cost RGB-D sensors, *e.g.*, the Microsoft Kinect, using additional depth information to boost the performance of recognition and classification tasks has received an increasing interest through out the computer vision community [1]–[3]. Particularly, the problem of recognizing RGB images captured by conventional surveillance cameras through leveraging a set of labeled RGB-D data has been presented in [4]–[6]. This new task is considered as an unsupervised domain adaptation (UDA) problem, which aims to take advantage of the additional depth information in the source domain and reduce the data distribution mismatch between the source and target domains simultaneously.

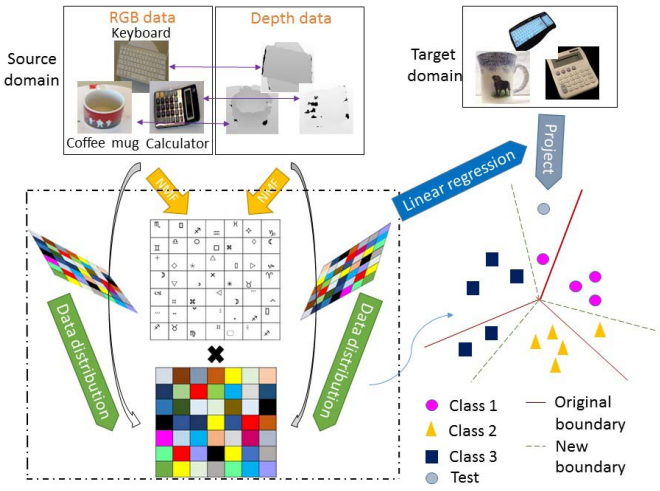

Fig. 1. The outline of the proposed method. We have RGB and depth features in the source domain, and RGB features in the target domain. Our main idea aims to find a shared latent space so that the shared parts between RGB and depth images can be preserved. Our aVDE can automatically adapt to the target latent space so as to further correct the classification errors. Examples from three classes are used to show the difference between the original decision boundaries and the new decision boundaries which are obtained by matching and reweighting.

The training data in UDA consists of labeled RGB-D source data and unlabeled RGB target examples [7]. It is different from traditional classification problems which often assume that the labeled training data comes from the same distribution as that of the test data. In realistic scenarios, the source and target domains follow different distributions, especially when images are acquired from different cameras, or in various conditions. The classifier which is trained on the previous dataset would fail to classify the following dataset correctly without adaptation.

To this end, there are two challenges in our task: 1) How to address the domain shifting problem between the source and target domains? 2) How to effectively explore the additional depth information to boost the performance further? A very fruitful line of work has been focusing on solving domain adaptation problem, where labeled target data is not needed, yielding excellent results [8]–[10]. However, none of them can incorporate depth information. On the other hand, many methods using the additional depth information have been proposed for classification tasks as well [11], [12]. However, these methods take the unrealistic assumption that the training and testing data are from the same domain.

Z. Cai is with the College of Automation, Nanjing University of Posts and Telecommunications, China (e-mail: caiziyun@163.com).

Y. Long is with the Open Lab, School of Computing, University of Newcastle, Newcastle upon Tyne NE4 5TG, U.K. (e-mail: yang.long@ieee.org).

L. Shao is with the Inception Institute of Artificial Intelligence, Abu Dhabi, United Arab Emirates, and also with the School of Computing Sciences, University of East Anglia, Norwich NR4 7TJ, U.K. (e-mail: ling.shao@ieee.org).

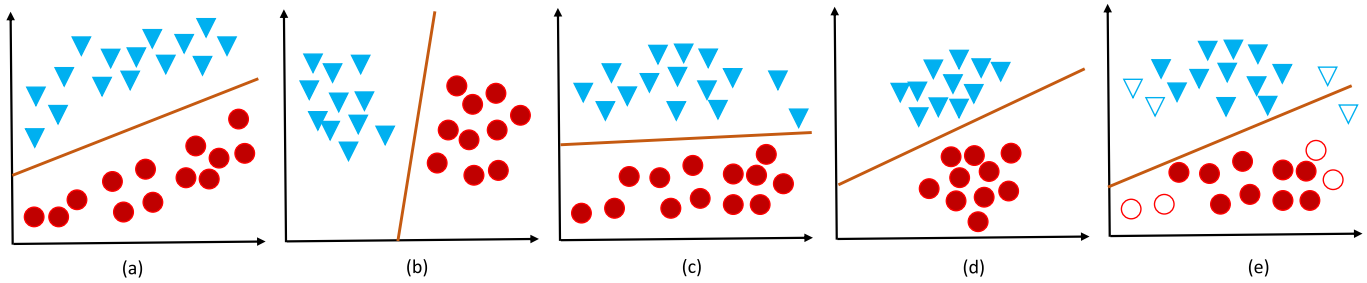

Fig. 2. Samples from the shared latent space and the projected target domain. (a) Shared latent space; (b) Projected target domain; (c) Shared latent space after feature matching; (d) Projected target domain after feature matching. The domain distance is still large after feature matching. (e) Further instance reweighting on shared latent space. The irrelevant shared latent space instances (shown as unfilled markers) are now down-weighted to further reduce the domain difference.

In this paper, we aim to solve above two challenges simultaneously by a novel RGB-D UDA method, referred to as **a**daptive **V**isual-**D**epth **E**mbedding (aVDE). The motivation behind our aVDE is as follows: depth images contain useful discriminative information, which shows a different feature distribution compared to the corresponding RGB image domain. To enhance the discriminative capability of the original learning system, joint learning is considered through combining depth information and RGB data into one model. The pipeline of our idea is described in Fig. 1. In the visual-depth embedding step, we capture the shared latent bases and individual subspaces between two representations of labeled RGB and depth modalities in the source domain first. We utilize the advantages of Nonnegative Matrix Factorization (NMF) [13]–[15] for the discovery of the shared components between RGB and depth images. In addition, since NMF cannot discover the intrinsic geometrical and discriminating structure of the data space, to preserve as much of the significant structure of the original RGB-D data as possible, we solve this problem from the probability distribution perspective, *i.e.* to minimize the Jensen-Shannon divergence (JSD) between the probability distributions in RGB and depth spaces. Then we transfer the knowledge of depth information to the target dataset through an orthogonal projection to align the data in the shared latent feature space with the target domain. In the adaptive embedding step, we minimize the nonparametric Maximum Mean Discrepancy (MMD) in an infinite dimensional reproducing kernel Hilbert space (RKHS) [16] for feature matching, and minimize the $\ell_{2,1}$-norm structured sparsity penalty [17] on the shared latent space instances for instance reweighting. We match features and reweight instances jointly across the shared latent space and the projected target domain in a principled dimensionality reduction procedure for an adaptive classifier. Feature matching can discover a shared feature representation through the combination of the distribution difference reduction and the important properties of input data preservation (see Fig. 2 (c) (d)). However, when the domain difference is substantially large, some shared latent space instances are still not relevant to the projected target instances even in the feature-matching subspace. Therefore, we introduce instance reweighting which can minimize the distribution difference through reweighting the shared latent data (see Fig. 2 (e)). Comprehensive experiments for object recognition and scene classification on five pairs of real-world datasets show that our aVDE can significantly outperform state-of-the-art methods.

To summarize, our main contributions are:

i) We propose a novel UDA method which can effectively leverage depth information to recognize RGB images. The target domain does not contain the additional depth information.

ii) aVDE can learn compact shared space uncovering the latent semantics and simultaneously preserve the joint probability distribution of data in the source domain, then transfers the knowledge of depth information to the target dataset.

iii) Through matching features and reweighting instances jointly across domains, a bridge between the shared latent space and the projected target domain can be built.

The rest of this paper is organized in the following way. Section 2 reviews related work on domain adaptation. In Section 3, The proposed adaptive Visual-Depth Embedding method is illustrated. The experimental setup, results and analysis on aVDE for several domain adaptation based vision tasks are shown in Section 4. Finally, the conclusion is given in Section 5.

II. Related Work

In the literature, only a few work focus on recognizing RGB images from RGB-D data. Our algorithm is mostly related to the methods in [5] and [6]. Reference [6] which uses cross-domain dictionary learning over both RGB and depth images in the training step and then spanned the intra-class diversities to maximize the inter-class distances while minimizing the intra-class distances. Since some labels in the test domain are used, it is a semi-supervised domain adaptation problem. In contrast, our domain adaption is completely unsupervised, just like Multi-view to single-view (DA_M2S) adaptation [5] attempts to seek an optimal projection matrix to map samples from two different domains into a common feature space, in which no label of the test domain is used.

Our work is also related to unsupervised domain adaptation methods. Transfer Component Analysis (TCA) [18] tries to learn some transfer components across domains in a Reproducing Kernel Hilbert Space (RKHS) through Maximum Mean Discrepancy (MMD), which only takes advantage of feature matching but ignores the advantages of instance reweighting. Sampling Geodesic Flow (SGF) [19] creates intermediate representations of data between two domains through viewing the generative subspaces created from these domains as

TABLE I
NOTATIONS AND DESCRIPTIONS

Notation	Description	Notation	Description
$\mathcal{D}_s$, $\mathcal{D}_t$	Source/target domain	$\mathcal{P}_{\mathcal{D}_t}$	Projected target domain
A, B	RGB/depth modality	X	Input data matrix
V	Shared data space	K	Kernel matrix
P_A, P_B	Probability distributions	M	Adaptation matrix
$\mathcal{P}$	Orthogonal projection	Δ	MMD matrix
Λ	Connection matrix	$\widehat{\mathcal{G}}$	Diagonal sub-gradient matrix
D	Number of bases	k	Subspace bases
η, μ	Regularization parameter	Z	Subspace embedding

points on the Grassmann manifold, and then obtains subspaces which can provide a description of the underlying domain shift. Landmark (LMK) [20] exploits a subset of source domain that is most similar to the target domain. Besides, there still exist some approaches that use NMF to achieve domain adaptation. Transfer Nonnegative Matrix Factorization (TNMF) in [21] minimizes the distribution divergence between labeled and unlabeled images, and incorporates this criterion into the objective function of NMF to construct new robust representations. TNMF is a semi-supervised transfer learning approach. Unsupervised Nonnegative Embedding (UNE) [22] generates a non-negative embedding for the source and target tasks as a shared feature space of two aligned sets of their corresponding non-negative basis vectors for a prototype matrix. However, although NMF on DA has been proposed, our aVDE is significantly different from these methods. i) The methods using NMF to achieve domain adaptation are not applicable to our problem: recognizing RGB images from RGB-D data. Both of them focus on RGB source domain and RGB target domain, which is a completely different task. ii) Our NMF-related equations are designed for visual-depth embedding, not for domain adaptation. iii) Conventional NMF is widely known to be not robust to data distribution discrepancy. To preserve as much of the significant structure of the original RGB-D data as possible and balance the difference of data distributions between the RGB and depth modalities, we consider Jensen-Shanon divergence in addition. These methods perform poorly on RGB-D scenarios. We provide the extensive comparison to these methods in our experiments, from which we demonstrate the advantages of our method.

III. ADAPTIVE VISUAL-DEPTH EMBEDDING

A. Notations

In this paper, we denote a vector by a lowercase letter in bold. The transpose of a vector or a matrix is denoted by the superscript T. We define I as an identity matrix. Besides, Table I shows the list of frequently used notations.

Problem (Adaptive Visual-Depth Embedding): Given two labeled modalities A and B in the source domain $\mathcal{D}_s$ with label set $Y = [y_1, \cdots, y_{N_s}]$ and an unlabeled target domain $\mathcal{D}_t$. To find the shared component space V and the projected target domain $\mathcal{P}_{\mathcal{D}_t}$ under the different marginal probability distribution and conditional probability distribution, then learn a new feature space to reduce the domain distance by feature matching and instance reweighting across V and $\mathcal{P}_{\mathcal{D}_t}$.

B. Shared Component Problem Formulation

We use A and B to define the two modalities in the source domain $\mathcal{D}_s$ with dimensions and sample sizes $M_1 \times N_s$ and $M_2 \times N_s$ respectively: $A = [\mathbf{a}_1, \cdots, \mathbf{a}_{N_s}] \in \mathbb{R}_{\geq 0}^{M_1 \times N_s}$ and $B = [\mathbf{b}_1, \cdots, \mathbf{b}_{N_s}] \in \mathbb{R}_{\geq 0}^{M_2 \times N_s}$. NMF is used to find two nonnegative matrices from A: $U \in \mathbb{R}_{\geq 0}^{M_1 \times D_1}$ and $V_1 \in \mathbb{R}_{\geq 0}^{D_1 \times N_s}$ and two nonnegative matrices from B: $W \in \mathbb{R}_{\geq 0}^{M_2 \times D_2}$ and $V_2 \in \mathbb{R}_{\geq 0}^{D_2 \times N_s}$ with full rank whose product can approximately represent the original matrix A and B, *i.e.*, $A \approx UV_1$ and $B \approx WV_2$. In practice, we set $D_1 < \min(M_1, N_s)$ and $D_2 < \min(M_2, N_s)$. NMF aims to achieve the minimization of the following objective functions

$$\mathcal{L}_{NMF}^{A} = \|A - UV_1\|^2, \quad s.t.\ U, V_1 \geq 0, \tag{1}$$

$$\mathcal{L}_{NMF}^{B} = \|B - WV_2\|^2, \quad s.t.\ W, V_2 \geq 0, \tag{2}$$

where $\|\cdot\|$ is the Frobenius norm. The matrix V_1 and V_2 obtained in NMF are regarded as the low-dimensional representations while the matrix U and W denote the basis matrixes.

To learn fully shared spaces between RGB and depth modalities, the basic idea is to find suitable M_1 basis vectors for U and M_2 basis vectors for W via a shared coefficient matrix V. To learn the required shared space, we jointly optimize a convex combination of two constrained least squares problems: $V_1 = V_2 = V \in \mathbb{R}_{\geq 0}^{D \times N_s}$. The resulted objective function is:

$$\min_{U,W,V} \|A - UV\|^2 + \lambda \|B - WV\|^2, \quad s.t.\ U, W, V \geq 0, \tag{3}$$

where parameter λ is given to balance the importance of the two terms. In our paper, since RGB information and depth data are assumed equally important, for simplicity, we set $\lambda = 1$. The training model is used to identify the latent shared bases determined via both RGB and depth data. Such jointed NMF can preserve shared components that make the model leads to a high-level representation V of the training RGB-D images in the bases space.

C. Data Distribution Divergency Reduction

NMF can learn a parts-based representation. Theoretically, it is expected that the shared data space V given by our NMF-based shared structure learning algorithm can obtain locality structure from the original data spaces A and B. However, NMF cannot discover the intrinsic geometrical and discriminating structure of the data space, which is important for our recognition task. Therefore, to preserve as much of the significant structure of the original RGB-D data as possible, we hope the latent space can also balance the difference of data distribution between the RGB and depth modalities. We consider this problem from probability distribution aspect. Let P_A and P_B be the probability distributions in space A and B. We aim to find the joint probability distribution in the shared space Q that can be shared by P_A and P_B as much as possible. In this paper, we simply assume RGB and depth are equally important, *i.e.*, we hope the probability distribution Q in the latent space V can be $Q = \frac{1}{2}(P_A + P_B)$. We can then minimize the Jensen-Shannon divergence (JSD) between P_A and P_B so

that their structural difference can be mutually mitigated:

$$JSD(P_A||P_B) = \frac{1}{2}KL(P_A||Q) + \frac{1}{2}KL(P_B||Q), \quad (4)$$

where $KL(.||.)$ estimates the Kullback-Leibler divergence between the joint probability distributions.

P_A and P_B can be denoted as point-wise from p_A^{ij} and p_B^{ij}. Q can be represented as q_{ij}. The pairwise similarities in the original data space p_A^{ij} and p_B^{ij} are defined as:

$$p_A^{ij} = \frac{\exp\left(-\|\mathbf{a}^i - \mathbf{a}^j\|^2/2(\sigma_A^i)^2\right)}{\sum_{k\neq l}\exp\left(-\|\mathbf{a}_k - \mathbf{a}_l\|^2/2(\sigma_A^k)^2\right)}, \quad (5)$$

$$p_B^{ij} = \frac{\exp\left(-\|\mathbf{b}^i - \mathbf{b}^j\|^2/2(\sigma_B^i)^2\right)}{\sum_{k\neq l}\exp\left(-\|\mathbf{b}_k - \mathbf{b}_l\|^2/2(\sigma_B^k)^2\right)}, \quad (6)$$

where the conditional probability p_A^{ij} means the similarity between data points $\mathbf{a}^i$ and $\mathbf{a}^j$, and p_B^{ij} means the similarity between data points $\mathbf{b}^i$ and $\mathbf{b}^j$, where $\mathbf{a}^j$ and $\mathbf{b}^j$ are picked in proportion to their probability density under a Gaussian centered at $\mathbf{a}^i$ and $\mathbf{b}^i$ respectively. σ_A^k and σ_B^k are the variances of the Gaussian distribution which is centered on data point a^i and b^i respectively. Each data point a^i or b^i makes a significant contribution to the cost function. In the shared space, using the probability distribution that is heavy tailed, the joint probabilities q_{ij} can be defined as:

$$q_{ij} = \frac{(1+\|\mathbf{v}_i - \mathbf{v}_j\|^2)^{-1}}{\sum_{k\neq l}(1+\|\mathbf{v}_k - \mathbf{v}_l\|^2)^{-1}}. \quad (7)$$

We set p_A^{ii}, p_B^{ii} and q_{ii} to zero for only significant points needed to model pairwise similarities. Meanwhile, it has the characteristics that $p_A^{ij} = p_A^{ji}$, $p_B^{ij} = p_B^{ji}$ and $q_{ij} = q_{ji}$ for $\forall i, j$. Since the definition in Eq. (7) is an infinite mixture of Gaussians which does not have an exponential, it is much faster to evaluate the density of a point than a single Gaussian. This representation also makes the mapped points invariant to the changes in the scale for the embedded points that are far apart. Thus, the cost function based on JSD can effectively measure the significance of the data distribution.

We use q_{ij} to jointly model p_A^{ij} and p_B^{ij}:

$$JSD = \frac{1}{2}\sum_i\sum_j p_A^{ij}\log p_A^{ij} - p_A^{ij}\log q_{ij} + \frac{1}{2}\sum_i\sum_j p_B^{ij}\log p_B^{ij} - p_B^{ij}\log q_{ij}. \quad (8)$$

Therefore, with this regularization, through combining the data structure preserving part in Eq. (8) and the shared structure technique in Eq. (3), we minimize the following objective function:

$$\min_{U,W,V} \|A - UV\|^2 + \|B - WV\|^2 + \eta JSD, \quad s.t.\ U, W, V \geq 0, \quad (9)$$

where $A \in \mathbb{R}^{M_1\times N_s}$, $B \in \mathbb{R}^{M_2\times N_s}$, $V \in \mathbb{R}^{D\times N_s}$, $A, B, U, W, V \geqslant 0$, $U \in \mathbb{R}^{M_1\times D}$, $W \in \mathbb{R}^{M_2\times D}$, and η controls the smoothness of the new representation.

The shared space data only from NMF-based shared structure algorithm is not effective and meaningful for real world applications. Therefore, we introduce JSD to preserve the structure of the original RGB-D data which can obtain better results.

D. Optimization

Let the Lagrangian of our problem be:

$$\mathcal{L} = \|A - UV\|^2 + \|B - WV\|^2 + \eta JSD + tr(\Phi U^T) + tr(\Theta W^T) + tr(\Psi V^T), \quad (10)$$

where matrices Φ, Θ and Ψ are three Lagrangian multiplier matrices. In order to make the derivation clearer, ηJSD is simply denoted as G. We define two auxiliary variables d_{ij} and Z as follows:

$$d_{ij} = \|\mathbf{v}_i - \mathbf{v}_j\| \text{ and } Z = \sum_{k\neq l}(1+d_{kl}^2)^{-1}. \quad (11)$$

There is a need to note that if $\mathbf{v}_i$ changes, the only pairwise distances that change are d_{ij} and d_{ji}. Therefore, the gradient of function G with respect to $\mathbf{v}_i$ can be given by

$$\frac{\partial G}{\partial \mathbf{v}_i} = 2\sum_{j=1}^{N}\frac{\partial G}{\partial d_{ij}}(\mathbf{v}_i - \mathbf{v}_j). \quad (12)$$

Then $\frac{\partial G}{\partial d_{ij}}$ can be calculated by Kullback-Leibler divergence in Eq. (8):

$$\frac{\partial G}{\partial d_{ij}} = -\frac{\eta}{2}\sum_{k\neq l}(p_A^{kl} + p_B^{kl})\left(\frac{1}{q_{kl}Z}\frac{\partial((1+d_{kl}^2)^{-1})}{\partial d_{ij}} - \frac{1}{Z}\frac{\partial Z}{\partial d_{ij}}\right). \quad (13)$$

Since $\frac{\partial((1+d_{kl}^2)^{-1})}{\partial d_{ij}}$ is nonzero if and only if $k = i$ and $l = j$, and $\sum_{k\neq l} p_{kl} = 1$, the gradient function can be simplified as

$$\frac{\partial G}{\partial d_{ij}} = \eta(p_A^{ij} + p_B^{ij} - 2q_{ij})(1+d_{ij}^2)^{-1}. \quad (14)$$

Eq. (14) can be substituted into Eq. (12). Therefore, the gradient of the Kullback-Leibler divergence between P and Q is

$$\frac{\partial G}{\partial \mathbf{v}_i} = 2\eta\sum_{j=1}^{N}(p_A^{ij} + p_B^{ij} - 2q_{ij})(\mathbf{v}_i - \mathbf{v}_j)(1+\|\mathbf{v}_i - \mathbf{v}_j\|^2)^{-1}. \quad (15)$$

Since we have the gradient of G in Eq. (15), we make the gradients of $\mathcal{L}$ be zeros to minimize O_f:

$$\frac{\partial \mathcal{L}}{\partial V} = 2(-U^TA + U^TUV - W^TB + W^TWV) + \frac{\partial G}{\partial V} + \Psi = \mathbf{0}, \quad (16)$$

$$\frac{\partial \mathcal{L}}{\partial U} = 2(-AV^T + UVV^T) + \Phi = \mathbf{0}, \quad (17)$$

$$\frac{\partial \mathcal{L}}{\partial W} = 2(-BW^T + WVV^T) + \Theta = \mathbf{0}. \quad (18)$$

In addition, we also have KKT conditions: $\Phi_{ij}U_{ij} = 0$, $\Theta_{ij}W_{ij} = 0$ and $\Psi_{ij}V_{ij} = 0$, $\forall i, j$. Then multiplying V_{ij},

U_{ij} and W_{ij} in the corresponding positions on both sides of Eqs. (16), (17) and (18) respectively, we obtain

$$\left(2(-U^TA+U^TUV-W^TB+W^TWV)+\frac{\partial G}{\partial \mathbf{v}_i}\right)_{ij} V_{ij}=0, \tag{19}$$

$$2(-AV^T+UVV^T)_{ij}U_{ij}=0, \tag{20}$$

$$2(-BV^T+WVV^T)_{ij}W_{ij}=0. \tag{21}$$

Note that

$$\left(\frac{\partial G}{\partial \mathbf{v}_j}\right)_i = \left(2\eta\sum_{k=1}^{N}\frac{(p_A^{jk}+p_B^{jk}-2q_{jk})(\mathbf{v}_j-\mathbf{v}_k)}{1+\|\mathbf{v}_j-\mathbf{v}_k\|^2}\right)_i$$
$$= 2\eta\sum_{k=1}^{N}\frac{(p_A^{jk}+p_B^{jk}-2q_{jk})(V_{ij}-V_{ik})}{1+\|\mathbf{v}_j-\mathbf{v}_k\|^2}.$$

The multiplicative update rules of bases of both W and U for any i and j are obtained:

$$U_{ij} \leftarrow \frac{(AV^T)_{ij}}{(UVV^T)_{ij}}U_{ij}, \tag{22}$$

$$W_{ij} \leftarrow \frac{(BV^T)_{ij}}{(WVV^T)_{ij}}W_{ij}. \tag{23}$$

The update rule of the shared space preserving coefficient matrix V between RGB and depth data spaces is:

$$V_{ij} \leftarrow \frac{(U^TA)_{ij}+(W^TB)_{ij}+\Upsilon}{(U^TUV)_{ij}+(W^TWV)_{ij}+\Gamma}V_{ij}, \tag{24}$$

where for simplicity, we let $\Upsilon = \eta\sum_{k=1}^{N}\frac{(p_A^{jk}+p_B^{jk})V_{ik}+2q_{jk}V_{ij}}{1+\|\mathbf{v}_j-\mathbf{v}_k\|^2}$, $\Gamma = \eta\sum_{k=1}^{N}\frac{(p_A^{jk}+p_B^{jk})V_{ij}+2q_{jk}V_{ik}}{1+\|\mathbf{v}_j-\mathbf{v}_k\|^2}$.

All the elements in U, W and V can be guaranteed that they are nonnegative from the allocation. It proves that the objective function is monotonically non-increasing after each update of U, W or V. The proof of convergence about U, W and V follows similar lines in [23]–[25].

After U, W and V are converged, we can obtain the shared structure representation by a linear projection matrix. Since our algorithm is NMF-based, a direct projection from the target domain to the shared space does not exist for data embedding. Therefore, inspired by [26], linear regression is used to compute our projection matrix. It is equivalent to find a rotation to align the data in the current feature space with another, which is a classic Orthogonal Procrustes problem [27]. Through solving this problem, we can make the projection orthogonal:

$$\min_{\mathcal{P}}\|\mathcal{P}A-V\|, \quad s.t.\ \mathcal{P}^T\mathcal{P}=I, \tag{25}$$

where $\mathcal{P}$ is the orthogonal projection for target domain. According to [28], the advantages on using orthogonal projection can be summarized as: 1) The orthogonal projection can preserve the Euclidean distance between points; 2) The orthogonal projection can distribute the variance more evenly across the dimensions; 3) The orthogonal projection can learn maximally uncorrelated dimensions, which leads more compact representations. For the optimal solution, we firstly use the singular value decomposition algorithm to decompose the matrix: $A^TV = Q\Sigma S^T$. Then we calculate $\mathcal{P} = S\Lambda Q^T$, where Λ is a connection matrix as $\Lambda = [I, \mathbf{0}] \in \mathbb{R}^{D\times M}$ and $\mathbf{0}$ indicates all zeros matrix. Once we obtain the orthogonal projection $\mathcal{P}$, RGB data in the target domain $\hat{\mathbf{a}} \in \mathbb{R}^{M_1\times 1}$ can be projected into the latent space:

$$\mathbf{v}_{\hat{\mathbf{a}}} = \mathcal{P}\hat{\mathbf{a}}. \tag{26}$$

E. Adaptive Embedding

Although our above Visual-Depth Embedding (VDE) can correct the noise by projecting RGB into the shared space, the domain shifting problem remains unsolved. In the following, we propose an adaptive strategy to make VDE adaptive to target domain RGB data. In aVDE, we define the target domain as $\mathcal{D}_t = [\hat{\mathbf{a}}_1, \cdots, \hat{\mathbf{a}}_{N_t}]$. The projected target domain is defined as $\mathcal{P}_{\mathcal{D}_t} = [\mathbf{v}_{\hat{\mathbf{a}}_1}, \cdots, \mathbf{v}_{\hat{\mathbf{a}}_{N_t}}] \in \mathbb{R}^{D\times N_t}$. The shared component space is $V = [\mathbf{v}_1, \cdots, \mathbf{v}_{N_s}] \in \mathbb{R}^{D\times N_s}$.

The model proposed above learns the relationship between RGB data space and shared bases, and the shared bases are determined via both RGB data space and depth data space in the source domain. Since exploring feature matching and instance reweighting independently may not be effective enough when the domain difference is substantially large, we match features and reweight instances jointly across the latent shared space V and the new space projected from the target domain to the shared space in a principled dimensionality reduction procedure for an accurate classifier. If we only consider matching the feature distributions based on MMD minimization, it is not good enough for domain adaptation. This strategy only matches the first- and high-order statistics, and the distribution matching is far from perfect. When the domain difference is large, there will still exist some shared latent space instances that are not relevant to the projected target instances even in the feature matching subspace. Therefore, combining feature matching and instance reweighting procedures should be considered to handle this difficult setting. But it is difficult to reweight source instances when we match the feature distributions in the infinite dimensional RKHS simultaneously. In this step, we impose the $\ell_{2,1}$-norm structured sparsity regularizer on the transformation matrix for Kernel PCA M, which can introduce row-sparsity to the transformation matrix.

We first mix projected target data with the source data, on which we perform Principal Component Analysis (PCA) for data reconstruction. Let $X = [\mathbf{x}_1, \cdots, \mathbf{x}_n] = [\mathbf{v}_1, \cdots, \mathbf{v}_{N_s}, \mathbf{v}_{\hat{\mathbf{a}}_1}, \cdots, \mathbf{v}_{\hat{\mathbf{a}}_{N_t}}] \in \mathbb{R}^{D\times n}$ as the input data matrix, and $H = I - \frac{1}{n}\mathbf{1}$ as the centering matrix, where $n = N_s + N_t$ and $\mathbf{1}$ indicates all ones matrix, then the covariance matrix can be computed as XHX^T. PCA can find an orthogonal transformation matrix $T \in \mathbb{R}^{D\times k}$, where k is the subspace bases such that embedded data variance is maximized

$$\max_{T^TT=I} tr(T^TXHX^TT). \tag{27}$$

Above optimization problem can be efficiently solved by eigen-decomposition $XHX^TT = T\Omega$, where

$\Omega = \text{diag}(\omega_1, \cdots, \omega_k) \in \mathbb{R}^{k\times k}$ are the k largest eigenvalues. Then we find the optimal k-dimensional representation by $Z = [\mathbf{z}_1, \cdots, \mathbf{z}_n] = T^T X$.

To work in the RKHS $\mathcal{H}$, consider kernel mapping φ: $x \rightarrow \varphi(\mathbf{x})$, or $\varphi(X) = [\varphi(\mathbf{x}_1), \cdots, \varphi(\mathbf{x}_n)]$, and kernel matrix $K = \varphi(X)^T\varphi(X) \in \mathbb{R}^{n\times n}$. We utilize the Representer theorem $T = \omega(X)M$ to kernelize PCA as

$$\max_{M^T M=I} \; tr(M^T K H K^T M), \tag{28}$$

where $M \in \mathbb{R}^{n\times k}$ is the transformation matrix for Kernel PCA. We also call M an adaptation matrix. The subspace embedding becomes $Z = M^T K$.

Then we adopt the empirical MMD [18] as the nonparametric distance measure for comparing distributions based on the RKHS. Through k-dimensional embeddings extracted by Kernel-PCA, MMD computes the distance between the empirical expectations of shared component source and target data:

$$\left\| \frac{1}{N_s}\sum_{i=1}^{N_s} M^T k_i - \frac{1}{N_t}\sum_{j=N_s+1}^{N_s+N_t} M^T k_j \right\|_{\mathcal{H}}^2 = tr(M^T K \Delta K^T M), \tag{29}$$

where Δ is the MMD matrix and can be computed as

$$\Delta_{ij} = \begin{cases} \dfrac{1}{N_s N_s}, & x_i, x_j \in V \\ \dfrac{1}{N_t N_t}, & x_i, x_j \in \mathcal{P}_{\mathcal{D}_t} \\ -\dfrac{1}{N_s N_t}, & otherwise \end{cases} \tag{30}$$

Through minimizing Eq. (29) such that Eq. (28) is maximized, the first-order and high-order statistics of feature distributions are matched in the new representation $Z = M^T K$.

To impose the $\ell_{2,1}$-norm structured sparsity regularizer on the transformation matrix M, we introduce row-sparsity to the transformation matrix, which is able to facilitate adaptive instance reweighting essentially. Our target is to reweight source instances by their relevance to the target instances. The instance reweighting regularizer can be defined as

$$\|M_s\|_{2,1} + \|M_t\|_F^2, \tag{31}$$

where $M_s := M_{1:N_s,:}$ is the transformation matrix corresponding to the source instances, and $M_t := M_{N_s+1:n,:}$ is the transformation matrix corresponding to the target instances. Through minimizing Eq. (31), Eq. (27) is maximized, and the source instances which are relevant to the target instances are reweighted adaptively with greater importance in the representation $Z = M^T K$. On the contrary, the source instances which are irrelevant to the target instances are adaptively reweighted with less importance in $Z = M^T K$. Therefore, aVDE can be robust to the domain difference which is caused by the irrelevant instances. Note that the Frobenius norm: $\|M\|_F = \sqrt{\sum_{i=1}^n \|m^i\|_2^2}$. The $\ell_{2,1}$-norm: $\|M\|_{2,1} = \sum_{i=1}^n \|m^i\|_2$. Therefore, by combining Eq. (29) and Eq. (31) into Eq. (28), the optimization problem is defined as

$$\min_{M^T X H X^T M=I} tr(M^T K \Delta K^T M) + \mu(\|M_s\|_{2,1} + \|M_t\|_F^2), \tag{32}$$

where μ is the regularization parameter to trade off feature matching and instance reweighting, $M_s := M_{1:N_s,:}$ is the transformation matrix corresponding to the source instances, and $M_t := M_{N_s+1:n,:}$ is the transformation matrix corresponding to the target instances, $n = N_s + N_t$. In addition, $M \in \mathbb{R}^{n\times k}$, $M_s \in \mathbb{R}^{N_s\times k}$ and $M_t \in \mathbb{R}^{N_t\times k}$. In aVDE, when $\mu \rightarrow 0$, aVDE optimization problem degenerates. When $\mu \rightarrow \infty$, the joint feature matching and instance reweighting is not performed. Therefore, we set $\mu = 1$.

To impose the $\ell_{2,1}$-norm structured sparsity regularizer on the transformation matrix M, we introduce row-sparsity to the transformation matrix, which is able to facilitate adaptive instance reweighting essentially. Our target is to reweight source instances by their relevance to the target instances. Through minimizing Eq. (31), Eq. (28) is maximized, and the source instances which are relevant to the target instances are reweighted adaptively with greater importance in the new representation $Z = M^T K$. On the contrary, the source instances which are irrelevant to the target instances are adaptively reweighted with less importance in $Z = M^T K$. Therefore, aVDE can be robust to the domain difference which is caused by the irrelevant instances.

Since $\Omega = \text{diag}(\omega_1, \cdots, \omega_k) \in \mathbb{R}^{k\times k}$ is denoted as the Lagrange multiplier, through deriving the Lagrange function of problem Eq. (32) as

$$\begin{aligned}\mathscr{L} = {} & tr(M^T K \Delta K^T M) + \|M_s\|_{2,1} + \|M_t\|_F^2 \\ & + r((I - M^T K \Delta K^T M)\Omega).\end{aligned} \tag{33}$$

Let $\frac{\partial \mathscr{L}}{\partial M} = 0$, we obtain generalized eigendecomposition

$$(K \Delta K^T + \widehat{\mathcal{G}})M = K H K^T M \Omega. \tag{34}$$

$\widehat{\mathcal{G}}$ is a diagonal sub-gradient matrix with ith element equal to

$$\widehat{\mathcal{G}}_{ii} = \begin{cases} \dfrac{1}{2\|m^i\|}, & x_i \in V,\ m^i \neq 0 \\ 0, & x_i \in V,\ m^i = 0 \\ 1, & x_i \in \mathcal{P}_{\mathcal{D}_t} \end{cases} \tag{35}$$

The optimal adaptation matrix M is then reduced to solve Eq. (34) for the k smallest eigenvectors. An adaptive classifier f can be obtained by training on $\{M^T k_i, yi\}_{i=1}^{N_s}$. The convergence analysis of our adaptive embedding is similar to the methods in [29] and [30]. Finally, Algorithm 1 provides the details on aVDE. Since labeled and unlabeled data are sampled from different distributions that results in impossibly tuning the optimal parameters using cross validation, following [31], nearest neighbor classifier (NN) which does not require tuning cross-validation parameters is chosen as the base classifier.

F. Computational Complexity Analysis

The computational complexity of aVDE consists of three parts. We compare the cost of the basic NMF algorithm in [23] and our shared component part in Eq. (3). For an $M_1 \times N$

Algorithm 1 adaptive Visual-Depth Embedding (aVDE)

Input:
The source domain $\mathcal{D}_s$: $A \in \mathbb{R}^{M_1 \times N}$ and $B \in \mathbb{R}^{M_2 \times N}$; the target domain $\mathcal{D}_t$; number of bases D; the subspace bases k; the regularization parameter η; ground truth Y in source domain;.

Output: The basis matrix U, W, adaptation matrix M, embedding Z, adaptive classifier f.

1: Initialize U, W and V with uniformly distributed random values between 0 and 1.
2: **repeat**
3: Compute the basis matrixes U and W and the shared structure representation matrix V via Eqs. (22), (23) and (24), respectively;
4: **until** convergence
5: SVD decomposes the matrix $A^T V$ to obtain $Q\Sigma S^T$ and calculate $\mathcal{P} = S\Omega Q^T$
6: The shared component embedded representation of the coming target domain data $\mathbf{v}_{\hat{\mathbf{a}}} \in \mathbb{R}^{D\times 1}$ is defined in Eq. (26).
7: Compute MMD matrix Δ by Eq. (30), and kernel matrix K by $K_{ij} \leftarrow K(x_i, x_j)$ where $K(\cdot,\cdot)$ is a predefined kernel. Set $\Delta \leftarrow \Delta / \|\Delta\|_F$, $\widehat{\mathcal{G}} \leftarrow I$;
8: **repeat** Solve Eq. (34) and choose the k smallest eigenvectors to construct the adaptation matrix M, and $Z \leftarrow M^T K$. Update $\widehat{\mathcal{G}}$ by Eq. (35);
9: **until** convergence
10: Obtain an adaptive classifier f by training on $\{M^T k_i, yi\}_{i=1}^{N_s}$.

matrix A and an $M_2 \times N$ matrix B, assuming that the shared latent space dimensionality for decomposition of A and B is D, then computational complexity for the shared component part per iteration is $O(max\{M_1ND, M_2ND\})$. The basic NMF algorithm in [23] applied for A and B separately will have complexity of $O(M_1ND)$ and $O(M_2ND)$ respectively. This shows that the first part of aVDE has the same complexity as the basic NMF. The second part is the computation of matrices P_A, P_B and Q which has the complexity $O(2N^2D)$. The last part is adaptive embedding procedure whose complexity is $O(kn^2 + mn^2)$. Therefore, the total computational complexity of aVDE is: $O(max\{M_1ND, M_2ND\}t_1 + 2N^2D + t_2kn^2 + mn^2)$, where t_1 is the number of iterations when learning shared source space V, *i.e.*, from Line 1 to Line 4. t_2 is the number of iterations when learning adaptive classifier f, *i.e.*, from Line 5 to Line 9.

IV. Experiments and Results

In this section, we evaluate the effectiveness of our aVDE for object recognition and scene classification on five pairs of datasets (see Table II). Fig. 3 shows some example images of these five pairs of datasets. Since the images in source and target domains are from different kinds of cameras and various conditions, the domain difference between source and target is large. The details of the datasets, experimental settings, relevant experimental results, important parameter analysis and algorithm analysis are shown in the rest of this section. All experiments are performed using Matlab 2014a on a server configured with a 16-core processor and 500G of RAM running the Linux OS.

TABLE II
Statistics of the Benchmark Image Datasets

Dataset	Type	# Examples	# Features	# Classes
Object→ Caltech-256	Object	2059/1131	1000/4096	10
Object→ ImageNet	Object	1805/968	1000/4096	10
B3DO→ Caltech-256	Object	1129/776	1000/4096	8
B3DO→ ImageNet	Object	1135/789	1000/4096	8
NYU v1→ Scene-15	Scene	907/930	1000/4096	4

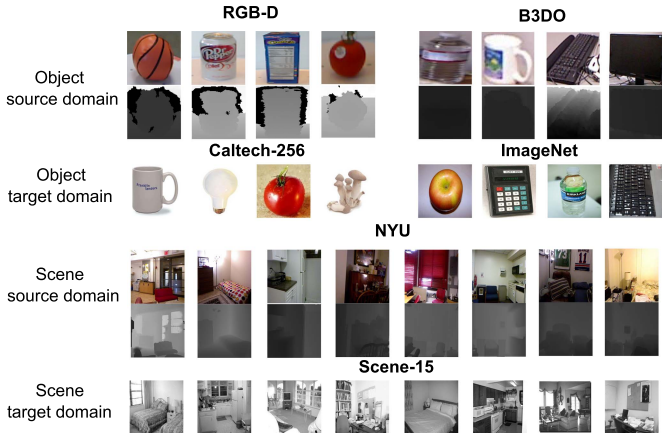

Fig. 3. Some example images from our selected datasets: RGB-D, B3DO, Caltech-256, ImageNet, NYU and Scene-15.

A. Datasets

1) Object Recognition:

1. Object→Caltech-256: We choose the RGB-D Object dataset [32] as the source domain and the Caltech-256 dataset [33] as the target domain for object recognition. RGB-D Object dataset contains 51 categories about 300 everyday objects. The Caltech-256 dataset only contains color images. They share ten common categories: "ball", "calculator", "cereal box", "coffee mug", "flashlight", "keyboard", "light bulb", "mushroom", "soda can" and "tomato". Since the RGB-D Object dataset is recorded as video sequences, we uniformly choose images with an interval of two seconds for each category resulting in 2059 training samples in the source domain. Note that each RGB image corresponds to a depth image. The 1131 RGB images from ten categories in the Caltech-256 dataset are used as the target domain to evaluate the performance of our aVDE.

2. Object→ImageNet: We choose the RGB-D Object dataset as the source domain and the ImageNet dataset [34] as the target domain. ImageNet contains more than 100, 000 categories, which is organized according to the WordNet hierarchy. The ImageNet dataset only contains color images. We select ten common categories of RGB-D and ImageNet datasets, "apple", "banana", "coffee mug", "keyboard", "soda can", "water bottle", "plate", "calculator", "cereal box" and "light bulb" to demonstrate our aVDE. Finally, we have 1805

RGB-D training image pairs in the source domain and 968 RGB images in the target domain.

3. B3DO→Caltech-256: We choose the B3DO dataset [35] as the source domain and the Caltech-256 dataset as the target domain. B3DO dataset contains 849 RGB images with its corresponding depth images. We apply the provided bounding boxes to crop the objects from these images. We randomly choose eight objects which are shared by B3DO and Caltech-256 dataset. The eight common categories are "bottle", "can", "cup", "keyboard", "monitor", "mouse", "phone" and "spoon". We have 1129 training image pairs and 776 RGB images in the target domain.

4. B3DO→ImageNet: The B3DO dataset is chosen as the source domain, while ImageNet dataset is the target domain. The common eight categories between these two datasets are used, "bottle", "cup", "keyboard", "monitor", "mouse", "phone", "plate", "spoon" - are used to evaluate our aVDE. We obtain 1135 RGB-D image pairs in the source domain and 789 RGB images in the target domain.

2) Scene Classification: For scene classification, we select the NYU Depth v1 dataset [36] as the source domain and the Scene-15 dataset [37] as the target domain. NYU Depth v1 dataset consists of video sequences from many indoor scenes. Scene-15 dataset contains only RGB images. We use the same four categories of NYU Depth v1 and Scene-15 datasets, "bedroom", "kitchen", "living room" and "office" to demonstrate our proposed algorithm. Finally, we have 907 RGB-D training image pairs in the source domain and 930 RGB images in the target domain to evaluate the performance of aVDE.

B. The Selected Methods and Settings

In our experiment, for a comprehensive and fair comparison, we select following five categories as the baselines including: 1) Naive Approach: SVM_A and 1-Nearest Neighbor Classifier which are trained by the RGB features in the source domain without considering the domain adaptation and the depth information compensation; 2) Multi-view Learning: Kernelisation of Canonical Correlation Analysis (KCCA) [38] and SVM2K [39] which use the two-view data in the source domain for training; 3) Learning Using Privileged Information: SVM+ [40] and Rank Transfer (RT) [41] which use the additional depth features in the source domain as privileged information; 4) Unsupervised Domain Adaptation: Kernel Mean Matching (KMM) [8], Domain Adaptation Machine (DAM) [42], Sampling Geodesic Flow (SGF) [19], TCA [18], Landmark (LMK) [20], Subspace Alignment (SA) [43], Geodesic Flow Kernel (GFK) [31], UNE [22] and Domain Invariant Projection (DIP) [44] which use the visual features from both domains for training the classifiers, and then predict target data based on the visual features. 5) Using Privileged Information and Unsupervised Domain Adaptation: Domain Adaptation from Multi-view to Single-view (DA-M2S) which uses the additional depth features in the source domain as privileged information and reduces the data distribution mismatch between the source and target domains.

We take the factor of feature performance into consideration, and then choose shallow features and deep features to evaluate aVDE respectively. For shallow features, we extract Gradient kernel descriptors (KDES) features and LBP KDES features [11] which are successful in RGB-D object dataset from each pair of RGB/depth images. The vocabulary size is set as 1000. Three level of pyramids (1×1, 2×2, 3×3) are used. For deep features, we choose ImageNet-CNN features [45] which are learned from the pre-trained Caffe model [46] on image classification dataset (*i.e.* ImageNet) for object classification, and the Places-CNN [47] scene features which are learned from the pre-trained Caffe model on scene classification dataset (*i.e.* Places dataset) for scene classification. Both of these two kinds of models obtain great success for object and scene classification respectively. In addition, according to Object→ImageNet and B3DO→ImageNet, since the features are obtained by fine-tuning on ImageNet and the study includes experiments on the imageNet dataset, the experimental results on these two pairs of datasets will perform a little higher. We add some experiments based on CNN features which are not fine-tuned. In this case, we extract features directly on the fully connected layer (fc7) in the ImageNet trained network, which follows the strategy in [48]. More specifically, the CNN model is considered as a feature extractor in the added experiments. The feature dimension after CNN is 4096. Note that the depth image is encoded as HHA image as in [49] before extracting the features.

From Eq. (9) and algorithm 1, the size of matrices $U \in \mathbb{R}^{M_1 \times D}$, $W \in \mathbb{R}^{M_2 \times D}$ and $V \in \mathbb{R}^{D \times N_s}$ should be predefined. M_1, M_2 and N_s are known when the data is given. However, the value of number of latent bases D is difficult to be pre-determined. In aVDE, an improper D will result in the limitation of identification of latent topics or the increase of possibility of overfitting. In order to investigate the effects of D, we choose different number of bases, *e.g.*, 40, 60, 80, 100, 120 and 140. We also explore the sensitivity of the parameter η in Eq. (9) on the performance of aVDE. We set the parameter η by searching $\eta \in \{0, 1/8, 1/4, 1/2, 1, 2, 4, 8\}$. Besides, the subspace bases k is also related. We analyze the behavior of aVDE by searching $k \in \{10, 20, \cdots, 100\}$. We limit the maximum number of t_1 with 1000, and let $t_2 = 10$ in aVDE learning phase.

C. Experimental Results

We evaluate all selected methods by strictly choosing the parameters according to their original papers, and then report the best results of each method. The experimental results of aVDE compared with the 16 baseline methods discussed before on the two pairs of source and target domains are reported in Table III. In Table III, the first column is the number corresponding to the category of the selected methods, the second column indicates method names, the third and forth columns, the fifth and sixth columns, the seventh and eighth columns, the ninth and tenth columns present the recognition results when the RGB-D object dataset or B3DO dataset is used as the source domain and the Caltech-256 dataset or ImageNet dataset is used as the target domain, and the eleventh and twelfth columns give recognition rate when the NYU Depth v1 is used as the source domain and

TABLE III

ACCURACIES (%) FOR OBJECT RECOGNITION AND SCENE CLASSIFICATION WITH SHALLOW AND DEEP FEATURES (BOLD NUMBERS INDICATE THE BEST RESULTS)

Methods		Object → Caltech-256		Object → ImageNet		B3DO → Caltech-256		B3DO → ImageNet		NYU v1→ Scene-15	
		KDES	ImageNet -CNN	KDES	ImageNet -CNN	KDES	ImageNet -CNN	KDES	ImageNet -CNN	KDES	Places -CNN
1	SVM_A	18.21	47.21	26.65	51.76	24.61	47.81	21.80	46.13	17.42	49.46
	1-NN	18.30	48.36	27.27	55.79	27.58	50.00	22.31	49.18	19.78	50.75
2	KCCA	18.39	49.60	34.61	52.69	28.22	51.29	21.63	49.56	19.68	53.33
	SVM2K	20.79	51.72	33.57	54.34	27.84	51.68	23.70	51.33	21.61	53.23
3	SVM+	18.57	48.63	29.86	60.23	25.52	58.63	26.87	47.28	19.46	51.94
	RT	17.15	46.51	23.66	49.79	20.23	46.78	19.65	44.49	16.77	49.03
4	KMM	18.13	47.21	25.21	58.78	23.71	54.51	20.28	48.16	17.53	49.57
	DAM	18.21	49.60	25.41	57.85	24.87	55.28	23.70	49.30	17.10	49.25
	SGF	19.27	50.04	37.81	64.88	27.32	61.63	29.28	49.94	19.25	55.27
	TCA	25.11	56.23	33.47	68.08	28.98	64.69	26.87	55.26	22.04	59.03
	LMK	19.45	52.34	35.23	69.32	33.76	63.79	30.04	51.71	25.81	54.73
	SA	21.13	54.64	36.57	70.35	34.54	54.77	25.48	56.27	27.42	62.69
	GFK	18.48	51.02	41.63	68.70	41.24	61.21	30.16	50.57	24.19	53.23
	UNE	24.76	56.23	42.25	71.90	40.72	64.56	29.78	53.23	26.34	59.68
	DIP	25.46	57.38	41.63	69.21	40.21	60.57	29.91	57.67	25.48	58.60
5	DA-M2S	30.06	61.54	43.49	75.31	**46.26**	68.81	32.70	64.26	31.08	64.52
	aVDE	**35.75**	**70.18**	**50.21**	**80.06**	**46.26**	**69.72**	**34.60**	**68.57**	**33.98**	**69.46**

the Scene-15 dataset is used as the target domain. We test the shallow and deep features on both of these five pairs of datasets. In addition, we also illustrate some samples with highest recognition accuracies from selected datasets in Fig. 4.

From Table III, we observe that our method outperforms all other baseline methods, sometimes by a large margin. It demonstrates the effectiveness of our method by exploring additional depth images in the source domain and reducing the domain distribution mismatch between the source and target domains. Generally, the domain difference between source and target in scene classification (*e.g.* NYU v1→ Scene-15) tasks is larger than the domain difference between source and target in object recognition (*e.g.* Object→ Caltech-256) tasks. From the results of the selected five pairs of datasets for object recognition and scene classification, we can see that not only the accuracy in object recognition has a significant improvement, but also the accuracy in scene classification increases dramatically, which demonstrates the effectiveness of our proposed method in the condition of a larger domain difference. From the results, we find that RT performs the worst possibly because it is based on Rank SVM which is designed for ranking task rather than classification task. SVM_A and 1-NN which do not consider the depth information and domain discrepancy perform poorly. KCCA, SVM2K and SVM+ obtain better performance generally when compared with SVM_A and 1-NN by utilizing the additional depth features. However, these three methods do not reduce the distribution mismatch between the source and target domains. The domain adaptation methods as KMM and DAM perform in a general way or even worse than SVM_A and 1-NN, which maybe because both approaches are unsuitable in this application. SGF, TCA, LMK, SA, GFK, UNE and DIP perform better than other nonadaptation methods, which reveals that considering the domain mismatch across domains is useful. Our proposed aVDE also outperforms DA-M2S which uses privileged information and unsupervised domain adaptation as well. It is possible because the domain mismatch between our shared latent space and the projected target domain is less than the domain mismatch in DA-M2S.

TABLE IV

ACCURACIES (%) FOR OBJECT→IMAGENET AND B3DO→IMAGENET WITH IMAGENET-CNN FEATURES WHICH ARE NOT FINE-TUNED (BOLD NUMBERS INDICATE THE BEST RESULTS)

Methods		Object → ImageNet	B3DO → ImageNet
		ImageNet-CNN (w/o fine-tuning)	ImageNet-CNN (w/o fine-tuning)
1	SVM_A	32.75	27.76
	1-NN	34.19	29.28
2	KCCA	38.64	29.78
	SVM2K	37.19	30.16
3	SVM+	41.32	28.14
	RT	33.88	26.36
4	KMM	42.67	30.54
	DAM	41.63	28.39
	SGF	44.83	29.15
	TCA	45.25	34.47
	LMK	46.07	35.74
	SA	50.21	39.29
	GFK	49.28	34.35
	UNE	51.34	37.77
	DIP	52.79	38.02
5	DA-M2S	55.27	40.81
	aVDE	**59.92**	**44.36**

Additionally, from the comparison of shallow and deep features, we can observe that all deep features have higher classification performances than shallow features. For example, the accuracy of our aVDE method on Object→Caltech-256 classification task increases from 35.75% to 70.18%, which indicates that the deep features can effectively remove the domain bias. It is possible because deep learning models (*i.e.* ImageNet model and Places model) are pre-trained by abundant images which are from different datasets and webs. Note that the proposed method still outperforms other

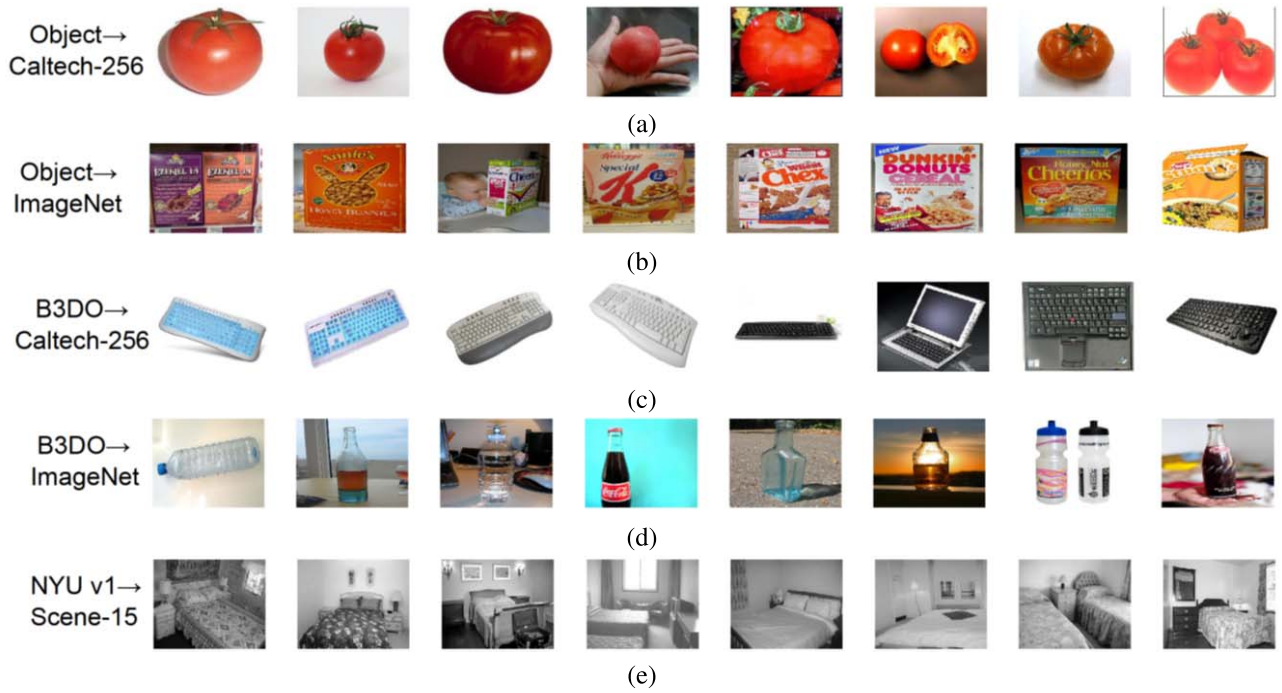

Fig. 4. Example images with highest accuracy results from five selected dataset pairs. (a) tomato, shallow features accuracy = 70.31%, deep features accuracy = 92.86% (b) cereal box, shallow features accuracy = 80.27%, deep features accuracy = 93.77% (c) keyboard, shallow features accuracy = 77.62%, deep features accuracy = 86.57% (d) bottle, shallow features accuracy = 71.29%, deep features accuracy = 85.94% (e) bedroom, shallow features accuracy = 69.78%, deep features accuracy = 89.66%.

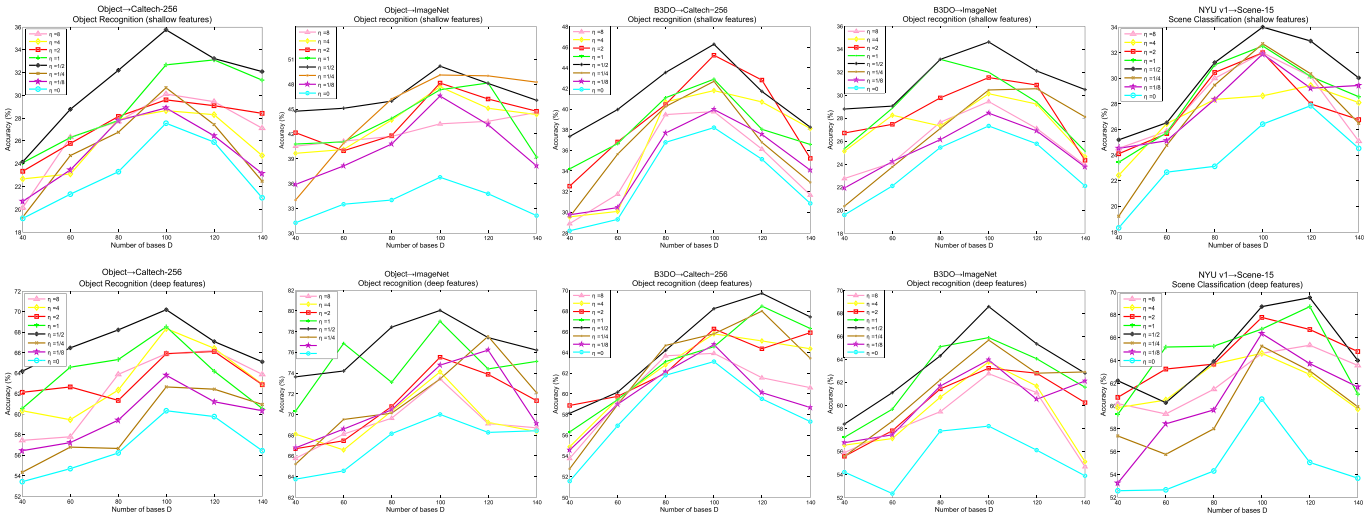

Fig. 5. Parameter sensitivity analysis on the considered datasets with the shallow and deep features.

methods with deep features. Furthermore, the added experiments based on CNN features which are not fine-tuned are reported in Table IV. From Table III and Table IV, we can observe that the performance of the CNN features which are not fine-tuned is worse than that of the fine-tuned ImageNet-CNN features on the two cases: Object→ImageNet and B3DO→ImageNet. On the other hand, the CNN features which are not fine-tuned perform better than the selected shallow features. The result comparison among the selected methods shows a similar rule with the shallow features and fine-tuned CNN features generally. There is a need to note that aVDE still outperforms the selected methods in the condition of the CNN features without fine-tuning.

D. Parameter Sensitivity Analysis

In the proposed aVDE, two parameters D and η are involved for model tuning. We demonstrate the accuracies with different values of D from $\{40, 60, 80, 100, 120, 140\}$ and different values of η from $\{0, 1/8, 1/4, 1/2, 1, 2, 4, 8\}$ on five pairs of datasets with the shallow and deep features in Fig. 5. From Fig. 5, we can find that with the increase of number of bases, the performance of aVDE

TABLE V

COMPARISON OF ACCURACIES (%) BETWEEN aVDE AND TWO SPECIAL CASES

	Object → Caltech-256		Object → ImageNet		B3DO → Caltech-256		B3DO → ImageNet		NYU v1→ Scene-15	
	KDES	ImageNet -CNN	KDES	ImageNet -CNN	KDES	ImageNet -CNN	KDES	ImageNet -CNN	KDES	Places -CNN
aVE	27.67	59.15	41.53	69.01	39.18	63.92	29.91	57.16	27.31	62.04
VDE	22.81	53.58	35.23	60.95	35.95	62.89	26.74	52.85	23.76	55.70
aVDE	**35.75**	**70.18**	**50.21**	**80.06**	**46.26**	**69.72**	**34.60**	**68.57**	**33.98**	**69.46**

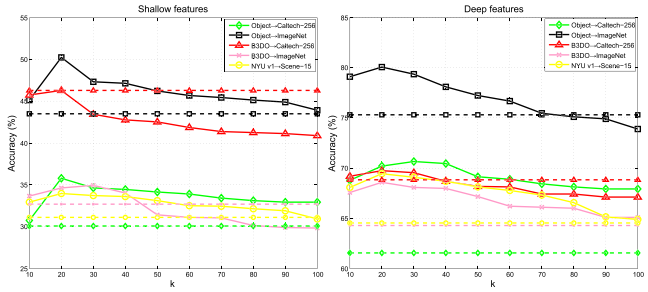

Fig. 6. Parameter k sensitivity analysis on the considered datasets with the shallow and deep features. Dashed lines show the best baseline results.

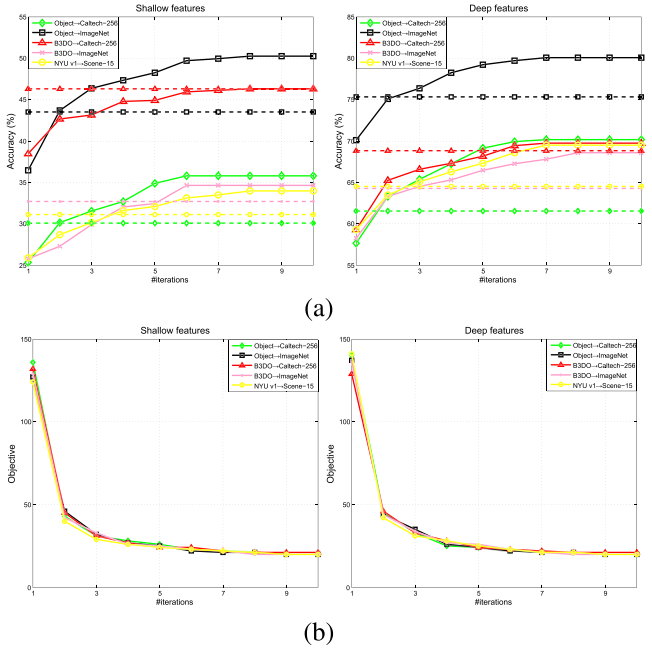

Fig. 7. Convergence study for aVDE on the considered datasets with the shallow and deep features. Dashed lines show the best baseline results. (a) accuracy w.r.t. #iterations. (b) objective w.r.t. #iterations.

(with $\eta \in \{0, 1/8, 1/4, 1/2, 1, 2, 4, 8\}$) becomes better and better until around 100 bases in general. Only B3DO → Caltech-256 (deep features) and NYU v1 → Scene-15 (deep features) achieve the highest points when $\eta = 1/2$ and $D = 120$, and other cases reach the best points when $\eta = 1/2$ and $D = 100$. In addition, when η is zero, the accuracies are lowest which indicates that learning without this regularization leads to poor performance. Therefore, we can conclude that the regularization term is important for our algorithm.

We also run aVDE with different values of k. We plot classification accuracies with different values of $k \in \{10, 20, \cdots, 100\}$ in Fig. 6. From Fig. 6, we can find that when k is small, data reconstruction is accurate in general. Therefore, when comparing with the baseline methods, we set $k = 20$.

E. Convergence Analysis

We evaluate the convergence property of aVDE by experiment. Fig. 7(a) shows that the classification accuracy increases steadily with more iterations and converges within only 10 iterations. Fig. 7(b) shows that the objective function values decrease rapidly at the first few iterations and become stable after about 6 iterations. Both of them show that aVDE converges in a couple of iterations. Therefore, we can draw a conclusion that aVDE is convergent.

F. Analysis on aVDE

We explore two special cases of our aVDE for a better understanding of our algorithm.

Case1: We do not consider depth information, which is denoted as aVE. We remove $\|B - WV\|^2$ in Eq. (3) and $KL(P_B\|Q)$ in Eq. (4) which result in the minimization of another objective function as:

$$\min_{U,V} \|A - UV\|^2 + \frac{\eta}{2} KL(P_A\|Q), \quad s.t.\ U, V \geq 0. \quad (36)$$

Case2: We do not consider domain adaptation, which is denoted as VDE. We directly use the V which is acquired from Eq. (24) to build a NN classifier. Then the embedded representation of the coming RGB target domain data $\hat{\mathbf{a}} \in \mathbb{R}^{M_1 \times 1}$ can be obtained as $\mathbf{v}_{\hat{\mathbf{a}}}$ by Eq. (26).

From Table V, we can find that the results of the special cases are worse than aVDE, which shows it is beneficial to exploit the additional depth features and domain adaptation for learning an adaptive classifier. Moreover, in Case 1, since depth features in the source domain contain additional

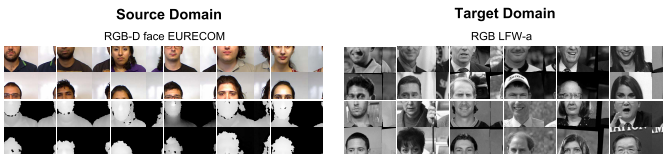

Fig. 8. Some example images from RGB-D face dataset EURECOM and RGB dataset Labeled Faces in the Wild-a (LFW-a).

TABLE VI

ACCURACIES (%) FOR GENDER RECOGNITION (BOLD NUMBERS INDICATE THE BEST RESULTS). RGB-D FACE DATASET EURECOM IS CHOSEN AS THE SOURCE DOMAIN, AND THE RGB DATASET LABELED FACES IN THE WILD-A (LFW-A) IS CHOSEN AS THE TARGET DOMAIN

SVM_A	1-NN	KCCA	SVM2K	KMM	DAM	SGF	TCA	LMK	SA	GFK	UNE	DIP	DA-M2S	aVE	VDE	**aVDE**
64.22 ±1.6	64.53 ±1.82	63.60 ±1.34	67.33 ±1.92	64.25 ±1.43	63.91 ±1.57	67.22 ±1.38	65.24 ±0.88	65.02 ±1.55	67.38 ±1.39	66.78 ±1.73	67.83 ±1.24	64.84 ±4.80	68.44 ±1.44	66.84 ±1.64	64.72 ±1.51	**70.46 ±1.37**

information about shapes and depth, the RGB data in the target domain are projected into the latent space obtained in the visual-depth embedding step, which can help the correction of the noise and make the projected target domain take advantage of the shape and depth information from the source domain. The additional depth features in the source domain can be considered as privileged information for learning the final adaptive classifier. In Case 1, it shows that the performance decreases by 5% to 10% when depth information is not considered.

G. Extension to Gender Recognition

We also extent our aVDE to gender recognition task. In our experiment, the RGB-D face dataset EURECOM [50] is chosen as the source domain, and the RGB dataset Labeled Faces in the Wild-a (LFW-a) [51] is chosen as the target domain. Fig. 8 shows some example images from these two datasets. The EURECOM dataset contains 728 pairs of RGB-D images from 196 females and 532 males. The LFW-a dataset only contains color images with 13144 images from 2960 females and 10184 males. Following the experimental setup in [5], we use the Gradient-LBP features [4] to represent the RGB and depth images for both of the source and target domains in the same way. In addition, 196 male images from EURECOM dataset are randomly sampled to balance the training samples, since male images are much more than female images. 3000 samples are randomly sampled from the target samples for the baseline. According to aVDE, we select the set of parameters which reach the best points in most of the cases for object and scene classification tasks: $\eta = 1/2$, $D = 100$ and $k = 20$. At last, the mean recognition accuracy and the standard deviation are calculated from ten rounds of experiments.

The experimental results of aVDE compared with the baseline methods are reported in Table VI. From Table VI, we can obtain the similar observations as in the object and scene classifications. Our aVDE still outperforms all other baseline methods in gender recognition with a margin from around 2% to 7%, which illustrates the effectiveness of our method again. From the results, SVM_A and 1-NN still perform poorly, since both of them do not consider the depth information and domain discrepancy. SVM2K shows better performance than SVM_A and 1-NN by utilizing the additional depth information. SGF, TCA, LMK, SA, GFK, UNE and DIP perform better than some other nonadaptation methods. aVDE also outperforms DA-M2S which uses privileged information and unsupervised domain adaptation. Moreover, two special cases of our aVDE (aVE and VDE) are also explored in gender recognition. The better performance of aVDE illustrates the benefit to take advantage of the additional depth features and domain adaptation for an adaptive classifier.

V. CONCLUSION

In this paper, we have proposed a novel method aVDE which can utilize the additional depth information in the source domain and simultaneously reduce the domain mismatch between the source and target domains. The latent shared space is identified in Visual-Depth embedding. Aiming to alleviate the mismatch between data distributions, aVDE matches features and reweights instances jointly across the shared latent space and the projected target domain in a principled dimensionality reduction procedure. On five real-world image datasets, the experimental results illustrate that the proposed method significantly outperforms the state-of-the-art methods.

REFERENCES

[1] J. Shen, D. Wang, and X. Li, "Depth-aware image seam carving," *IEEE Trans. Cybern.*, vol. 43, no. 5, pp. 1453–1461, Oct. 2013.

[2] Z. Cai, J. Han, L. Liu, and L. Shao, "RGB-D datasets using microsoft Kinect or similar sensors: A survey," *Multimedia Tools Appl.*, vol. 76, no. 3, pp. 4313–4355, 2017.

[3] L. Shao, Z. Cai, L. Liu, and K. Lu, "Performance evaluation of deep feature learning for RGB-D image/video classification," *Inf. Sci.*, vols. 385–386, pp. 266–283, 2017.

[4] T. Huynh, R. Min, and J.-L. Dugelay, "An efficient LBP-based descriptor for facial depth images applied to gender recognition using RGB-D face data," in *Proc. Asian Conf. Comput. Vis.*, 2012, pp. 133–145.

[5] L. Chen, W. Li, and D. Xu, "Recognizing RGB images by learning from RGB-D data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1418–1425.

[6] Y. Huang, F. Zhu, L. Shao, and A. F. Frangi, "Color object recognition via cross-domain learning on RGB-D images," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2016, pp. 1672–1677.

[7] M. Gong, K. Zhang, T. Liu, D. Tao, C. Glymour, and B. Schölkopf, "Domain adaptation with conditional transferable components," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 2839–2848.

[8] J. Huang, A. Gretton, K. M. Borgwardt, B. Schölkopf, and A. J. Smola, "Correcting sample selection bias by unlabeled data," in *Proc. Adv. Neural Inf. Process. Syst.*, 2006, pp. 601–608.

[9] B. Geng, D. Tao, and C. Xu, "DAML: Domain adaptation metric learning," *IEEE Trans. Image Process.*, vol. 20, no. 10, pp. 2980–2989, Oct. 2011.

[10] Z. Cui, W. Li, D. Xu, S. Shan, X. Chen, and X. Li, "Flowing on Riemannian manifold: Domain adaptation by shifting covariance," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2264–2273, Dec. 2014.

[11] L. Bo, X. Ren, and D. Fox, "Depth kernel descriptors for object recognition," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, Sep. 2011, pp. 821–826.

[12] S. Gupta, P. Arbeláez, R. Girshick, and J. Malik, "Indoor scene understanding with RGB-D images: Bottom-up segmentation, object detection and semantic segmentation," *Int. J. Comput. Vis.*, vol. 112, no. 2, pp. 133–149, 2015.

[13] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *J. Mach. Learn. Res.*, vol. 5, pp. 1457–1469, Dec. 2004.

[14] D. Cai, X. He, X. Wu, and J. Han, "Non-negative matrix factorization on manifold," in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2008, pp. 63–72.

[15] B. Shen and L. Si, "Non-negative matrix factorization clustering on multiple manifolds," in *Proc. AAAI Conf. Artif. Intell.*, 2010, pp. 575–580.

[16] A. Gretton, K. M. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola, "A kernel method for the two-sample-problem," in *Proc. Adv. Neural Inf. Process. Syst.*, 2006, pp. 513–520.

[17] M. Masaeli, J. G. Dy, and G. Fung, "From transformation-based dimensionality reduction to feature selection," in *Proc. Int. Conf. Mach. Learn.*, 2010, pp. 751–758.

[18] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Trans. Neural Netw.*, vol. 22, no. 2, pp. 199–210, Feb. 2011.

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

[20] B. Gong, K. Grauman, and F. Sha, "Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. I-222–I-230.

[21] T. Wang, T. Ye, and C. Gurrin, "Transfer nonnegative matrix factorization for image representation," in *Proc. Int. Conf. Multimedia Model.*, 2016, pp. 3–14.

[22] I. Redko and Y. Bennani, "Non-negative embedding for fully unsupervised domain adaptation," *Pattern Recognit. Lett.*, vol. 77, pp. 35–41, Jul. 2016.

[23] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2001, pp. 556–562.

[24] W. Zheng, Y. Qian, and H. Tang, "Dimensionality reduction with category information fusion and non-negative matrix factorization for text categorization," in *Proc. Int. Conf. Artif. Intell. Comput. Intell.*, 2011, pp. 505–512.

[25] D. Cai, X. He, J. Han, and T. S. Huang, "Graph regularized nonnegative matrix factorization for data representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1548–1560, Aug. 2011.

[26] D. Cai, X. He, and J. Han, "Spectral regression for efficient regularized subspace learning," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2007, pp. 1–8.

[27] P. H. Schönemann, "A generalized solution of the orthogonal procrustes problem," *Psychometrika*, vol. 31, no. 1, pp. 1–10, 1966.

[28] X. Zhang, F. X. Yu, R. Guo, S. Kumar, S. Wang, and S.-F. Chang, "Fast orthogonal projection based on kronecker product," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 2929–2937.

[29] F. Nie, H. Huang, X. Cai, and C. H. Ding, "Efficient and robust feature selection via joint $\ell_{2,1}$-norms minimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 1813–1821.

[30] Y. Yang, H. T. Shen, Z. Ma, Z. Huang, and X. Zhou, "$\ell_{2,1}$-norm regularized discriminative feature selection for unsupervised learning," in *Proc. Int. Joint Conf. Artif. Intell.*, 2011, vol. 22. no. 1, pp. 1589–1594.

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

[32] K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi-view RGB-D object dataset," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 1817–1824.

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

[34] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[35] A. Janoch *et al.*, "A category-level 3D object dataset: Putting the Kinect to work," in *Proc. Consum. Depth Cameras Comput. Vis.*, 2013, pp. 141–165.

[36] N. Silberman and R. Fergus, "Indoor scene segmentation using a structured light sensor," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Nov. 2011, pp. 601–608.

[37] L. Fei-Fei and P. Perona, "A Bayesian hierarchical model for learning natural scene categories," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2. Jun. 2005, pp. 524–531.

[38] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor, "Canonical correlation analysis: An overview with application to learning methods," *Neural Comput.*, vol. 16, no. 12, pp. 2639–2664, 2004.

[39] J. Farquhar, D. Hardoon, H. Meng, J. S. Shawe-taylor, and S. Szedmak, "Two view learning: SVM-2K, theory and practice," in *Proc. Adv. Neural Inf. Process. Syst.*, 2005, pp. 355–362.

[40] V. Vapnik and A. Vashist, "A new learning paradigm: Learning using privileged information," *Neural Netw.*, vol. 22, nos. 5–6, pp. 544–557, 2009.

[41] V. Sharmanska, N. Quadrianto, and C. H. Lampert, "Learning to rank using privileged information," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 825–832.

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

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

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

[45] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[46] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. ACM Int. Conf. Multimedia*, 2014, pp. 675–678.

[47] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 487–495.

[48] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.

[49] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik, "Learning rich features from RGB-D images for object detection and segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 345–360.

[50] R. Min, N. Kose, and J.-L. Dugelay, "KinectFaceDB: A Kinect database for face recognition," *IEEE Trans. Syst., Man, Cybern. A, Syst.*, vol. 44, no. 11, pp. 1534–1548, Nov. 2014.

[51] L. Wolf, T. Hassner, and Y. Taigman, "Effective unconstrained face recognition by combining multiple descriptors and learned background statistics," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 10, pp. 1978–1990, Oct. 2011.