Active-Learning-Incorporated Deep Transfer Learning for Hyperspectral Image Classification

Jianzhe Lin, *Student Member, IEEE*, Liang Zhao, Shuying Li, Rabab Ward, *Fellow, IEEE*, and Z. Jane Wang, *Fellow, IEEE*

***Abstract*—A hyperspectral image (HSI) includes a vast quantity of samples, a large number of bands, and randomly occurring redundancy. Classifying such complex data is challenging, and its classification performance can be affected significantly by the amount of labeled training samples, as well as the quality, position, and others factors of these samples. Collecting such labeled training samples is labor and time consuming, motivating the idea of taking advantage of labeled samples from other pre-existing related images. Therefore, transfer learning, which can mitigate the semantic gap between existing and new HSIs, has drawn increasing research attention. However, existing transfer learning methods for HSIs (which mainly concentrate on how to overcome the divergence among images) may fail to carefully consider the contents to be transferred and thus limit their performances. In this paper, we present two novel ideas: 1) we, for the first time, introduce an active learning process to initialize the salient samples on the HSI data, which would be transferred later; and 2) we propose constructing and connecting higher level features for the source and target HSI data to further overcome the cross-domain disparity. Different from existing methods, the proposed framework requires no *a priori* knowledge on the target domain, and it works for both homogeneous and heterogeneous HSI data. Experimental results on three real-world HSIs support the effectiveness of the proposed method for HSI classification.**

***Index Terms*—Hyperspectral image (HSI), salient samples, supervised classification, transfer learning.**

I. Introduction

SUPERVISED hyperspectral image (HSI) classification has long been investigated for HSIs and generally can provide satisfying performance when enough training samples are available and both training and testing data are on the same feature space. However, such assumptions may not always hold true. Obtaining sufficient training samples for the newly collected HSI data could be time and labor consuming. Considering that an HSI includes a vast quantity of samples, a large number of bands, and close relations between them, this training sample requirement is even more demanding. To address this concern, recently, researchers have resorted to using pre-existing related HSIs as auxiliary information to exploit the prior knowledge for classification of newly collected ones. However, a concern is the semantic disparity between the auxiliary and objective HSIs.

Arduous efforts have been made to tackle this problem, and probably, the most prevalent solution is the transfer-learning-based approach. A general framework is to transfer the HSI data from the source domain (the auxiliary HSI) and the target domain (the objective HSI) to a common subspace to overcome the cross-domain semantic disparity. This semantic disparity is large among HSI data. Due to different acquisition conditions and sensors, the spectra observed on a new scene can be quite different from the existing one even if they represent the same type of objects [1], [2]. Therefore, transfer learning in many cases may not provide decent results. How to reduce the difference between the data while preserving the original data characteristics remains a challenging issue, especially for heterogeneous transfer learning whose data on the source and target domains are with different dimensionality in different feature spaces.

One major issue for heterogeneous transfer learning is the distribution divergence and the feature bias between the two domains. Existing transfer learning methods that adopt linear or nonlinear kernel functions to transfer the data on both domains to a common space to bridge this cross-domain gap may not be effective enough. Recently, researchers reported that deep neural networks (DNNs) that exploit high-level features of data can facilitate the minimization of this semantic gap. In the high-level feature space, the data from both domains are more likely to have less differences and bias.

A representative direction is the stacked autoencoder (SAE)-based methods, which search for high-level common features across domains as inputs for the following supervised classification. However, as most of the state-of-the-art methods do not consider the relationship between the source and target domains layer by layer in the deep network, the data bias between the two domains could accumulate at each layer. Another limitation is that the current SAE does not address the problem "what to transfer" since most of the current methods only transfer the

Manuscript received April 24, 2018; revised July 5, 2018 and September 20, 2018; accepted September 22, 2018. This work was supported by the Qatar National Research Fund (a member of the Qatar Foundation) through the National Priorities Research Program under Grant 7-684-1-127. *(Corresponding author: Shuying Li.)*

J. Lin and Z. J. Wang are with the Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC V6T 1Z2, Canada, and also with the School of Information Science and Technology, Northwest University, Xi'an 710069, China (e-mail: jianzhelin@ece.ubc.ca; zjanew@ece.ubc.ca).

L. Zhao is with the School of Software Technology, Dalian University of Technology, Dalian 116023, China (e-mail: matthew1988zhao@mail.dlut.edu.cn).

S. Li is with the School of Automation, Xi'an University of Post and Telecommunications, Xi'an 710121, China (e-mail: angle_lisy@163.com).

R. Ward is with the Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC V6T 1Z2, Canada (e-mail: rababw@ece.ubc.ca).

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

Digital Object Identifier 10.1109/JSTARS.2018.2874225

randomly selected training samples from the source domain to the target domain, which would introduce noises. Current methods could introduce a low-rank framework [3] to overcome the noise issue during the transfer process. However, this way could not get rid of the noise fundamentally, while selecting the content to transfer could address the noise issue more effectively.

To address the above-mentioned noise problem, this paper proposes a deep-mapping-based heterogeneous transfer learning model via querying salient examples (referred as DTSE). First, salient samples on both source and target domains are queried actively. These informative samples can at the great extent explore the structure information of the data on each domain and may have low correlation with each other. Then, these salient samples on both the domains are used to construct their own autoencoders with multiple domain-based layers. The source and target domains layer by layer in this deep network are correlated by the canonical correlation analysis (CCA). Such correlations further propagate back and fine-tune the layers of the network. The forward- and backpropagating iterates until the final output features of these two autoencoders have lowest divergence and the correlation is maximized. Our contributions can be summarized as follows.

1) To our knowledge, this is the first attempt to using deep active transfer learning for HSI classification, which explores the high-level feature correlation between remote sensing images with different dimensions and from different sensors.
2) We design a query principle that searches salient samples for every class of the training samples on both source and target domains. Such salient training samples are the most informative for representing their corresponding classes. Salient samples should also be prominent and sparse enough with low correlations [4] and thus facilitate more robust and informative transfer learning.
3) We propose a new principle for fine-tuning the neural network for autoencoder on the HSI dataset. During the training process, the autoencoders on both domains are set up, and high-level features are explored on both domains. The correlation between both domains is sought by CCA layer by layer, and the neural networks on both domains are also further fine-tuned by the CCA restriction. The final common feature space is found after this iterative fine-tuning process.

II. Related Work

In this section, two general lines of learning processes are reviewed: active learning and deep transfer learning. We also point out a few existing problems in the literature that would be addressed in this paper.

A. Active Learning

Sample selection is a promising novel front of HSIs to improve the inferior quality of training samples [5]–[7]. The main purpose of sample selection is to locate the most valuable data from the HSI to benefit the supervised training process of classification or segmentation [8]. Unlike traditional supervised methods, which generally set parameters based on randomly selected training samples, active learning is an effective sample selection approach that gives the learner the freedom to augment the training data based on specific designed criteria. This process is also named query, to label [9]. Two popular criteria used in selecting valuable sample are informativeness and representativeness of data samples.

The first widely acknowledged category of active learning is querying the most informative samples. One representative method is uncertainty sampling, which can query the samples with lowest certainty [10]–[12]. However, this method may fail due to the existence of outliers. Query-by-committee [13]–[15], as another well-known exemplar approach, measures the informativeness of samples by calculating the degree of agreement with several variants, which are named committees. A following work, the error-reduction-based sampling [16], uses sampling estimation to estimate the future error directly to solve the efficiency problem. The batch mode [17] that selects multiple instances in every iteration further improves the efficiency.

The second branch of active learning is querying the most representative samples that exploit the distribution of unlabeled data efficiently. One popular scheme is the coarse-to-find strategy: first cluster the initial data and choose the representative samples to manually label them and, then, propagate the learnt decision [18]. Hierarchical clustering, as representative work, can detect and exploit the clusters whose structures are loosely aligned with class labels [19]. Chattopadhyay *et al.* [20] propose a novel criterion to concurrently select a set of query samples by directly minimizing the difference between distributions of labeled and unlabeled data.

The third category of active learning is querying samples, which are both most informative and representative. A typical approach is to seek for data points, which are hard to predict and at the same time representative enough to explore the distribution information of testing data [21]. The uncertainty and the density of data are dynamically balanced to acquire the optimal candidates of training samples [22]. Huang *et al.* [9] noted that the representativeness of samples may be enhanced if considering the distributions of both labeled and remaining unlabeled data. However, a concern is that the informativeness and representativeness are measured, respectively, and balancing them may lead to a suboptimal solution.

Existing AL frameworks for transfer learning on HSIs [23], [24], as well as other types of images [25], [26], are all based on the first category, querying the most informative samples. In [27] and [28], the proposed frameworks iteratively select the most informative samples for defining a training set by exploiting the HSI classification rule. These incremental training procedures search for the underlying queries solely based on the available small number of labeled examples and fail to consider the representativeness of the actively learnt samples, which might lead to samples bias. Different from such existing works, the main assumption of our work is that the informativeness and representativeness criteria are not independent from each other, and we should consider them simultaneously. The representativeness controls the macrostructure of data, the informativeness reflects the microfeatures, and they should be correlated.

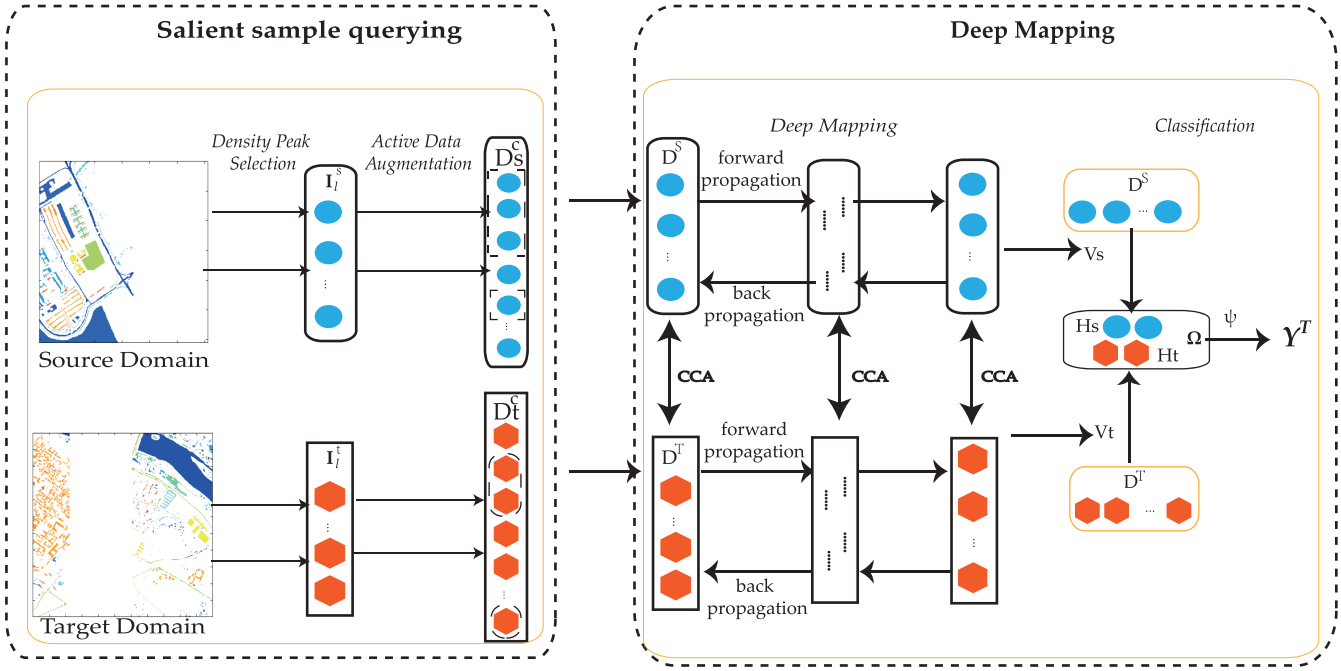

Fig. 1. Flowchart of the proposed DTSE framework for HSI classification.

This is the main motivation of the proposed active learning framework.

B. Deep Transfer Learning

DNNs composed of multiple nonlinear transformations can learn a better feature representation than traditional shallow models [29]–[31]. Recently, the concept of deep learning was incorporated into transfer learning to uncover high-level correlation spaces of the source and target domains. For homogeneous cases, Glorot *et al.* [32] introduce the stacked denoising autoencoder (SDA) to learn robust features. Based on this baseline, more followup works for homogeneous transfer learning were proposed. By considering the scalability with high-dimensional features of the SDA, the marginalized SDA (mSDA) was proposed [33]. In [34], a task-driven deep transfer learning model is proposed, in which the deep feature and the classifier are obtained simultaneously, and more discriminative features are generated. In a more recent work [35], the label information is further encoded and exploited. For the layer-to-layer correlation principle, CCA is used as the baseline, as it can maximize correlations between domains by deriving the projection subspace as a joint representation [36]–[38].

The heterogeneous transfer learning case is a more general challenging topic, as most of the data around us have different distributions and dimensions. This problem is more obvious on HSIs as most of the HSIs are collected by different sensors under various conditions. Existing work for heterogeneous transfer learning can be found in [39] without deep features. Thus, there is still space for improving the classification performance. A direct assumption for heterogeneous deep transfer learning is to exploit and unify the deep features from the source and target domains with the aid of labeled source datasets or co-occurrence datasets. Related typical work can be found in [40], which is based on an extension of mSDA, where the domains are bridged by the co-occurrence instances given in advance. Another recent typical work is the generalized deep transfer network [41] for knowledge propagation in heterogeneous domains, which generates weakly shared representations and parameters to exploit the rich cross-domain information for transfer learning. The correlation of parameters in DNNs on both domains is calculated and shown to be effective. However, with millions of parameters in deep networks of HSIs, negative transfer [42], [43] as well as serious overfitting problems may ruin the transfer learning performance.

Here, we propose exploiting the cross-domain correlation in the DNN layer by layer on the HSI. We actively select salient samples to train the neural network to avoid negative transfer. The two heterogeneous domains are more closely related through layer-by-layer correlating, mapping, and fine-tuning. Domain-specific networks and shared interdomain representation are thus jointly learnt.

III. Proposed Framework

The major components of the proposed DTSE framework are illustrated in Fig. 1, including the salient sample querying (SSQ) and deep mapping (DM). First, salient samples I_l^s in the source domain and I_l^t in the target domain are queried. The sample pairs with the same class label on both domains are recognized as co-occurrence data, and they are further used for the fine-tuning process of DM. Second, we exploit deep features belonging to the data of both domains, and the DNN is fine-tuned based on the correlation constructed by the co-occurrence data. We now elaborate the two parts in the following subsections.

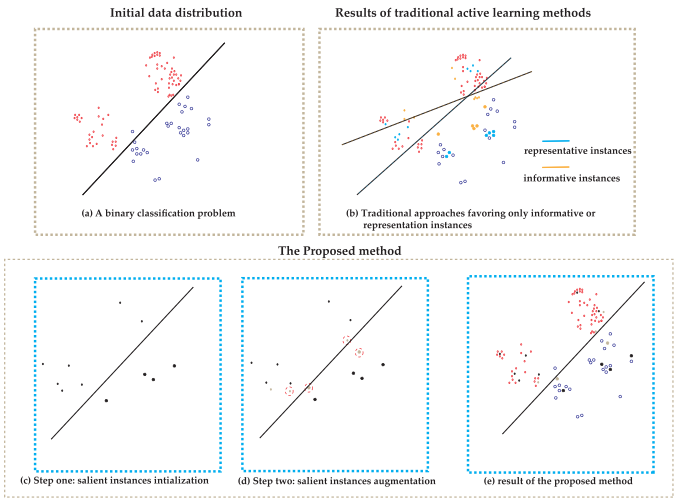

Fig. 2. Illustrative comparison of the HSI sample selection based on different criteria.

A. Density Peak Selection

Fig. 2 shows a synthesized example that emphasizes the importance of SSQ in HSI classification. Fig. 2(a) shows a binary classification problem, in which two kinds of instances are represented by different legends. The objective is to query 1% instances on the two datasets (datasets on the source and target domains) to generate the correlation model. The results of traditional active learning methods are illustrated in Fig. 2(b). In comparison, the results of the proposed method are shown step by step in Fig. 2(c)–(e). As indicated in Fig. 2(b), the approach favoring the informative instances, represented by the yellow legend, tends to select the samples with the highest uncertainty and thus may lead to sample bias; the approach favoring the representative instances, representative by the blue legend, mainly considers the data structure, which ignores the details, and thus is likely to result in errors especially for the instances near the boundary.

The proposed SSQ framework is illustrated in Fig. 2(c)–(e). Each of the two major steps shown in Fig. 2(c) and (d) for SSQ has its own effect, but they should be taken as an integrated one. To be more specific, in Fig. 2(c), we first initialize the selected salient instances by searching for density peaks, which consider both the overall structure and local details, and these points should be the most representative ones. Then, in Fig. 2(d), we actively augment the instances to 1% instances, which would be used for the supervision of the following fine-tuning for the deep network. We sequentially select the most informative data by the min–max criterion [9] based on the formerly learnt representative ones. At last, based on the selected training instances, we finally get classification results for all data, as shown in Fig. 2(e), which has higher accuracy when compared with results shown in Fig. 2(b).

We further formulate the problem as follows. Suppose the HSI data are denoted by $I = \{(x_1, y_1), (x_2, y_2), ...(x_{n_l}, y_{n_l}), x_{n_l+1}, \ldots, x_n\}$, consisting of n_l labeled points and $n_u = n - n_l$ unlabeled ones. x_i represents a pixel in the HSI, which is a d-dimensional vector and $y_i \in \{-1, +1\}$ is the label of x_i. In our method, we first denote the current labeled samples by I_l. We also represent the unlabeled ones by $I_a = I_u \cup \{x_a\}$, including the unlabeled ones I_u and the current salient pixel x_a, which is selected based on the active learning framework [9]. Their corresponding labels are denoted by $Y = \{Y_l, y_a, Y_u\}$, in which $Y_a = \{y_a, Y_U\}$ is assigned after the learning process.

When selecting salient examples, we need to consider both the overall structure of the data and the local details. One popular strategy is based on clustering, which selects cluster centers as the salient ones. This strategy traditionally barely takes the overall data distribution into consideration, neglecting the local distribution of data. As a result, it can just find the representative data, while their informativeness is ignored. To overcome this drawback, we propose the following density peaks based on the sample selection idea.

1) Global Density and Local Peaks: We first introduce two concepts: *global density* and *local peaks*.

To find the instance representatives, we first calculate the density of each instance. For a certain radius r_i centering on the instance x_i, the number of neighboring instances is the *global density* d_i of x_i. The instances with higher global densities are more likely to be selected.

However, one problem exists in this density-based selection: We are likely to select instances with similar characteristics. Several neighboring instances with the similar highest global densities would be chosen based on the global density. Choosing these instances would decrease other instances' chances of being selected even if they are informative and representative. Therefore, another concept, *local peaks*, is explored. Although it is difficult to define the local area considering each instance as an individual, it is much easier if we take each two instances as a *group* and compare their global density and the distance between them. To be more specific, besides d_i of x_i, we assign another characteristic to each instance, the codistance s_i, by considering the instance group. We first rank the density of each instance and rearrange them as $X_d = \{x_{d_1}, x_{d_2}, \ldots, x_{d_n}\}$, where x_{d_1} has the highest density and x_{d_n} the lowest. The distance between x_{d_i} and $x_{d_{i+1}}$ is defined as s_i. In this group, the density of x_{d_i} is a little higher than, but the most proximate to, $x_{d_{i+1}}$. After getting s_i, we can define that *local peaks* as with both the highest global density d and codistance s. Local peaks will never be neighboring as they need to be with high codistance. The final instances selected are these *local peaks*.

2) Active Local Peak Querying: To find the local peaks, we introduce the objective function as

$$a^* = \arg\max_{n_l < a < n} (s_a + \lambda d_a) \tag{1}$$

where a^* is the label of the selected instance, s_a and d_a are the codistance and the density of instance a, respectively, and λ is the coefficient between the two terms. However, for this objective function, the global optimal λ is hard to be found considering all the instances. Therefore, we plan to obtain the solution for this problem actively [4], which can skip the step of optimization of λ. Equation (1) is our criterion for selecting

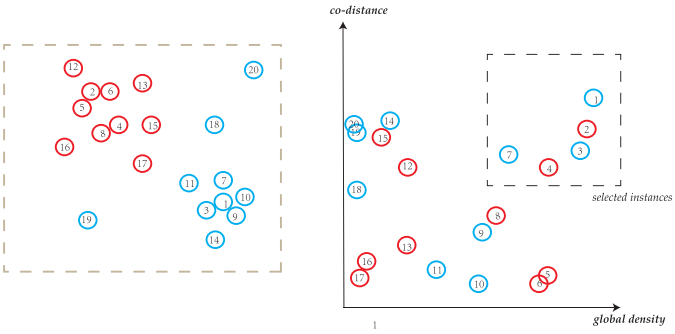

Fig. 3. Example for the active local peak querying.

the instances. We show a simple example in Fig. 3 for actively obtaining a solution with randomly chosen 20 instances.

In Fig. 3, all data in the left figure are first put on the coordinate system, as shown in the right figure, where the x-axis represents the global density d and the y-axis represents the codistance s. Here, we have two categories of data, indicated by red and blue. Suppose we need to select 15% training data in total. We note that samples like x_{d_5} and x_{d_6}, which have quite high global density but low codistance and might be chosen by traditional clustering methods, will not be selected in our framework, since they are quite similar to x_{d_2}, while x_{d_7} whose overall density is not high enough will be selected, since it is quite informative: from the perspective of global density, it is quite similar with the red type; from the perspective of location, it is quite close to the blue type and belongs to this type. Therefore, we think this point has quite high uncertainty. Finally, x_{d_1}, x_{d_3}, and x_{d_7} are selected actively. We get $I_l = \{(x_{d_1}, y_{d_1}), (x_{d_3}, y_{d_3}), (x_{d_7}, y_{d_7})\}$, where I_l means the labeled training data.

We note that not all instances can be successfully introduced into this model. Two extreme cases during the querying process may occur. The first case is that, for many instances, the densities are zero, as they may have no nearing neighbors. Suppose that such instances are denoted by $X_n = x_{n_1}, x_{n_2}, \ldots, x_{n_p}$. For x_{n_i}, suppose that the Euclidean distance between x_{n_i} and its nearest neighboring instance is $e_{x_{n_i}}$; we define the density of x_{n_i} to be $d_{x_{n_i}} = \frac{1}{d_{x_{n_i}}}$; we find $d_{x_{n_i}}$ would be quite small and x_{n_i} will not be chosen finally. The second case is that, for x_{d_1} with the highest global density but zero codistance (as no other sample would be with higher global density than x_{d_1}), we heuristically assign s_{d_1} with the highest value because it must be selected.

B. Active Data Augmentation

Though the above-mentioned querying process considers the informativeness of each instance in a local area, to find the most informative instances for learning the final classification model, the already labeled few instances are not enough. Therefore, after getting the labeled data by the density peak querying, we would select the current pixel x_a that is based on the active learning framework [9] via a sequential augmentation process. By augmenting the data, the ultimate goal is to learn the best classification model. Since support vector machine (SVM) is employed in the learning process, we first review SVM.

1) Brief Review for SVM: The key of the traditional SVM is to learn an optimal hyperplane to separate the labeled instances with the maximum marginal. Taking advantage of the kernel, SVM finds the hyperplane

$$f(x) = \omega^T x + b \tag{2}$$

by solving the optimization problem

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2}\|w\|^2 \\ \text{s.t.} \quad & y_i(w^T x_i + b) \geq 1, \quad i = 1, 2, ..., n. \end{aligned} \tag{3}$$

Here, we do not describe the slack variable in the traditional SVM as it is not relevant to our current major problem. By introducing the Lagrange multiplier α, we can further transfer (4) to the Lagrange function as

$$\varphi(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_i^n \alpha_i (y_i(w^T x_i + b) - 1). \tag{4}$$

By maximizing this function, the optimal f can be learnt as

$$\theta(w) = \min_{\alpha_i \geq 0} \varphi(w, b, a). \tag{5}$$

As $w = \sum_{i=1}^n \alpha_i y_i x_i$, we have $f(x) = \sum_{i=1}^n \alpha_i y_i \langle x_i, x \rangle + b$. We hold the view that the restriction $\min \frac{1}{2}\|w\|^2$ would be realized if $\min_{f \in \mathscr{H}} \frac{1}{2}|f|^2_{\mathscr{H}}$ is obtained. Therefore, we further rewrite (5) as

$$\theta(x) = \min_{f \in \mathscr{H}} \frac{1}{2}|f|^2_{\mathscr{H}} + \sum_{i=1}^n l\left(y_i, f\left(x_i\right)\right) \tag{6}$$

where $\mathscr{H}$ is a reproducing kernel Hilbert space, and $l(y, f(x))$ is the loss function.

2) Framework for Data Augmentation: To motivate the following framework, we separate x in (5) into two sets: I_l and x_a. Moreover, to identify the most informative example, we consider the worst case for analysis by selecting the unlabeled instance a that leads to a small value for the objective function [see (7)] regardless of its assigned class label y_a. To achieve this goal, we consider the new objective function as

$$\begin{aligned} \theta(I_l, x_a) = & \max_{y_a \in \{-1, +1\}} \min_{f \in \mathscr{H}} \frac{1}{2}|f|^2_{\mathscr{H}} + \sum_{i=1}^{n_l} l\left(y_i, f\left(x_i\right)\right) \\ & + l(y_a + f(x_a)). \end{aligned} \tag{7}$$

In order to find the most informative instances, we have

$$a = \arg\min_{n_l < a < n} \theta(I_l, x_a). \tag{8}$$

By this way, we are more likely to select the instance closest to the decision boundary, and x_a tends to be more informative. We select the new unlabeled instances sequentially by this way until the co-occurrence data are augmented to achieve the expected percentage.

C. DM Mechanism

The formerly chosen salient samples I_l^s in the source domain are denoted as $D_S^C = \{C_i^S\}_{i=1}^{n_s}$, and I_l^t in the target domain

are denoted $D_T^C = \{C_i^T\}_{i=1}^{n_t}$. The labeled data on the source domain, denoted as $D^S = \{X_i^S, Y_i^S\}_{i=1}^{n_s}$, are used to supervise the DM-based classification. The unlabeled data on the target domain are demoted as $D^T = \{X_i^T\}_{i=1}^{n_t}$. The deep network in the source domain is denoted by $\Theta^S = \{W^S, b^S\}$. The deep network in the target domain is denoted by $\Theta^T = \{W^T, b^T\}$. The common subspace is represented by Ω, and the final classifier is represented by Ψ. The labeled data D^S from the source domain are used to predict the label of D^T by applying $\Psi(\Omega(D^T))$.

Inspired by CCA, which can maximize the correlation between two domains, we apply the CCA within both DNNs Θ^S and Θ^T to construct a multilayer correlation model. As shown in Fig. 1, first, a DNN is set up in the source domain and another in the target domain by forward propagation based on the co-occurrence data C^S and C^T. The correlation coefficients between the hidden layers of these two domains are found using CCA. After setting up Θ^S in the source domain and Θ^T in the target domain, the high-level common subspace is finally obtained. D^S and D^T are both projected to the common subspace, on which the labeled D^S are used for training and predicting the labels of D^T. A more detailed mathematic formulation is shown as follows.

We employ the SAEs in the source domain Θ^S and in the target domain Θ^T. For the hidden layers of Θ^S and Θ^T, the hidden features A^S and A^T can be represented by

$$A^S(n+1) = f(W^S(n) \times A^S(n) + b^S(n)), \quad n > 1$$
$$A^S(n) = f(W^S(n) \times C^S + b^S(n)), \quad n = 1 \tag{9}$$
$$A^T(n+1) = f(W^T(n) \times A^T(n) + b^T(n)), \quad n > 1$$
$$A^T(n) = f(W^T(n) \times C^T + b^T(n)), \quad n = 1. \tag{10}$$

Here, W^S and b^S are parameters for the neural network Θ^S, and W^T and b^T are parameters for the neural network Θ^T. $A^S(n)$ and $A^T(n)$ mean the co-occurrence nth hidden layers in the source domain and in the target domain, respectively. The correlation matrices $V^S(n)$ and $V^T(n)$ project features of D^S and D^T to a correlating common subspace Ω. These $V^S(n)$ and $V^T(n)$ are obtained by CCA. Therefore, to set up the optimal neural networks in the source domain and in the target domain, we need to satisfy two objectives: to minimize the reconstruction error of the neural network in the source domain and error of the neural network in the target domain, and to maximize the correlation between the two neural networks. This objective function can be formulated as

$$\min J = J_S(W^S, b^S) + J_T(W^T, b^T) - \Gamma(V^S, V^T) \tag{11}$$

where $J_S(W^S, b^S)$ and $J_T(W^T, b^T)$ are the reconstruction errors in the source domain and in the target domain, respectively, which are defined as follows:

$$J_S(W^S, b^S) = \left[\frac{1}{m}\sum_{i=1}^{m}\left(\frac{1}{2}||h_{W^S,b^S}(C_i^S) - C_i^S||^2\right)\right] + \frac{\lambda}{2}\sum_{l=1}^{n^s-1}\sum_{j=1}^{n_l^S}\sum_{k=1}^{n_{l+1}^S}\left(W_{kj}^{S(l)}\right)^2 \tag{12}$$

$$J_T(W^T, b^T) = \left[\frac{1}{m}\sum_{i=1}^{m}\left(\frac{1}{2}||h_{W^T,b^T}(C_i^T) - C_i^T||^2\right)\right] + \frac{\lambda}{2}\sum_{l=1}^{n^T-1}\sum_{j=1}^{n_l^T}\sum_{k=1}^{n_{l+1}^T}\left(W_{kj}^{T(l)}\right)^2 \tag{13}$$

where $h_{W^S,b^S}(C_i^S)$ and $h_{W^T,b^T}(C_i^T)$ are the output results of the two neural networks, n^S and n^T are the number of their layers, n_l^S and n_l^S are the number of neurons in layer l, and λ is the tradeoff parameter.

The third term $\Gamma(V^S, V^T)$ in (11) is the correlation matching matrix between the source domain and the target domain. The objective is to optimize the correlation matrices V^S and V^T by maximizing the correlation between the source-domain data and the target-domain data, which is defined as

$$\Gamma(V^S, V^T) = \sum_{l=2}^{n^S-1} \frac{V^{S(l)^T} \sum_{ST} V^{T(l)}}{\sqrt{V^{S(l)^T} \sum_{SS} V^{S(l)}}\sqrt{V^{T(l)^T} \sum_{TT} V^{T(l)}}} \tag{14}$$

where $\sum_{ST} = A^{S(l)}A^{T(l)^T}$, $\sum_{SS} = A^{S(l)}A^{S(l)^T}$, and $\sum_{TT} = A^{T(l)}A^{T(l)^T}$. By minimizing (11), we can collectively train the two neural networks $\theta^T = \{W^T, b^T\}$ and $\theta^S = \{W^S, b^S\}$.

After constructing the multiple layers of the networks by (11), a final CCA is employed at the top layer to fine-tune both the neural networks in the source and target domains by backpropagation, and the high-level common subspace can be obtained. On such a common subspace, the classification of the unlabeled D^T can be conducted under the supervision of the labeled D^S.

D. Classification on the Common Semantic Subspace

The final classification is performed on the common subspace Ω. The unlabeled data on the target domain D^T and the labeled D^S are both projected to the common subspace Ω by the correlation coefficients $V^S(n_S)$ and $V^T(n_T)$. The projection is formulated as $H^S = A^S(n_S)V^S(n_S)$ and $H^T = A^T(n_T)V^T(n_T)$. The standard SVM algorithm is applied on Ω. The classifier Ψ is trained by $\{H_i^S, Y_i^S\}_i^{n_S}$. This trained classifier Ψ is applied to D^T as $\Psi(H^T)$. The pseudocode of the proposed approach can be found in Algorithm 1.

We would like to emphasize that the proposed DM framework does not require labeled samples from the target domain. The transferred data from the source domain are sufficient for training the classifier. The transfer process is only under the supervision of the co-occurrence data; prior label information from the data in the target domain is not transformed. Therefore, the classification process can be viewed as unsupervised classification. This is a major advantage and innovation of the proposed method, when compared with traditional transfer learning frameworks, for which the labeled data from the target domain are needed.

Algorithm 1: Classification on the Common Semantic Subspace.

Input: X^S, Y^S, V^S, X^T, V^T
Input: $\Theta(W^S, b^S), \Theta(W^T, b^T), n^s, n^t$
Output: Y^T
function SVMTRAINING($X^S, Y^S, V^S, \Theta(W^S, b^S), n^s$)
  for $i = 1, 2, 3, .., n^s$ **do**
    Calculate $A^S(n^s)$ for $X^S(n^s)$ by $\Theta(W^S, b^S)$ as (9)
    $H^S \leftarrow A^S(n^s)V^S(n^s)$
  end for
  $\Psi \leftarrow \{H^S, Y^S\}$
end function
function SVMTESTING($X^T, V^T, \Theta(W^T, b^T), n^t$)
  for $j = 1, 2, 3, .., n^T$ **do**
    Calculate $A^T(n^t)$ for $X^T(n^t)$ by $\Theta(W^T, b^T)$ as (10)
    $H^T \leftarrow A^T(n^t)V^T(n^t)$
    $Y^T \leftarrow \Psi(H^T)$
  end for
end function

IV. Experiments

Experiments are carried out on three HSI datasets: the Pavia dataset, the Washington DC Mall dataset, and the Urban dataset. We focus on the co-occurrence data-supervised heterogeneous transfer learning problem for HSI classification. The co-occurrence data on the two domains are first chosen by SSQ for the transfer process. After that, randomly chosen labeled training samples on the source domain are used for the prediction of unlabeled target-domain data. The transfer is conducted between Pavia University and Washington DC Mall, Urban and Washington DC Mall, and Pavia University and Pavia Center. These three pairs also divide the experiments into three major parts. More detailed settings and results are described in the following sections.

A. Experimental Dataset Descriptions

We choose three sets of data, from the most related to unrelated. The detailed descriptions are as follows. As there is no criterion to determine the source and target domains, we heuristically set the more complicated datasets as the target-domain data and the other one as the source-domain data. With the assistance of source-domain data, the improvement of classification performance on target-domain data is obvious.

The *Pavia Center and Pavia University* are two scenes acquired by the ROSIS sensor during a flight campaign over Pavia, Northern Italy. The number of spectral bands is 102 for Pavia Centre and 103 for Pavia University. Pavia Centre is a 1096×1096 image, and Pavia University is with 610×610 pixels, but some samples in both images contain no information and have to be discarded before the analysis. The geometric resolution is 1.3 m.

The *Washington DC Mall dataset* is a hyperspectral digital imagery collection experiment (HYDICE) image of the Washington DC Mall collected in 1995. The number of spectral bands is 210, and 191 channels are left after discarding the water absorption channels. The original large HSI is divided into Washington DC Mall Area 1 with 307×850 pixels and Washington DC Mall Area 2 with 305×280 pixels. The geometric resolution is 2.8 m.

The *Urban dataset* was also captured by HYDICE in 1995. The detailed area is located at Copperas Cove near Fort Hood, TX, USA. The image includes 307×307 pixels with 210 bands, and 162 bands remain after removing the noisy and water absorption bands. The geometric resolution is 2 m.

Three data pairs are used in the experiments. The first and second are Urban dataset (source-domain data) and Washington DC Mall Area 1 (target-domain data), and Pavia University Data (source-domain data) and Washington DC Mall Area 2 (target-domain data), which have quite low correlation, and the third is Pavia University (source-domain data) and Pavia Center (target-domain data), which is more correlated and easier for the transfer process.

B. Comparative Methods and Evaluation

The proposed DTSE framework mainly exploits the salient samples, further obtains their high-level features, and then constructs the correlation among each domain-specific network and completes the classification process. Therefore, CCA-SVM [36], [44], CCA- and deep-transfer-learning-based SVM (CDTL-SVM), regular representative-informative-sample-transfer-learning-based SVM (RITL-SVM) [9] are adopted as the baseline methods. In addition, we also compare the proposed method with the currently best supervised method IRHTL [39], which yields the highest overall classification accuracy as far as we know, and the most recent dual-space unsupervised structure preserving transfer learning (DSTL) method, which provides the best performance among unsupervised frameworks [45].

The DSTL is an unsupervised HSI classification method, which exploits the structure information of HSI data on dual space. As this method is especially for the homogeneous data, we first process our heterogeneous data to same dimension by dimension reduction. For the classification process, SVM is applied. *We need to point out that the dimension reduction process could significantly affect the performance of* DSTL. This method can get *up to now the best performance for unsupervised transfer-learning-based classification*.

The CCA-SVM uses CCA to derive the correlation between the source and target domains and find the common feature subspace based on linear transformation; then, a standard SVM is employed for classification. As this step is based on shallow transfer learning, the comparison between this method and the proposed deep transfer learning based method can *verify the effectiveness of the DNN*.

The CDTL-SVM and the RITL-SVM are another two baseline methods also proposed in this paper for HSI classification. For the former, without salient sample querying, the method is

TABLE I
DATASETS USED IN EXPERIMENT 1

domain	Source (Urban)	Target (Washington DC Mall)
#total samples	3272	9847
#bands	162	191
#classes	6	6

mainly used for comparison. The co-occurrence data are chosen randomly and a joint representation for data on the source and target domains is found by CDTL. For the latter, we apply the most famous representative-informative-based active learning method to select the salient samples. SVM is explored for final classification. These comparison methods are used to *verify the effectiveness of the proposed active learning process*.

The IRHTL learns two projection matrices to find the similarity and the new data representation in the source and target domains based on the weighted SVM. This method can get *up to now the best performance for supervised transfer-learning-based classification*.

All parameters are fine-tuned with the training data. Detailed setting can be found in each experiment. The proposed DTSE method is trained at four layers including two hidden layers.

For performance evaluation, we report the overall accuracy. As training samples in the SVM classifier are all randomly chosen, each classification process is repeated 100 times with 100 sets of randomly chosen training and testing data to avoid data bias, and the average overall accuracies are used for evaluation. In the following subsections, three experiments on different datasets are reported.

C. Experiment 1: Urban Dataset and Washington DC Mall Area 1

In the first experiment, we conduct our study on the Urban dataset and Washington DC Mall Area 1, as shown in Table I, and the ground truth is shown in Fig. 4. The classification accuracies obtained by different methods are shown in Table II. We construct 15 (C_6^2) binary classification tasks of six categories for comparison. For each task, we select five co-occurrence data samples from each class by the SSQ process for the setup of correlation between the source and target domains. Then, another five samples for each class from the source domain are selected randomly for training on the correlated common subspace. We still repeat the experiment 100 times with 100 sets of randomly chosen training and testing data to avoid data bias. This setting applies for DSTL, CCA-SVM, CDTL-SVM, RITL-SVM, and the proposed DTSE.

As for the deep network setting, we train four-layer neural networks correlated by CCA, in which the number of neurons is $162 \rightarrow 118 \rightarrow 74 \rightarrow 30$ for the source domain and $191 \rightarrow 137 \rightarrow 83 \rightarrow 30$ for target SAEs. This DNN setting applies for CDTL+SVM and the proposed method. The CCA-SVM does not have this deep correlation process, and the DSTL framework that does not apply CCA also uses this deep network to transfer the original heterogeneous data to homogeneous ones.

For the method IRHTL, we choose five training samples from each class on both the source and target domains, the same number as the co-occurrence data. Actually, this setting would favor IRHTL since the co-occurrence data are unlabeled.

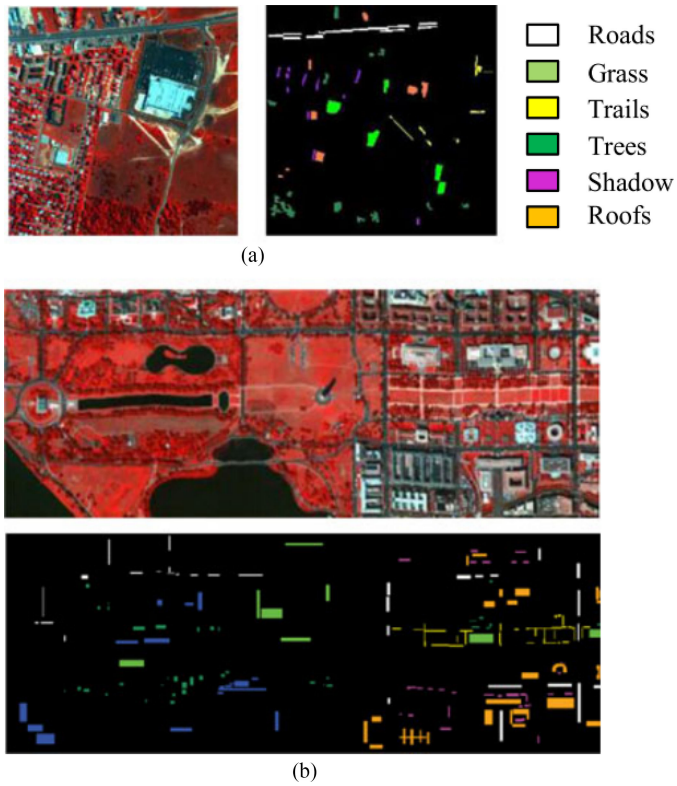

Fig. 4. Dataset used in experiment 1. (a) Source-domain data. (b) Target-domain data.

For the final classification, we apply the same one-against-one SVM classifier for all studied methods, and the Gaussian kernel is chosen for SVM.

The average classification accuracy results of all 15 binary tasks and six categories (Road, Grass, Trails, Trees, Shadow, and Roofs) are shown in Fig. 5 and Table II, in which AA represents average accuracy and OA represents overall accuracy. From the figure, we can note that, overall, the proposed DTSE method yields the best performance, followed by IRHTL, CDTL-SVM, CCA-SVM, and RITL-SVM, while DSTL is the poorest probably because the dimension reduction process may have affected the performance of this unsupervised method. From the table, we can note that Roofs and Trails are the hardest for classification, while the proposed DSTE method still yields 93.48% and 97.80% for these two categories.

From the comparison between the proposed DTSE and IRHTL, we note that the performance of DTSE is more stable, with 90%+ accuracy for each task. IRHTL also provides satisfying results for most of the tasks; however, for tasks 5, 9, and 14, the accuracies decrease to less than 90%, and it gets a poor 18.6% for task 12. IRHTL cannot classify "Roofs" accurately and thus bring the ill results for related tasks. The significance of active learning and deep learning process can be, respectively, supported by comparing the proposed DSTE, CDTL-SVM (RITL-SVM), and CCA-SVM. It can be noted that the active learning process brings an increase of 4% accuracy

TABLE II
CLASSIFICATION ACCURACY RESULTS ON WASHINGTON DC MALL AREA 1

Type/method	Road	Grass	Trails	Trees	Shadow	Roofs
DSTL	0.8335 ± 0.0321	0.8275 ± 0.0459	0.7000 ± 0.0542	0.8645 ± 0.0532	0.7460 ± 0.0389	0.8050 ± 0.0655
CCA-SVM	0.9113 ± 0.0346	0.9395 ± 0.0446	0.9535 ± 0.0465	0.9490 ± 0.0423	0.9440 ± 0.0392	0.9088 ± 0.0456
CDTL-SVM	0.9330 ± 0.0220	0.9603 ± 0.0232	0.9468 ± 0.0248	0.9708 ± 0.0211	0.9495 ± 0.0375	0.8743 ± 0.0233
RITL	0.8844 ± 0.0332	0.9125 ± 0.0329	0.8468 ± 0.0325	0.9251 ± 0.0299	0.9059 ± 0.0344	0.8237 ± 0.0338
IRHTL	0.9608 ± 0.0211	0.9581 ± 0.0336	0.8226 ± 0.0232	0.9574 ± 0.0335	0.9660 ± 0.0254	0.7273 ± 0.0376
DTSE	**0.9693±0.0181**	**0.9773±0.0206**	**0.9780±0.0149**	**0.9815±0.0167**	**0.9848±0.0136**	**0.9348±0.0252**
Type/method	AA	OA				
DSTL	0.7961 ± 0.0483	0.7836 ± 0.0506				
CCA-SVM	0.9344 ± 0.0421	0.9253 ± 0.0424				
CDTL-SVM	0.9391 ± 0.0253	0.9239 ± 0.0241				
RITL	0.8831 ± 0.0328	0.8716 ± 0.0331				
IRHTL	0.8987 ± 0.0291	0.8689 ± 0.0310				
DTSE	**0.9708±0.0182**	**0.9619±0.0204**				

The best performance is emphasized in boldface.

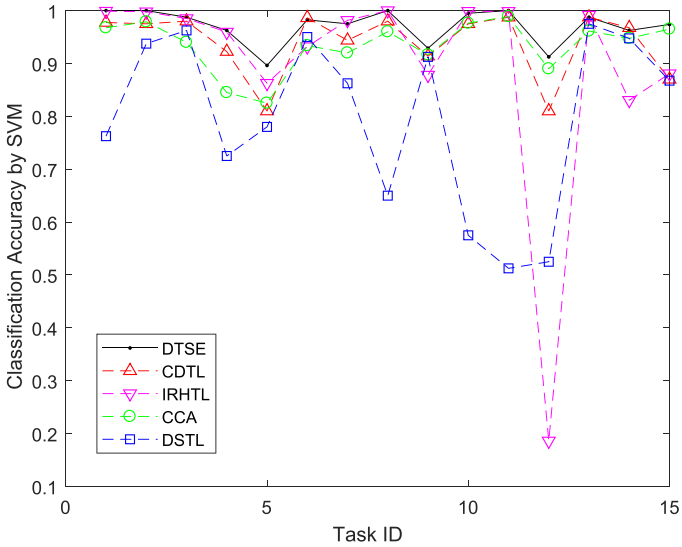

Fig. 5. Urban dataset versus Washington DC Mall Area 1: Classification accuracy results of 15 tasks with five co-occurrence instances and 80 random testing instances. It is worth mentioning that IRHTL needs the label information of instances in the target domain, while other methods do not.

in average by comparing DSTE and CDTL-SVM. By comparing the DSTE and RITL-SVM, it can be found the increase of around 7% accuracy verifies the effectiveness of the newly proposed active learning framework. With deep learning, another 3–4% accuracy increase can be found by comparing CDTL-SVM and CCA-SVM, between which the only difference is the DM process. The effectiveness of co-occurrence data can be illustrated by comparing CCA-SVM with the unsupervised DSTL. With the training process based on co-occurrence data, the accuracy is more steady for each task, and an overall 5% accuracy increase is noted.

It should also be emphasized that the larger the semantic difference between the source and target domains, the more performance improvements can be found in the proposed DSTE, which is probably the most important advantage of the proposed DSTE. This observation can be drawn by comparing DTSE and IRHTL. The difference of "Roofs" in two domains is the largest when observing the spectral information of the two HSI images. IRHTL often fails on the classification of this category, while DTSE still works well likely due to its employment of the DM process. By mapping between each corresponding layer of the autoencoder on the source and target domains, the semantic gap between the two domains is minimized. The final spectral distributions of "Roofs" on these two HSIs are almost the same and thus benefit the following classification process.

D. Experiment 2: Pavia University Data and Washington DC Mall Area 2

In the second experiment, we conduct our study on Pavia University Data and Washington DC Mall Area 2, with detailed information being shown in Table III, and the ground truth is as shown in Fig. 6. The classification accuracies obtained by different methods are shown in Table IV. We construct six (C_4^2) binary classification tasks of four categories. The co-occurrence data samples from each class are still 5, and 80 testing samples from each class are randomly chosen. Another five samples from the source domain are randomly chosen for training on the correlated common subspace. The experiments are repeated for 100 times with 100 sets of randomly chosen training and testing data to avoid data bias. This data setting applies for all five methods under comparison.

For the deep network setting, the number of neurons in four-layer domain networks correlated by CCA are $103 \rightarrow 79 \rightarrow 55 \rightarrow 30$ on the source domain and $191 \rightarrow 137 \rightarrow 81 \rightarrow 30$ for target SAEs. This DNN setting applies for CDTL+SVM and the proposed method, as in Experiment 1. For the comparison method IRHTL, we choose five training samples from each class on both the source and target domains, the same number as the co-occurrence data. The one-against-one SVM classifier is employed for the final classification.

The average classification accuracy results of all six binary tasks and four categories are given in Fig. 7 and Table III. From the figure, we can note that, overall, the proposed DTSE and IRHTL provide the best performances, while DTSE yields more consistent performances. The accuracies of DTSE are close to or above 90%, while for IRHTL, the classification of "Roads" and "Roofs" is 86.42%, and the accuracy is 70.40% for "Road" and "Roofs," which is the poorest performance among five methods. For the remaining four tasks, DTSE and IRHTL are with almost the same accuracies (the difference is less than 1%). These

TABLE III
CLASSIFICATION ACCURACY RESULTS ON WASHINGTON DC MALL AREA 2

Type/method	Road	Bare soil	Vegetation	Roofs	AA	OA
DSTL	0.8458 ± 0.0323	0.8625 ± 0.0351	0.8292 ± 0.0554	0.7542 ± 0.0604	0.8229 ± 0.0458	0.8357 ± 0.0472
CCA-SVM	0.8825 ± 0.0445	0.8921 ± 0.0311	0.9246 ± 0.0221	0.8342 ± 0.0463	0.8831 ± 0.0360	0.8657 ± 0.0420
CDTL-SVM	0.9184 ± 0.0531	0.8905 ± 0.0276	0.9259 ± 0.0338	0.9054 ± 0.0145	0.8851 ± 0.0323	0.9106 ± 0.0317
RITL-SVM	0.8842 ± 0.0392	0.8924 ± 0.0325	0.9358 ± 0.0311	0.8216 ± 0.0377	0.8835 ± 0.0351	0.8618 ± 0.0372
IRHTL	0.9511 ± 0.0421	0.8990 ± 0.0412	**0.9964±0.0021**	0.8558 ± 0.0332	0.9256 ± 0.0297	0.9078 ± 0.0344
DTSE	**0.9667±0.0132**	**0.9542±0.0104**	0.9917 ± 0.0033	**0.9375±0.0169**	**0.9625±0.0110**	**0.9548±0.0137**

The best performance is emphasized by boldface.

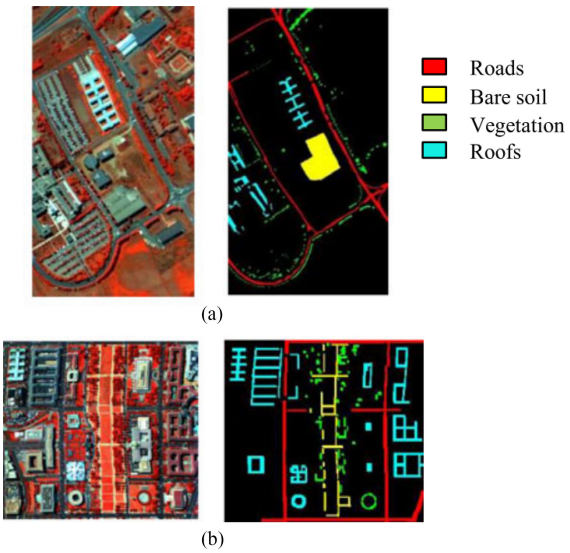

Fig. 6. Dataset used in experiment 2. (a) Source-domain data. (b) Target-domain data.

TABLE IV
DATASETS USED IN EXPERIMENT 2

domain	Source(Pavia University)	Target(Washington DC Mall)
#total samples	18168	11868
#bands	103	191
#classes	4	4

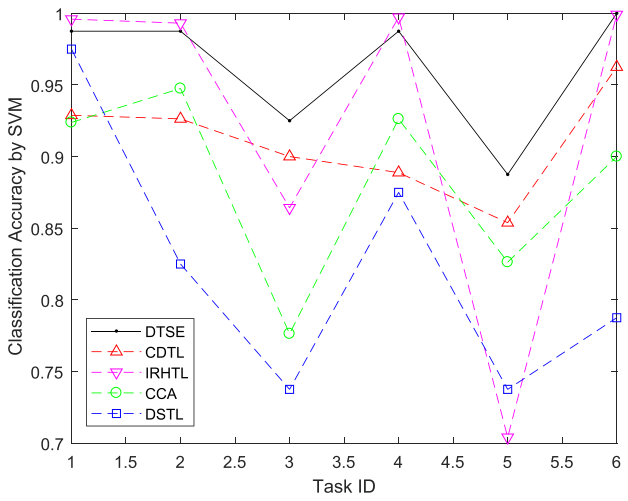

Fig. 7. Pavia University Data and Washington DC Mall Area 2: Classification accuracy results of six tasks with five co-occurrence instances and 80 random testing instances. It is worth mentioning that IRHTL needs the label information of instances in the target domain, while other methods do not.

TABLE V
DATASETS USED IN EXPERIMENT 3

domain	Source(Pavia University)	Target(Pavia Center)
#total samples	18168	62046
#bands	103	102
#classes	4	4

observations suggest that, via the DM process, the semantic gap is effectively handled in the proposed DTSE. Thus, DTSE is especially suitable for the transfer process between domains with large semantic difference.

It can be noted that active learning brings an increase of 4–10% accuracy in average when comparing DSTE and CDTL-SVM. As the CDTL-SVM has obvious better classification performances on tasks 3, 5, and 6 and almost the same performance for the rest tasks, this improvement indicates the effectiveness of the DM process. The effectiveness of the proposed active learning method can also be verified by comparing it with RITL-SVM, as an increase of around 8% accuracy be found. There is a huge performance difference (more than 10% accuracy gap for most tasks) between CCA-SVM and DSTL, meaning that the training process based on co-occurrence data cannot be neglected.

E. Experiment 3: Pavia University and Pavia Center Data

In the third experiment, we conduct our study on Pavia University Data and Pavia Center Data, with detailed information being shown in Table V, and the ground truth is as shown in Fig. 8. The classification accuracies obtained by different methods are shown in Table VI and Fig. 9. We construct six (C_4^2) binary classification tasks for four categories. The detailed data setting is the same as in Experiment 2.

For the deep network setting, the number of neurons in four-layer domain networks correlated by CCA is $103 \rightarrow 79 \rightarrow 55 \rightarrow 30$ on the source domain and $102 \rightarrow 78 \rightarrow 54 \rightarrow 30$ for target SAEs. Other settings are the same as in Experiment 2.

In this experiment, as the two domains are more correlated than the former two experiments, IRHTL yields relatively more stable results. It only fails on the classification of "Bitumen" and "Bare Soil," with about 70% accuracy. For the proposed DTSE, the accuracy results are almost all 95%+.

The proposed DTSE still outperforms other four methods. This observation is consistent with the assertion that the multilayer semantic mapping model can discover the deep shared subspace across domains and transfer the sufficient labeled source-domain data knowledge to the target domain for label

TABLE VI
CLASSIFICATION ACCURACY RESULTS ON PAVIA CENTER

Type/method	Trees	Self-Blocking Bricks	Bitumen	Bare Soil	AA	OA
DSTL	0.5550 ± 0.0931	0.8100 ± 0.0251	0.7058 ± 0.0681	0.6608 ± 0.0773	0.6829 ± 0.0659	0.6826 ± 0.666
CCA-SVM	0.9179 ± 0.0203	0.8717 ± 0.0224	0.8950 ± 0.0321	0.9171 ± 0.0213	0.9004 ± 0.0240	0.9023 ± 0.0225
CDTL-SVM	0.9529 ± 0.0123	0.9446 ± 0.0331	0.8946 ± 0.0436	0.9304 ± 0.0216	0.9306 ± 0.0277	0.9360 ± 0.0249
RITL-SVM	0.9323 ± 0.0165	0.9492 ± 0.0243	0.8923 ± 0.0289	0.9232 ± 0.0339	0.9243 ± 0.0259	0.9295 ± 0.0266
IRHTL	0.9479 ± 0.0221	0.9638 ± 0.0155	0.8619 ± 0.0342	0.8855 ± 0.0213	0.9148 ± 0.0233	0.9197 ± 0.0212
DTSE	**0.9688±0.0210**	**0.9750±0.0163**	**0.9458±0.0115**	**0.9729±0.0031**	**0.9656±0.0130**	**0.9698±0.0118**

The best performance is emphasized by boldface.

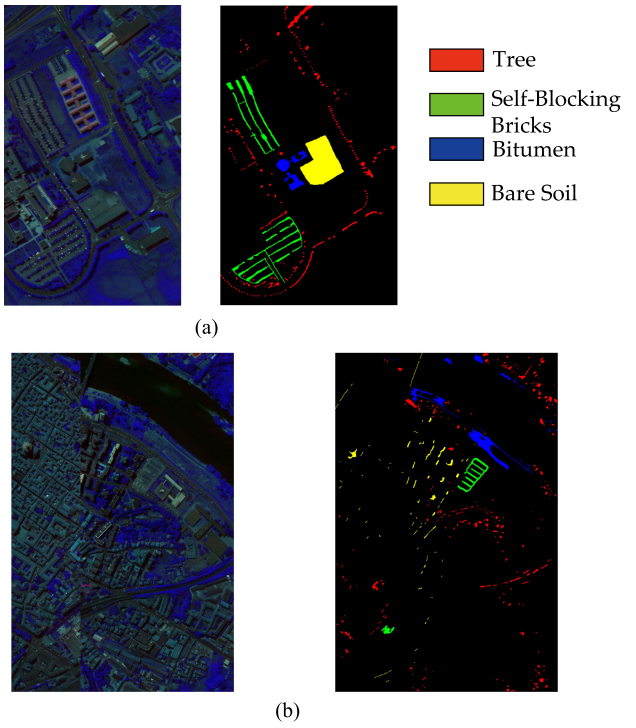

Fig. 8. Dataset used in experiment 3. (a) Source-domain data. (b) Target-domain data.

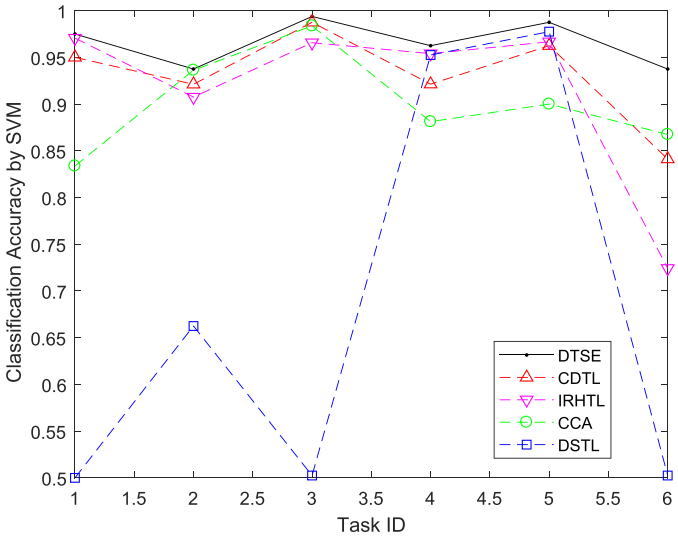

Fig. 9. Pavia University Data and Pavia Center: Classification accuracy results of six tasks with five co-occurrence instances and 80 random testing instances. It is worth mentioning that IRHTL needs the label information of instances in the target domain, while other methods do not.

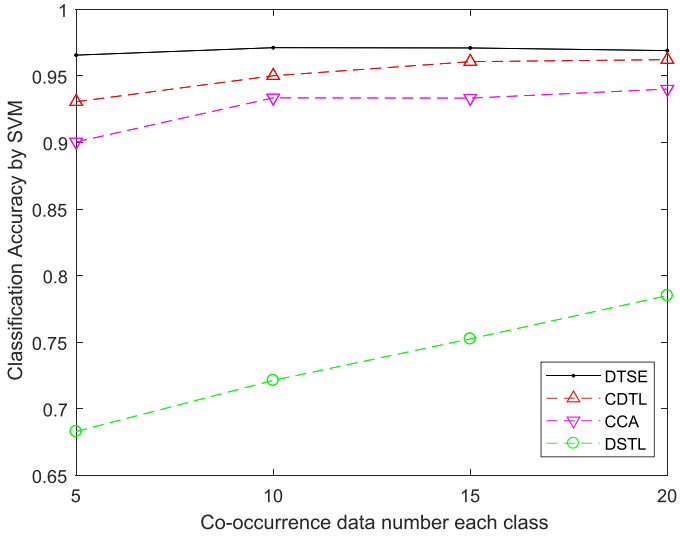

Fig. 10. Effects of the co-occurrence data size on four different methods when tested on the Pavia Center dataset.

prediction by exploring co-occurrence information. The deep transfer learning methods (DTSE, CDTL-SVM, and RITL-SVM) perform better than the shallow transfer learning methods (CCA-SVM, DSTL, and IRHTL). Based on this observation, the fact that discriminative semantic information can be embedded in a multilayer of the feature hierarchy is further proved. DNNs can capture this information through multiple nonlinear transformations. After DM, the semantic divergence and feature bias between the source and target domains are much lower. The active learning process that selects the co-occurrence data can improve the accuracy by 5% on average when comparing DTSE and CDTL-SVM. Compared with RITL-SVM, the effectiveness of the proposed active learning framework is shown.

F. Effective of Co-Occurrence Data

In this section, we discuss the effect of the number of co-occurrence data on the DM process. Here, we take the experiment on Pavia Center as an example. As the co-occurrence data are not used in the IRHTL method, we study the rest four methods in this subsection.

The accuracy of each method is the average accuracy of all six tasks. From Fig. 10, three observations can be noted.

First, the number of co-occurrence data matters but makes no significant difference in general. With the increase of the number of co-occurrence data, the classification accuracy increases gradually for all four methods. However, this improvement is

TABLE VII
EFFECTS OF THE NUMBER OF NEURONS AT THE LAST LAYER

Type/Neurons	Road	Grass	Trails	Trees	Shadow	Roofs	OA
10	0.8132 ± 0.0324	0.8071 ± 0.0319	0.8080 ± 0.0316	0.8190 ± 0.0253	0.8565 ± 0.0309	0.8350 ± 0.0375	0.8231 ± 0.0333
20	0.9011 ± 0.0102	0.9231 ± 0.0216	0.9245 ± 0.0216	0.9191 ± 0.0210	0.9003 ± 0.0161	0.8633 ± 0.0171	0.8957 ± 0.0172
30	**0.9693±0.0181**	**0.9773±0.0206**	**0.9780±0.0149**	**0.9815±0.0167**	**0.9848±0.0136**	**0.9348±0.0252**	**0.9619±0.0204**
40	0.8743 ± 0.0332	0.9116 ± 0.0219	0.9019 ± 0.0321	0.9115 ± 0.0249	0.9274 ± 0.0208	0.8144 ± 0.0243	0.8708 ± 0.0260
50	0.8350 ± 0.0266	0.8548 ± 0.0246	0.8875 ± 0.0310	0.8269 ± 0.0233	0.8962 ± 0.0222	0.8053 ± 0.0388	0.8368 ± 0.0301

TABLE VIII
EFFECTS OF THE NUMBER OF LAYERS

Type/Layers	Road	Grass	Trails	Trees	Shadow	Roofs	OA
2	0.7603 ± 0.0223	0.7025 ± 0.0239	0.6450 ± 0.0242	0.7755 ± 0.0433	0.6221 ± 0.0209	0.7128 ± 0.0235	0.7137 ± 0.0233
3	0.8954 ± 0.0433	0.9233 ± 0.0225	0.9319 ± 0.0311	0.9290 ± 0.0384	0.9620 ± 0.0212	0.9188 ± 0.0513	0.9202 ± 0.0387
4	**0.9693±0.0181**	**0.9773±0.0206**	**0.9780±0.0149**	**0.9815±0.0167**	**0.9848±0.0136**	**0.9348±0.0252**	**0.9619±0.0204**
5	0.8324 ± 0.0218	0.8845 ± 0.0356	0.8439 ± 0.0432	0.8964 ± 0.0209	0.8329 ± 0.0234	0.7967 ± 0.0262	0.8373 ± 0.0278

not significant, and five co-occurrence data samples may only bring less than 1% accuracy increase.

The second observation is that the effect of co-occurrence data on the comparison methods is larger than on the proposed method, and the effect on DSTL is especially distinct. The reason is likely as follows: With five co-occurrence data samples, the proposed DSTE method can get a robust classification model, while with more co-occurrence data, overfitting may occur. The accuracy even slightly decreases for DSTE with 20 co-occurrence data samples when compared with 15. For other methods, the classification model may still have space to improve with the increase of the co-occurrence data size.

The third observation is that, with enough co-occurrence data, CDTL-SVM can yield very close performance to that of the proposed DSTE. The reason is likely as follows: DSTE selects the most efficient co-occurrence data and five co-occurrence data samples are enough; for CDTL-SVM, it can achieve the same performance as DSTE as long as the most salient samples are included in the selected co-occurrence data although the required number of data samples is much larger and much redundancy may exist. This observation to some extent reflects the significance of the SSQ process, which makes the proposed method more efficient.

G. Parameter Sensitivity

In this section, we study the effect of different parameters in our networks. Both the number of layers and the number of neurons at each layer would affect the final classification result. Here, we take the overall classification result in Experiment 1 as an example. We first set different numbers of neurons in the four-layer network. We evaluate different number of neurons in a single hidden layer when fixing other layers. For example, we set the numbers of neurons in the first three layers as $162 \longrightarrow 118 \longrightarrow 74$ in the source domain and $191 \longrightarrow 137 \longrightarrow 83$ in the target domain. For the last layer, the neuron number changes from 10 to 50 in both domains, and the final accuracy is shown in Table VII.

From this table, we can note that, when the number of neuron is 30, the performance is the best. Therefore, in our former experiments, we use 30 neurons.

Second, we also test the effect of the number of layers. We choose the best number of neurons in each layer, and the results are shown in Table VIII.

We note that with the increase of number of layers, the performance first rises gradually, but when it reaches five layers, the accuracy falls suddenly. The reason might be with more layers, overfitting problem might be heavy. Based on this observation, we set the number of layers to be 4 in our former experiments.

V. CONCLUSIONS

In this paper, we propose a novel method, referred to as DTSE, for the classification of HSIs. In the proposed model, we first query the salient examples to obtain the co-occurrence data and then apply them to construct the DM network on the source and target domains. The CCA is applied at each layer to correlate the two domains' data. Then, at the top layer, we exploit the correlation matching between two domains to fine-tune the whole network in backpropagation. Therefore, the final correlated common subspace is identified, and the data on the source domain are projected to this subspace for training the SVM classifier for classification on the target domain. The proposed framework is tested on three HSI datasets and compared to other four state-of-the-art methods.

In the future work, we plan to apply the proposed method to more HSI datasets, as well as other forms of non-HSI datasets. Also, since the learned deep network is affected heavily by the parameter setting, more efforts would be made to find the best settings particularly suitable for each individual HSI dataset.

Moreover, we have to point out that the current framework is only suitable for binary classification. Extending it to multiclass classification should be another future work.

APPENDIX

Here, we want to optimize the objective function in (11), which is not joint convex considering $\Theta^T = \{W^T, b^T\}$, $\Theta^S = \{W^S, b^S\}$, V^S, and V^T. To solve this problem, we adopt the Lagrangian multiplier to update Θ^S and Θ^T and the stochastic gradient descent method to update V^S and V^T. The objective function is derived into two subproblems as follows.

A. Updating V^S and V^T With Fixed Θ^S and Θ^T

In (11), the optimization of V^S and V^T is just related to the third term, and the optimization of each layer $V^S(l)$ and $V^T(l)$ can be formulated as

$$\min_{V^{S(l)},V^{T(l)}} -\frac{V^{S(l)^T}\sum_{ST}V^{T(l)}}{\sqrt{V^{S(l)^T}\sum_{SS}V^{S(l)}}\sqrt{V^{T(l)^T}\sum_{TT}V^{T(l)}}}. \tag{15}$$

As $V^{S(l)^T}\sum_{SS}V^{S(l)}=1$ and $V^{T(l)^T}\sum_{TT}V^{T(l)}=1$, we have the Lagrangian multiplier

$$\begin{aligned}L(w_l,V^{S(l)},V^{T(l)}) &= -V^{S(l)^T}\sum_{ST}V^{T(l)} \\ &+ \frac{w_l^S}{2}\left(V^{S(l)^T}\sum_{SS}V^{S(l)}-1\right) \\ &+ \frac{w_l^T}{2}\left(V^{T(l)^T}\sum_{TT}V^{T(l)}-1\right).\end{aligned} \tag{16}$$

Then, we take the partial derivatives for (16) and obtain

$$\begin{aligned}\frac{\partial L}{\partial V^{S(l)}} &= \sum_{ST}V^{T(l)} - w_l^S\sum_{SS}V^{S(l)} = 0 \\ \frac{\partial L}{\partial V^{T(l)}} &= \sum_{ST}V^{S(l)} - w_l^T\sum_{SS}V^{T(l)} = 0.\end{aligned} \tag{17}$$

After reduction, we further have

$$V^{T(l)} = \frac{\sum_{TT}^{-1}\sum_{ST}V^{S(l)}}{w_l} \tag{18}$$

$$\sum_{ST}\sum_{TT}^{-1}\sum_{ST}^{T}V^{S(l)} = w_l^2\sum_{SS}V^{S(l)} \tag{19}$$

and $w_l = w_l^S = w_l^T$. So, $V^S(l)$ and w_l in (18) can be solved by the generalized eigenvalue decomposition, and the corresponding $V^T(l)$ can be obtained by (19).

B. Updating Θ^S and Θ^T With Fixed V^S and V^T

As Θ^S and Θ^T are mutual independent and of the same form, we just demonstrate the solution of Θ^S on the source domain (the solution of Θ^T can be derived similarly) as

$$\min_{\theta^S}\phi(\theta^S) = J_S(W^s,b^S) - \Gamma(V^S,V^T). \tag{20}$$

Here, we apply the gradient descent method to adjust the parameter as

$$\begin{aligned}W^{S(l)} &= W^{S(l)} - \mu^S\frac{\partial\phi}{\partial W^{S(l)}} \\ &= \frac{\partial J_S(W^S,b^S)}{\partial W^{S(l)}} - \frac{\partial\Gamma(V^S,V^T)}{\partial W^{S(l)}} \\ &= \frac{(\alpha^{S(l+1)} - \beta^{S(l+1)} + \omega_l\gamma^{S(l+1)})\times A^{S(l)}}{n_c + \lambda^S W^{S(l)}}\end{aligned} \tag{21}$$

$$\begin{aligned}b^{S(l)} &= b^{S(l)} - \mu^S\frac{\partial\phi}{\partial b^{S(l)}} \\ &= \frac{\partial J_S(W^S,b^S)}{\partial b^{S(l)}} - \frac{\partial\Gamma(V^S,V^T)}{\partial b^{S(l)}} \\ &= \frac{(\alpha^{S(l+1)} - \beta^{S(l+1)} + \omega_l\gamma^{S(l+1)})}{n_c}\end{aligned} \tag{22}$$

in which

$$\alpha^{S(l)} = \begin{cases}-(C^S - A^{S(l)})\bullet A^{S(l)}\bullet(1-A^{S(l)}), & l = n^S \\ W^{S(l)^T}\alpha^{S(l+1)}\bullet A^{S(l)}\bullet(1-A^{S(l)}), & l = 2,\ldots,n^S-1\end{cases} \tag{23}$$

$$\beta^{S(l)} = \begin{cases}0, & l = n^S \\ A^{T(l)}V^{T(l)}V^{S(l)^T}\bullet A^{S(l)}\bullet(1-A^{S(l)}), & l = 2,\ldots,n^S-1\end{cases} \tag{24}$$

$$\gamma^{S(l)} = \begin{cases}0, l = n^S \\ A^{S(l)}V^{S(l)}V^{S(l)^T}\bullet A^{S(l)}\bullet(1-A^{S(l)}), & l = 2,\ldots,n^S-1\end{cases}. \tag{25}$$

Here, the operator $\bullet$ stands for the dot product. The same optimization process works for Θ^T on the target domain. After these two optimizations for each layer, the CCA on the top hidden layer is employed to fine-tune all parameters of the whole network by the backpropagation process. As we just exploit the correlation of two domain networks, the objective function is defined as

$$\min_{\theta^S,\theta^T,V^S,V^T} J = -\Gamma(V^S,V^T). \tag{26}$$

The procedures of updating $\{V^S,V^T\}$ are the same as in (15) but with different parameters. Yet, in (20) for updating Θ^S and Θ^T, the settings should be $\alpha^S(l)=0$ and

$$\beta^{S(l)} = \begin{cases}A^{T(l)}V^{T(l)}V^{S(l)^T}\bullet A^{S(l)}\bullet(1-A^{S(l)}), & l = n^S \\ W^{S(l)}\beta^{S(l+1)}\bullet A^{S(l)}\bullet(1-A^{S(l)}), & l = 2,\ldots,n^S-1\end{cases} \tag{27}$$

$$\gamma^{S(l)} = \begin{cases}A^{S(l)}V^{S(l)}V^{S(l)^T}\bullet A^{S(l)}\bullet(1-A^{S(l)}), & l = n^S \\ W^{S(l)}\gamma^{S(l+1)}\bullet A^{S(l)}\bullet(1-A^{S(l)}), & l = 2,\ldots,n^S-1\end{cases}. \tag{28}$$

The optimization process is summarized in Algorithm 2.

In Algorithm 2, when it is converged, the domain-specific networks and the correlation coefficients between the domains are achieved for the DM process. It is noteworthy that, to guarantee

Algorithm 2: DM Model Training.

Input: $D^C = \{C_i^S, C_i^T\}_i^{n_C}$,
Input: $\lambda^S = 1, \lambda^T = 1, \mu^S = 0.5, \mu^T = 0.5$
Output: $\Theta(W^S, b^S), \Theta(W^T, b^T), V^S, V^T$

function INITIALIZATION
  Initialize $\Theta(W^S, b^S), \Theta(W^T, b^T) \leftarrow RandomNum$
  for $l = 1, 2, \ldots, n^S$ **do**
    $V^S \leftarrow \arg\min L(\omega_l, V^{S(l)})$
  end for
  for $l = 1, 2, \ldots, n^T$ **do**
    $V^T \leftarrow \arg\min L(\omega_l, V^{T(l)})$
  end for
  $\theta^S = \arg\min \phi(\theta^S), \theta^T = \arg\min \phi(\theta^T)$
end function
function LASTLAYERFINETUNING
  repeat
    Set $\alpha^S, \alpha^T = 0$, update β, γ by (27) and (28)
    for $l = 1, 2, \ldots, n^S$ **do**
      $V^S \leftarrow \arg\min L(\omega_l, V^{S(l)})$
    end for
    for $l = 1, 2, \ldots, n^T$ **do**
      $V^T \leftarrow \arg\min L(\omega_l, V^{T(l)})$
    end for
    $\theta^S, \theta^T = \arg\min -\Gamma(V^S, V^T)$
  until Convergence
end function

the performance of this process, the parameters are heuristically selected to yield the best results.

REFERENCES

[1] G. Foody and A. Mathur, "Toward intelligent training of supervised image classifications: Directing training data acquisition for SVM classification," *Remote Sens. Environ.*, vol. 93, no. 1, pp. 107–117, 2004.

[2] A. Samat, J. Li, S. Liu, P. Du, Z. Miao, and J. Luo, "Improved hyperspectral image classification by active learning using pre-designed mixed pixels," *Pattern Recognit.*, vol. 51, pp. 43–58, 2016.

[3] M. Shao, C. Castillo, G. Zhenghong, and Y. Fu, "Low-rank transfer subspace learning," in *Proc. IEEE Int. Conf. Data Mining*, 2012, pp. 1104–1109.

[4] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014.

[5] X. Zhou and S. Prasad, "Active and semisupervised learning with morphological component analysis for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 8, pp. 1348–1352, Aug. 2017.

[6] P. Liu, H. Zhang, and K. B. Eom, "Active deep learning for classification of hyperspectral images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 2, pp. 712–724, Feb. 2017.

[7] Z. Wang, B. Du, L. Zhang, L. Zhang, and X. Jia, "A novel semisupervised active-learning algorithm for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 6, pp. 3071–3083, Jun. 2017.

[8] J. Lin, R. Ward, and Z. Wang, "Deep transfer learning for hyperspectral image classification," in *Proc. IEEE Int. Workshop Multimedia Signal Process.*, 2018, pp. 1–4.

[9] S. Huang, R. Jin, and Z. Zhou, "Active learning by querying informative and representative examples," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 10, pp. 1936–1949, Oct. 2014.

[10] M. Balcan, A. Broder, and T. Zhang, "Margin based active learning," in *Proc. 20th Annu. Conf. Learn. Theory*, 2007, pp. 35–50.

[11] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," in *Proc. 17th Int. Conf. Mach. Learn.*, 2000, pp. 999–1006.

[12] D. Lewis and J. Catlett, "Heterogeneous uncertainty sampling for supervised learning," in *Proc. 11th Int. Conf. Mach. Learn.*, 1994, pp. 148–156.

[13] I. Dagan and S. Engelson, "Committee-based sampling for training probabilistic classifiers," in *Proc. 12th Int. Conf. Mach. Learn.*, 1995, pp. 150–157.

[14] Y. Freund, H. Seung, E. Shamir, and N. Tishby, "Selective sampling using the query by committee algorithm," *Mach. Learn.*, vol. 28, nos. 2/3, pp. 133–168, 1997.

[15] Z. Xu, K. Yu, V. Tresp, X. Xu, and J. Wang, "Query by committee," in *Proc. Int. Workshop Comput. Learn. Theory*, 1992, pp. 287–294.

[16] N. Roy and A. Mccallum, "Toward optimal active learning through sampling estimation of error reduction," in *Proc. 18th Int. Conf. Mach. Learn.*, 2001, pp. 441–448.

[17] Y. Guo and D. Schuurman, "Toward optimal active learning through sampling estimation of error reduction," in *Proc. Neural Inf. Process. Syst. Conf.*, 2007, pp. 593–600.

[18] Q. Wang, F. Zhang, and X. Li, "Optimal clustering framework for hyperspectral band selection," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 10, pp. 5910–5922, Oct. 2018.

[19] S. Dasgupta and D. HsuLewis, "Hierarchical sampling for active learning," in *Proc. IEEE Int. Conf. Mach. Learn.*, 2008, pp. 208–215.

[20] R. Chattopadhyay, Z. Wang, W. Fan, I. Davidson, S. Panchanathan, and J. Ye, "Batch mode active sampling based on marginal probability distribution matching," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2012, pp. 741–749.

[21] K. Yu, B. Ji, and V. Tresp, "Active learning via transductive experimental design," in *Proc. IEEE Int. Conf. Mach. Learn.*, 2006, pp. 1081–1088.

[22] P. Donmez, J. Carbonell, and P. Bennett, "Active learning via transductive experimental design," in *Proc. IEEE Int. Conf. Mach. Learn.*, 2007, pp. 116–127.

[23] G. Matasci, D. Tuia, and M. Kanevski, "SVM-based boosting of active learning strategies for efficient domain adaptation," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 5, no. 5, pp. 1335–1343, Oct. 2012.

[24] C. Persello, "Interactive domain adaptation for the classification of remote sensing images using active learning," *IEEE Geosci. Remote Sens. Lett.*, vol. 10, no. 4, pp. 736–740, Jul. 2013.

[25] L. Zhao, S. Pan, W. Xiang, E. Zhong, Z. Lu, and Q. Yang, "Active transfer learning for cross-system recommendation," in *Proc. Assoc. Adv. Artif. Intell.*, 2013, pp. 1205–1211.

[26] E. Gavves, T. Mensink, T. Tommasi, C. G. Snoek, and T. Tuytelaars, "Active transfer learning with zero-shot priors: Reusing past datasets for future tasks," 2015, arXiv:1510.01544.

[27] C. Persello and L. Bruzzone, "Active learning for domain adaptation in the supervised classification of remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 11, pp. 4468–483, Nov. 2012.

[28] C. Persello and L. Bruzzone, "A novel active learning strategy for domain adaptation in the classification of remote sensing images," in *Proc. IEEE Int. Conf. Geosci. Remote Sens. Symp.*, 2011, pp. 3720–3723.

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

[30] J. Li, H. Zhang, Y. Huang, and L. Zhang, "Visual domain adaptation: A survey of recent advances," *IEEE Signal Process. Mag.*, vol. 33, no. 3, pp. 53–69, May 2015.

[31] Q. Wang, Z. Yuan, and X. Li, "GETNET: A general end-to-end two-dimensional CNN framework for hyperspectral image change detection," *IEEE Trans. Geosci. Remote Sens.*, 2018, to be published, doi: 10.1109/TGRS.2018.2849692.

[32] X. Glorot, A. Bordes, and Y. Bengio, "Domain adaptation for large-scale sentiment classification: A deep learning approach," in *Proc. Int. Conf. Mach. Learn.*, 2011, pp. 513–521.

[33] M. Chen, Z. Xu, K. Weinberger, and F. Sha, "Marginalized denoising autoencoders for domain adaptation," in *Proc. Int. Conf. Mach. Learn.*, 2011, pp. 1–8.

[34] Z. Ding, N. Nasrabadi, and Y. Fu, "Task-driven deep transfer learning for image classification," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2016, pp. 2414–2418.

[35] F. Zhuang, X. Cheng, P. Luo, S. Pan, and Q. He, "Supervised representation learning: Transfer learning with deep autoencoders," in *Proc. Int. Joint Conf. Artif. Intell.*, 2015, pp. 4119–4125.

[36] D. 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.
[37] J. Lin, Q. Wang, R. Ward, and Z. Wang, "DT-LET: Deep transfer learning by exploring where to transfer," 2018, arXiv:1809.08541.
[38] G. Andrew, R. Arora, J. Bilmes, and K. Livescu, "Deep canonical correlation analysis," *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 1247–1255.
[39] X. Li, L. Zhang, B. Du, L. Zhang, and Q. Shi, "Iterative reweighting heterogeneous transfer learning framework for supervised remote sensing image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 5, pp. 2022–2035, May 2017.
[40] J. Zhou, J. Pan, I. Tsang, and Y. Yan, "Active transfer learning for cross-system recommendation," in *Proc. Assoc. Adv. Artif. Intell.*, 2014, pp. 2213–2220.
[41] J. Tang, X. Shu, Z. Li, G. Qi, and J. Wang, "Generalized deep transfer networks for knowledge propagation in heterogeneous domains," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 12, no. 4, 2016, Art. no. 68.
[42] X. Li, B. Liu, and S. Ng, "Negative training data can be harmful to text classification," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2010, pp. 218–228.
[43] C. Seah and Ong, and I. Tsang, "Combating negative transfer from predictive distribution differences," *IEEE Trans. Cybern.*, vol. 43, no. 4, pp. 1153–1165, Aug. 2013.
[44] Y. Yeh, C. Huang, and Y. Wang, "Heterogeneous domain adaptation and classification by exploiting the correlation subspace," *IEEE Trans. Image Process.*, vol. 23, no. 5, pp. 2009–2018, May 2014.
[45] J. Lin, C. He, Z. Wang, and S. Li, "Structure preserving transfer learning for unsupervised hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 10, pp. 1656–1660, Oct. 2017.

Jianzhe Lin (S'14) received the B.E. degree in optical engineering and the B.A. degree in English from the Huazhong University of Science and Technology, Wuhan, China, in 2013, and the master's degree from the Chinese Academy of Sciences, Beijing, China, in 2016. He is working toward the Ph.D. degree with the Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC, Canada.

His current research interests include computer vision and machine learning.

Liang Zhao received the B.S., M.S., and Ph.D. degrees in software engineering from the Dalian University of Technology, Dalian, China, in 2011, 2014 and 2018, respectively.

He is currently an Assistant Professor with the School of Software Technology, Dalian University of Technology.

Shuying Li received the B.E. degree from the University of Science and Technology of China, Hefei, China, in 2005, and the Ph.D. degree from the Chinese Academy of Sciences, Beijing, China, in 2010.

She is currently a Professor with the School of Automation, Xi'an University of Posts and Telecommunications, Xi'an, China. Her research interests include remote sensing, computer vision, and pattern recognition.

Rabab Ward (F'99) is currently a Professor Emeritus with the Electrical and Computer Engineering Department, University of British Columbia (UBC), Canada. Her research interests include signal, image, and video processing. She has made contributions in the areas of signal detection, image encoding, image recognition, restoration and enhancement, and their applications to multimedia and medical imaging, face recognition, infant cry signals, and brain computer interfaces. She has authored or coauthored about 500 refereed journal and conference papers and holds six patents related to cable television, picture monitoring, measurement, and noise reduction.

She is a Fellow of the Royal Society of Canada, the IEEE, the Canadian Academy of Engineers and the Engineering Institute of Canada. She has received many top awards such as the Society Award of the IEEE Signal Processing Society, the Career Achievement Award of CUFA BC, The Paradigm Shifter Award from The Society for Canadian Women in Science and Technology and British Columbia's APEGBC top engineering award The RA McLachlan Memorial Award and UBC Killam Research Prize and Killam Senior Mentoring Award. She is currently the President of the IEEE Signal Processing Society. She was the General Chair of IEEE ICIP 2000 and Co-Chair of IEEE ICASSP 2013.

Z. Jane Wang (M'02–F'17) received the B.Sc. degree from Tsinghua University, Beijing, China, in 1996, with the highest honor, and the M.Sc. and Ph.D. degrees from the University of Connecticut, Storrs, CT, USA, in 2000 and 2002 (under the supervision of Dr. Peter Willett), respectively.

Since August 2004, she has been with the Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC, Canada, where she is currently a Professor.