

# Hyperspectral Image Classification Based on Domain Adversarial Broad Adaptation Network

Haoyu Wang, Yuhu Cheng<sup>ID</sup>, Member, IEEE, C. L. Philip Chen<sup>ID</sup>, Fellow, IEEE,  
and Xuesong Wang<sup>ID</sup>, Member, IEEE

**Abstract**—For hyperspectral image (HSI) classification tasks, obtaining sufficient labeled samples is usually difficult, time-consuming, and expensive. To address the aforementioned issue, by transferring the labeled sample information of a relevant source domain to the unlabeled target domain, an HSI classification method based on the domain adversarial broad adaptation network (DABAN) is proposed. First, the bottleneck adaptation module composed of a bottleneck layer and a domain adaptation layer is constructed and introduced to the domain adversarial neural network; thus, the domain adversarial adaptation network (DAAN) is designed. By simultaneously performing domain adversarial learning, reducing both the marginal distribution difference and second-order statistic difference between two domains, the distributions of the source and target domains are aligned. Then, the conditional distribution adaptation regularization term based on the maximum mean discrepancy is embedded into a broad learning system to obtain the conditional adaptation broad network (CABN). On the one hand, CABN can perform the class-level distribution adaptation on the domain-invariant features extracted by DAAN. On the other hand, the representation ability of the domain-invariant features expanded by CABN can be further enhanced. Experimental results on ten real hyperspectral data pairs show that, compared with the existing mainstream methods, DABAN can effectively utilize relevant source-domain information to assist in improving the classification accuracy of the target domain.

**Index Terms**—Adversarial learning, broad learning, classification, domain adaptation, hyperspectral image (HSI).

## I. INTRODUCTION

THE goal of hyperspectral image (HSI) classification is to determine the class of each pixel according to the spectral and spatial information of HSI [1]. With the

Manuscript received August 15, 2021; revised September 21, 2021; accepted November 10, 2021. Date of publication November 15, 2021; date of current version February 17, 2022. This work was supported by the National Natural Science Foundation of China under Grant 61976215, Grant 62176259, Grant 61772532, and Grant U1813203. (Corresponding author: Xuesong Wang.)

Haoyu Wang, Yuhu Cheng, and Xuesong Wang are with the Engineering Research Center of Intelligent Control for Underground Space, Ministry of Education, China University of Mining and Technology, Xuzhou 221116, China, and also with the School of Information and Control Engineering, China University of Mining and Technology, Xuzhou 221116, China (e-mail: wanghaoyucumt@163.com; chengyuhu@163.com; wangxuesongcumt@163.com).

C. L. Philip Chen is with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China, and also with the Department of Computer and Information Science, Faculty of Science and Technology, University of Macau, Macau (e-mail: philip.chen@ieee.org).

Digital Object Identifier 10.1109/TGRS.2021.3128162

continuous development of remote sensing technology, HSI classification, as one of the research hotspots in the field of remote sensing, has been successfully applied to many fields, such as urban planning, mineral identification, and agricultural production [2]–[4]. In recent years, many supervised learning methods for HSI classification have been proposed, including a support vector machine [5], sparse representations [6], and a convolutional neural network (CNN) [7]. The supervised learning methods usually require a large number of labeled samples to maintain high classification accuracy [8]. However, obtaining sufficient labeled HSIs is quite difficult, time-consuming, and expensive. Thus, how to learn a model with strong generalization ability at low labeling cost has become an important research topic in the field of HSI classification field. In recent years, active learning has been successfully applied to HSI classification, which increases the number of training samples by selecting the most informative samples from unlabeled samples for manual labeling [9], but manual labeling is more costly. In addition, the semisupervised learning technique, which can utilize both labeled and unlabeled samples, has also been successfully applied to HSI classification [10]. However, semisupervised learning is only applicable when the training and testing datasets are derived from the same distribution.

Active learning and semisupervised learning can alleviate the problem of insufficient training samples caused by the difficulty of HSI labeling to a certain extent. However, when the training and testing datasets come from similar but different HSIs, that is, when the data distributions of the two HSI datasets are similar but different, it is difficult for active learning and semisupervised learning to obtain satisfactory results. As a particular form of transfer learning, domain adaptation can solve the above problem. It enables knowledge transfer from a source domain to similar but different target domains [11]–[13]. Traditional domain adaptation methods are mainly classified into three types: distribution-based, feature selection-based, and subspace learning-based. The first method aims to shorten the distance between different data distributions by using certain transformations [14]. The goal of feature selection-based domain adaptation methods is to learn domain-invariant features of the two domains [15]. The last one tries to map the source- and target-domain data to a subspace, where different data can be aligned [16].

Compared with the above traditional domain adaptation methods, the deep transfer learning model can learn deeper and more transferable features, which has shown breakthrough results on many domain adaptation datasets [17]. This is because deep transfer learning networks can extract discriminative features and invariant factors underlying data, and effectively group the features according to their correlations to the invariant factors [18]. Long *et al.* [18] proposed the deep adaptation network (DAN) based on CNN, which adapts the marginal distributions of two domains by adding the multiple-kernel maximum mean discrepancy (MMD) to the loss function of CNN. Wang *et al.* [19] embedded a common representation space into a deep neural network and implemented domain adaptation by aligning the distributions of source and target domains. Sun and Saenko [20] proposed the correlation alignment for deep adaptation (DCORAL), which adapted the second-order statistic information from both domains. Recently, deep network-based domain adaptation methods have been successfully applied to HSI classification. Riz *et al.* [21] used a stacked autoencoder to extract domain-invariant features and then trained a classifier with these features. Zhou and Prasad [22] first trained a convolutional recurrent neural network to extract features of two domains and then achieved domain adaptation by mapping the extracted features layer-by-layer in a common subspace for alignment. Deng *et al.* [23] brought the clusters formed by deep features in the source and target domains as close as possible to achieve domain adaptation. Li *et al.* [24] proposed an unsupervised domain adaptation method named MMD\_RECON by using two autoencoder networks to map the two domain data to a common subspace, where the features of both domains are adapted via the MMD loss.

The domain adversarial neural network (DANN), which is evolved from the generative adversarial network, is consisted of a feature extractor, a classifier, and a domain discriminator. DANN can effectively extract deep domain-invariant features underlying data by combining adversarial learning with domain adaptation [25]. On the basis of DANN, Cicek and Soatto [26] introduced a predictor to learn the probability distributions about class and domain label, so as to adapt the marginal and conditional distributions simultaneously. Pei *et al.* [27] proposed the multiadversarial domain adaptation network, which achieved the distribution adaptation between two domains using a multidomain discriminator. Kang *et al.* [28] proposed a contrastive adaptation network, which accomplished the class-level domain adaptation by maximizing the interclass domain discrepancy and minimizing the intraclass domain discrepancy. Kurmi *et al.* [29] used the probabilistic certainty of the discriminator to improve the attention of the classifier for transferable features. For HSI classification tasks, domain adversarial networks have achieved satisfactory success. Elshamli *et al.* [30] accomplished an unsupervised classification of HSI using domain adversarial networks. Liu *et al.* [31] proposed a classwise distribution adaptation network, which combined the domain adversarial strategy with the probability-prediction-based maximum mean discrepancy for feature alignment and achieved an excellent classification performance of HSI. Wang *et al.* [32]

first extracted the domain transferable features of HSI with CycleGan and then introduced the CORAL loss to reduce the difference in second-order statistics between two domains. Ma *et al.* [33] used multiclassifiers to construct a discriminator, used variational autoencoders to build a generator, and then, utilized the adversarial idea to adapt the distribution difference between the source and target domains of HSI. Although the above domain adversarial methods have achieved good results, they have not achieved a hierarchical and comprehensive alignment of distributions between two domains.

Recently, Chen and Liu [34] proposed the broad learning system (BLS), also known as a broad neural network, based on a random vector functional link neural network, which has been widely used in various fields [35]–[37]. Compared with deep learning, BLS has a simple structure, is easy to combine with other models, and can achieve feature broad expansion [34]. Xu *et al.* [38] proposed a recurrent BLS, which enabled the network to have the ability to remember historical information by adding feedback connections to the enhancement nodes. Chu *et al.* [35] designed a weighted BLS by using penalty factors to constrain each input. Zhang *et al.* [39] proposed the GCB-Net model by integrating graph convolution, CNN, and BLS, improving the classification ability of the model. Kong *et al.* [40] combined the class probability framework with a broad learning network to obtain a semisupervised BLS (SBLS). Kong *et al.* [41] proposed unsupervised broad learning, which added a manifold regularization term to BLS and maintained the manifold structure of original HSI. Wang *et al.* [42] proposed the domain adaptation broad learning (DABL) classification method, which aligned the marginal and conditional distributions according to the importance of distribution. As we all know, affected by factors such as long-distance shooting and illumination, HSIs often have complex spectral features, which leads to nonlinear separability. Although the aforementioned broad network models are simple and flexible in structure, it is difficult to fully represent complex HSI data with linear sparse features.

Aiming at the problem that the testing and training sets are similar but different, a novel HSI classification method is proposed. The main contributions of our work include the following.

- 1) By comprehensively combining adversarial learning, broad learning, and transfer learning, we designed a domain adversarial broad adaptation network (DABAN), completing the classification of the target-domain HSI only using the labeled samples in the source domain.
- 2) We constructed a bottleneck adaptation module composed of a bottleneck layer and a domain adaptation layer and further added it to the original DANN to obtain the domain adversarial adaptation network (DAAN). By simultaneously performing domain adversarial learning and reducing the marginal distribution difference and second-order statistic difference of two domains, the distributions of the source and target domains are aligned.
- 3) We designed a conditional adaptation broad network (CABN) by embedding the MMD-based conditional distribution adaptation regularization term into the

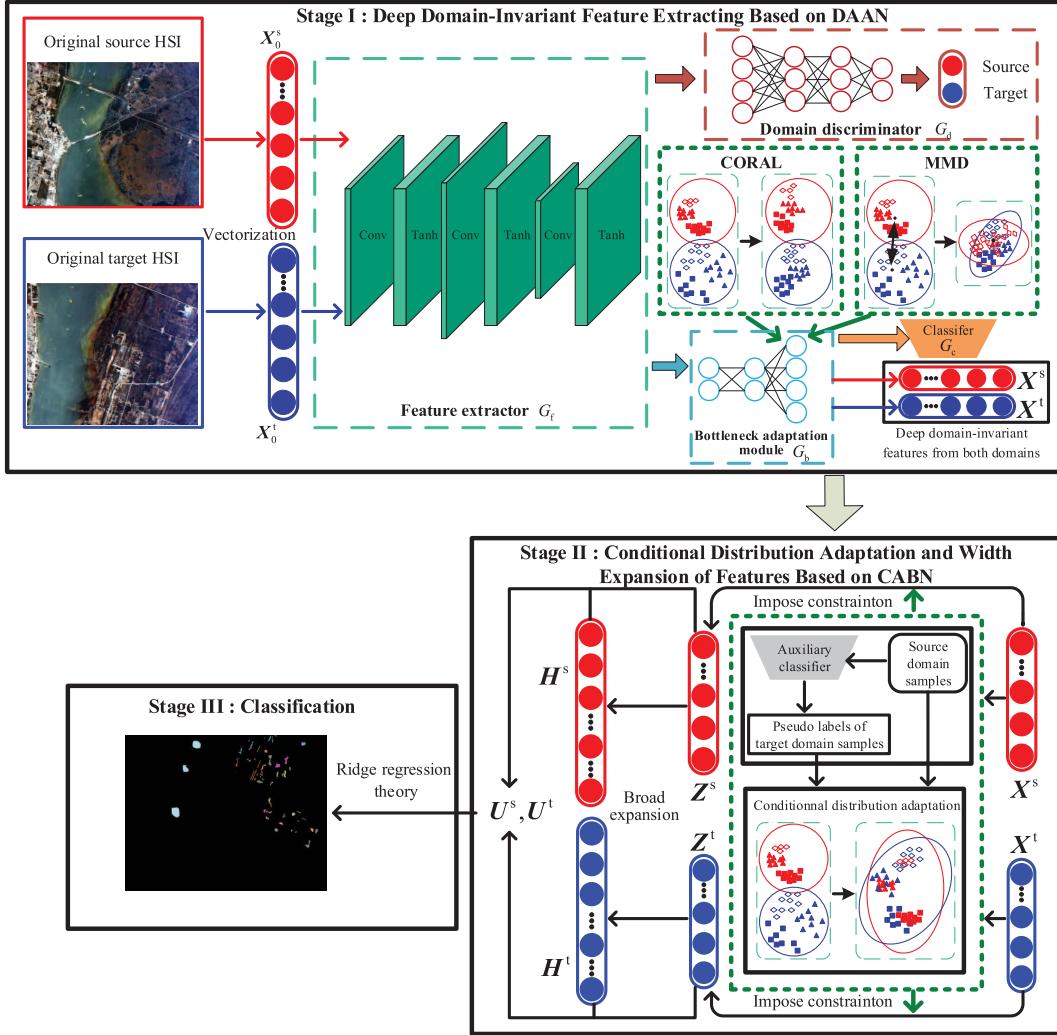


Fig. 1. Structure of DABAN for HSI classification.

feature mapping process of the original BLS. CABN can not only adapt the distribution of domain-invariant features extracted by DAAN at the class level but also further enhance the representation ability of domain-invariant features.

The remainder of this article is organized as follows. The proposed DABAN is elaborated in Section II. Experiments on ten HSI data pairs are shown in Section III, followed by a conclusion in Section IV.

## II. HSI CLASSIFICATION BASED ON DABAN

### A. Structure of DABAN

The structure of DABAN for HSI classification is shown in Fig. 1, which is mainly divided into three stages. The first stage is deep domain-invariant feature extracting based on DAAN. First, the original source and target HSIs are represented in the form of vectors to obtain  $X_0^s$  and  $X_0^t$ . Second, input  $X_0^s$  and  $X_0^t$  to feature extractor  $G_f$  to obtain deep features of two domains. Then, input the deep features of two domains to the domain discriminator  $G_d$  and the bottleneck adaptation module  $G_b$ , respectively, to adapt the source- and

target-domain distributions in an adversarial manner and, at the same time, adapt the marginal probability distributions and the second-order statistics of two domains. Finally, DAAN is trained according to labeled samples in the source domain and unlabeled samples in the target domain. Thus, the trained DAAN can be used to extract the deep domain-invariant features  $X^s$  and  $X^t$  of the original HSIs.

The second stage is conditional distribution adaptation and width expansion of features based on CABN. First, the pseudolabels of target-domain samples are obtained through the auxiliary classifier trained on the source domain. Second, according to the labels of the source-domain samples and pseudolabels of the target-domain samples, the conditional distribution adaptation regularization term is constructed based on MMD and applied to constrain the sparse autoencoder in BLS. Then, input  $X^s$  and  $X^t$  extracted by DAAN to CABN; thus, obtain  $Z^s$  and  $Z^t$ . Finally,  $Z^s$  and  $Z^t$  are expanded in width through random weights to obtain  $H^s$  and  $H^t$ .

The third stage is classification, i.e., to calculate the weight of the output layer of CABN according to the ridge regression theory and to predict labels of the target-domain HSI.

### B. DAAN

The DAAN consists of four parts: a feature extractor  $G_f$ , a domain discriminator  $G_d$ , a bottleneck adaptation module  $G_b$ , and a classifier  $G_c$ . The feature extractor  $G_f$  consists of 1-D convolutional and nonlinear layers, which is used to extract deep features of the original HSI. The domain discriminator  $G_d$  is used to discriminate whether the sample comes from the source or target domain. The bottleneck adaptation module  $G_b$  consists of a bottleneck layer and a domain adaptation layer with MMD and CORAL losses. The bottleneck layer is used to reduce the computational complexity of MMD and CORAL losses, and the domain adaptation layer is able to align the marginal distributions and second-order statistics of the two domains. The classifier  $G_c$  contains a fully connected layer, a nonlinear layer, and a softmax layer, which yields the prediction for the sample.

Define  $X_0^s \in \mathbb{R}^{n_s \times d}$  and  $X_0^t \in \mathbb{R}^{n_t \times d}$  as the source- and target-domain HSIs in the form of vectors,  $Y^s$  are class labels of the source samples, and  $D_s \in \mathbb{R}^{1 \times n_s}$  and  $D_t \in \mathbb{R}^{1 \times n_t}$  are the source- and target-domain labels, respectively. When  $D_s$  is 0,  $D_t$  is 1, and vice versa. The loss function of DAAN is defined as

$$\begin{aligned} L(X_0^s, X_0^t, Y^s, D_s, D_t; \theta_f, \theta_b, \theta_c, \theta_d) \\ = L_{\text{cls}}(X_0^s, Y^s; \theta_f, \theta_b, \theta_c) \\ - \lambda L_{\text{domain}}(X_0^s, X_0^t, D_s, D_t; \theta_f, \theta_d) \\ + \alpha_1 L_{\text{MMD}}(X_0^s, X_0^t; \theta_f, \theta_b) \\ + \alpha_2 L_{\text{CORAL}}(X_0^s, X_0^t; \theta_f, \theta_b) \end{aligned} \quad (1)$$

where  $L_{\text{cls}}(X_0^s, Y^s; \theta_f, \theta_b, \theta_c)$  is the source-domain classification loss,  $L_{\text{domain}}(X_0^s, X_0^t, D_s, D_t; \theta_f, \theta_d)$  is the domain discriminator classification loss,  $L_{\text{MMD}}(X_0^s, X_0^t; \theta_f, \theta_b)$  and  $L_{\text{CORAL}}(X_0^s, X_0^t; \theta_f, \theta_b)$  are adaptation losses of the marginal distribution and second-order statistics,  $\theta_f$ ,  $\theta_d$ ,  $\theta_b$ , and  $\theta_c$  are network parameters of the feature extractor, domain discriminator, bottleneck adaptation module, and classifier, respectively, and  $\lambda$ ,  $\alpha_1$ , and  $\alpha_2$  are predefined parameters.

For supervised multiclassification tasks, a cross-entropy function is normally used as the classification loss in deep networks, and the cross-entropy function  $L_s(\cdot, \cdot)$  can be expressed as

$$L_s(p, s) = - \sum_{c=1}^C s^c \log p^c \quad (2)$$

$$p^c = S(F_c) = \frac{\exp(F_c)}{\sum_{i=1}^C \exp(F_i)} \quad (3)$$

where  $p$  is the predicted probability output matrix,  $s$  is the label information,  $S(\cdot)$  represents the softmax activation function,  $p^c$  is the probability of the input belonging to the  $c$ th class,  $C$  is the total number of classes, and  $F_i$  represents the value in the  $i$ th dimension of the prediction value matrix.

The source-domain classification loss can be expressed as

$$\begin{aligned} L_{\text{cls}}(X_0^s, Y^s; \theta_f, \theta_b, \theta_c) \\ = \frac{1}{n_s} \sum_{(x_i, y_i) \sim (X_0^s, Y^s)} L_s(S(G_c(G_b(G_f(x_i)))), y_i) \end{aligned} \quad (4)$$

where  $n_s$  is the number of source-domain samples,  $x_i$  are the labeled source samples, and  $y_i$  are labels corresponding to  $x_i$ . The domain discriminator classification loss is

$$\begin{aligned} L_{\text{domain}}(X_0^s, X_0^t, D_s, D_t; \theta_f, \theta_d) \\ = \frac{1}{n_s + n_t} \sum_{(x_i, d_i) \sim (X_0^s \cup X_0^t, D_s \cup D_t)} L_s(S(G_d(G_f(x_i))), d_i) \end{aligned} \quad (5)$$

where  $n_t$  is the number of target-domain samples and  $d_i$  are domain labels corresponding to  $x_i$ . The adaptation losses of the marginal distribution and second-order statistic are defined as follows:

$$\begin{aligned} L_{\text{MMD}}(X_0^s, X_0^t; \theta_f, \theta_b) \\ = \left\| \frac{1}{n_s} \sum_{i=1}^{n_s} \vartheta(X_0^{s(i)}) - \frac{1}{n_t} \sum_{j=1}^{n_t} \vartheta(X_0^{t(j)}) \right\|_{\mathcal{H}}^2 \end{aligned} \quad (6)$$

$$\begin{aligned} L_{\text{CORAL}}(X_0^s, X_0^t; \theta_f, \theta_b) \\ = \frac{1}{4d} \|C_s - C_t\| \end{aligned} \quad (7)$$

where  $d$  is the dimension of the domain adaptation layer input, and  $C_s$  and  $C_t$  represent the data covariance matrices of the source domain and the target domain, respectively.  $\mathcal{H}$  is the reproducing kernel Hilbert space generated by  $\vartheta(\cdot)$ .

To find the optimal network parameters,  $\theta_f^*$ ,  $\theta_b^*$ ,  $\theta_c^*$ , and  $\theta_d^*$ , the objective function of DAAN can be expressed as

$$\begin{aligned} (\theta_f^*, \theta_b^*, \theta_c^*) \\ = \arg \min_{\theta_f, \theta_b, \theta_c} L(X_0^s, X_0^t, Y^s, D_s, D_t; \theta_f, \theta_b, \theta_c, \theta_d) \end{aligned} \quad (8)$$

$$\theta_d^* = \arg \max_{\theta_d} L(X_0^s, X_0^t, Y^s, D_s, D_t; \theta_f, \theta_b, \theta_c, \theta_d). \quad (9)$$

### C. CABN

In order to further adapt the deep domain-invariant features extracted by DAAN and enhance the feature representation ability, the conditional distribution adaptation regularization term is constructed based on MMD to constrain the sparse autoencoder in the original BLS, so as to obtain CABN. The workflow of CABN is given as follows:  $X = [X^s; X^t]$  extracted by DAAN are first mapped to “mapped features (MFs)” via the weight  $A$  and bias  $\beta_i^M$ . The  $i$ th group MF of CABN is

$$Z_i = \phi(XA + \beta_i^M), \quad i = 1, \dots, d^M \quad (10)$$

where  $Z_i \in \mathbb{R}^{(n_s+n_t) \times G^M}$ ,  $d^M$  is the number of groups of MFs,  $G^M$  is the dimension of each MF group, and  $\phi(\cdot)$  is the activation function. Then, MFs are mapped to enhanced nodes (ENs) through the random weight and bias to achieve the width expansion of features.

It's well known that aligning the source- and target-domain distributions at the class level requires labels for both domain samples. Therefore, an auxiliary classifier trained with labeled source-domain samples is used to classify target-domain samples to obtain corresponding pseudolabels. The conditional distribution adaptation between the source and

target domains can be realized by solving the following function:

$$\underset{A_i}{\operatorname{argmin}} \|XA_i - Z_i\|_2^2 + \|A_i\|_1 + \gamma T(P_s, P_t) \quad (11)$$

where  $\gamma$  is the regularization coefficient,  $P_s$  and  $P_t$  represent the conditional distributions of the source and target domains, respectively, and the conditional distribution adaptation regularization term  $T(P_s, P_t)$  is

$$\begin{aligned} T(P_s, P_t) &= \sum_{c=1}^C \|\mathbb{E}[f(z_s^{(c)})] - \mathbb{E}[f(z_t^{(c)})]\|_{\mathcal{H}}^2 \\ &= \operatorname{tr}(A^T X^T M X A) \end{aligned} \quad (12)$$

where  $f(\cdot)$  is the nonlinear mapping function in a reproducing kernel Hilbert space and  $M = \sum_{c=1}^C M_c$

$$(M_c)_{ij} = \begin{cases} \frac{1}{(n_s)_c^2}, & \mathbf{x}_i, \mathbf{x}_j \in D_s^{(c)} \\ \frac{1}{(n_t)_c^2}, & \mathbf{x}_i, \mathbf{x}_j \in D_t^{(c)} \\ -\frac{1}{(n_s)_c(n_t)_c}, & \begin{cases} \mathbf{x}_i \in D_s^{(c)}, \mathbf{x}_j \in D_t^{(c)} \\ \mathbf{x}_i \in D_t^{(c)}, \mathbf{x}_j \in D_s^{(c)} \end{cases} \\ 0, & \text{otherwise.} \end{cases} \quad (13)$$

Then, (12) can be expressed as

$$\underset{A_i}{\operatorname{argmin}} \|XA_i - Z_i\|_2^2 + \|A_i\|_1 + \gamma \operatorname{tr}(A_i^T X^T M X A_i). \quad (14)$$

Equation (14) can be solved through the alternating direction method of multipliers (ADMM). By introducing an auxiliary variable  $O$ , we obtain

$$\begin{aligned} &\underset{A_i, Z_i}{\operatorname{argmin}} \|XA_i - Z_i\|_2^2 + \|O\|_1 + \gamma \operatorname{tr}(A_i^T X^T M X A_i) \\ &\text{s.t. } A_i - O = 0. \end{aligned} \quad (15)$$

The Lagrangian expression of (15) is

$$\begin{aligned} J &= \underset{A_i}{\operatorname{argmin}} \|XA_i - Z_i\|_2^2 + \|O\|_1 \\ &+ \gamma \operatorname{tr}(A_i^T X^T M X A_i) \\ &+ \rho \eta^T(A_i - O) + \frac{\rho}{2} \|A_i - O\|_2^2 \end{aligned} \quad (16)$$

where  $\rho > 0$  is a constraint. According to ADMM,  $A_i$ ,  $O$ , and  $\eta$  are able to be updated alternately, i.e., one variable is updated, while the other two are fixed. The detailed workflow is described as follows.

1) The following equation yields  $A_i$ :

$$A_i^{(k+1)} = \underset{A_i}{\operatorname{argmin}} J(A_i, O, \eta). \quad (17)$$

Calculating the derivative of  $J$  with respect to  $A_i$  and making it zero, we obtain

$$A_i^{(k+1)} = \frac{X^T Z_i + \rho(O^{(k)} - \eta^{(k)})}{X^T X + \rho I + \gamma X^T M X}. \quad (18)$$

2) Updating  $O$

$$O^{(k+1)} = S_{\lambda/\rho}(A_i^{(k+1)} + \eta^{(k)}) \quad (19)$$

where  $S_k(\cdot)$  is a soft threshold operation [43] that can be calculated as

$$S_k(g) = \begin{cases} g - k, & g > k \\ 0, & |g| \leq k \\ g + k, & g < -k. \end{cases} \quad (20)$$

3) Updating  $\eta$

$$\eta^{(k+1)} = \eta^{(k)} + (A_i^{(k+1)} - O^{(k+1)}). \quad (21)$$

The above three steps are alternated until a predefined number of iterations is reached to calculate  $A_i$ . Calculated by (10), the MFs of the source and target domains are  $Z_i^s = X^s A_i$  and  $Z_i^t = X^t A_i$ . Mapping the MFs to ENs via the random weight  $W^E$  and bias  $\beta_j^E$ , we get

$$H = \phi(Z W^E + \beta_j^E), \quad j = 1, \dots, d^E \quad (22)$$

where  $Z = [Z_1, Z_2, \dots, Z_{d^M}]$ ,  $\phi(\cdot)$  is the tansig function here, and  $d^E$  denotes the number of ENs.

Finally, mapping both MFs and ENs to the output layer, the objective function of CABN is

$$\underset{W}{\operatorname{argmin}} \|U^s W - Y^s\|_2^2 + \delta \|W\|_2^2 \quad (23)$$

where  $U^s = [Z^s | H^s]$ , and  $\delta$  denotes the constrain on the sum of output layer weight  $W$ . Using the ridge regression theory to solve (23), we obtain

$$W = \frac{[Z^s | H^s]^T Y^s}{\delta I + [Z^s | H^s]^T [Z^s | H^s]}. \quad (24)$$

The predicted result of target HSI can be get as

$$Y^t = [Z^t | H^t] W. \quad (25)$$

### III. EXPERIMENTS

#### A. HSI Datasets

Real HSIs acquired by four remote sensing sensors are selected for the experiment.

The Botswana dataset (BOT5, BOT6, and BOT7) was obtained in May, June, and July by the Hyperion sensor for the same deltaic area of Okavango, Botswana, and South Africa. All three HSIs contain  $1476 \times 256$  pixels and nine surface objects. After removing noise, atmospheric and water absorption, and overlapping bands, 145 bands remain for the experiment. The specific information of the two HSIs can be seen in Table I. Any two HSIs were selected as data pairs for the experiments.

The Kennedy Space Center dataset (KSC1 and KSC3) was obtained by NASA in March 1996 by the AVIRIS sensor over Florida. The two HSIs include  $1476 \times 256$  pixels, 224 bands, and ten surface objects. After removing the absorbing and noisy bands, 176 bands of the images were retained for the experiment. The specific information of the KSC datasets can be seen in Table I. One of two HSIs is selected as the source domain and the other as the target domain.

The PaviaU and Houston datasets (H-P and P-H) consist of the Pavia University and Houston Bright datasets. Pavia University was captured by using ROSIS sensors over cities in northern Italy, including  $610 \times 610$  pixels, 103 bands, and nine

TABLE I  
DESCRIPTION OF HSI DATASETS

| Surface object     | BOT5 | BOT6 | BOT7 | Surface object        | KSC1 | KSC3 | Surface object | PaviaU | Houston |
|--------------------|------|------|------|-----------------------|------|------|----------------|--------|---------|
| Exposed soils      | 215  | 229  | 615  | Water                 | 927  | 1392 | Road           | 3113   | 1031    |
| Firescar           | 354  | 335  | 433  | Scrub                 | 761  | 422  | Grass          | 1789   | 1037    |
| Island interior    | 337  | 370  | 664  | Slash pine            | 161  | 166  | Tree           | 1324   | 1053    |
| Riparian           | 448  | 303  | 438  | Harwood swamp         | 105  | 248  | Roofs          | 1297   | 978     |
| Savanna            | 330  | 342  | 710  | Willow swamp          | 243  | 180  |                |        |         |
| Short mopane       | 239  | 299  | 330  | Graminoid marsh       | 432  | 453  |                |        |         |
| Primary floodplain | 437  | 308  | 584  | Salt marsh            | 419  | 156  |                |        |         |
| Woodlands          | 357  | 324  | 633  | Oak/broadleaf hammock | 229  | 274  |                |        |         |
| Water              | 297  | 361  | 590  | CP/oak hammock        | 252  | 132  |                |        |         |
|                    |      |      |      | CP hammock            | 256  | 431  |                |        |         |
| Total              | 3014 | 2871 | 4997 | Total                 | 3784 | 3854 | Total          | 7523   | 4099    |

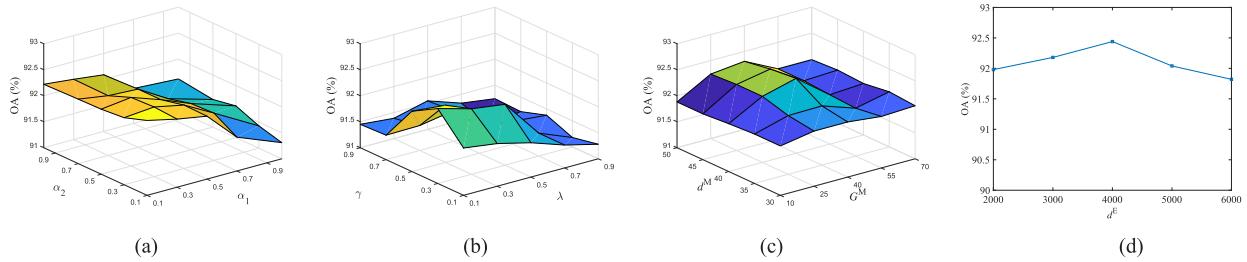


Fig. 2. OAs achieved by DABAN under different parameter settings (BOT5-BOT6). (a) OA versus  $\alpha_1$  and  $\alpha_2$ . (b) OA versus  $\lambda$  and  $\gamma$ . (c) OA versus  $G^M$  and  $d^M$ . (d) OA versus  $d^E$ .

types of definite surface objects. Houston Bright is the bright part of HSI captured over Houston city in the United States using ITRES-CASI 1500 sensor, including  $349 \times 1318$  pixels, 144 bands, and 15 types of definite surface objects. We select the central part of Pavia University and Houston Bright for the experiment. The specific information of the two HSIs can be seen in Table I. It can be observed from Table I that the number of each class is relatively balanced on the BOT, PaviaU, and Houston datasets, while the class imbalance problem is more prominent on the KSC dataset.

### B. Parameter Setting

When dealing with a single-domain classification task, researchers usually divide the dataset into a training set, a validation set, and a testing set. Cross-validation is usually used for parameter selection. However, due to different distributions of the source and target domains, Long *et al.* [44] pointed out that it is unrealistic to use cross-validation to obtain parameters in dealing with cross-domain classification tasks. Therefore, according to [44], the empirical search method is used to set parameters. It can be observed that the proposed DABAN is involved many parameters. Here, we mainly analyze the setting of several relatively important parameters, including  $\alpha_1$ ,  $\alpha_2$ ,  $\gamma$ ,  $\lambda$ ,  $G^M$ ,  $d^M$ , and  $d^E$ . For simplicity, other parameters are settled as the same, i.e., the learning rate is 0.001, the batchsize is 64, the maximum iteration is 2000, and the configuration of DAAN is shown in Table II. To obtain pseudolabels of the target-domain samples, a  $K$ -nearest neighbor classifier is used where  $K$  equals 7 and the Euclidean distance is used to measure the similarity between samples.

TABLE II  
CONFIGURATION OF DAAN

|                      | BOT |             | KSC |            | PaviaU & Houston |            |
|----------------------|-----|-------------|-----|------------|------------------|------------|
| Feature extractor    | C1  | Conv1d:7*8  | C1  | Conv1d:5*8 | C1               | Conv1d:7*8 |
|                      | N1  | Tanh        | N1  | Tanh       | N1               | Tanh       |
|                      | C2  | Conv1d:7*32 | C2  | Conv1d:7*8 | C2               | Conv1d:7*8 |
|                      | N2  | Tanh        | N2  | Tanh       | N2               | Tanh       |
|                      | C3  | Conv1d:5*32 | C3  | Conv1d:3*4 | C3               | Conv1d:5*8 |
|                      | N3  | Tanh        | N3  | Tanh       | N3               | Tanh       |
| Classifier           | Fc1 | size: 256   | Fc1 | size: 512  | Fc1              | size:64    |
|                      | Fc2 | size: 32    | Fc2 | size: 64   | Fc2              | size:64    |
| Domain discriminator | Fc1 | size: 64    | Fc1 | size: 512  | Fc1              | size:64    |
|                      | Fc2 | size: 512   | Fc2 | size: 1024 | Fc2              | size:512   |

We take the data pair BOT5-BOT6 as an example to analyze the parameter setting. First, we study the effect of  $\alpha_1$ ,  $\alpha_2$ ,  $\gamma$ , and  $\lambda$  on the overall accuracy (OA). Generally speaking, if the values of these four parameters are small, the domain adaptation degree between the source and target domains will be insufficient. On the contrary, it may lead to overadaptation. It can be seen from Fig. 2(a) that: 1) as  $\alpha_1$  increases, OA decreases; 2) OA is not greatly affected by parameter  $\alpha_2$ ; and 3) when  $\alpha_1 = 0.1$  and  $\alpha_2 = 0.1$ , OA reaches the highest. It can be seen from Fig. 2(b) that: 1) As  $\lambda$  increases, OA shows a downward trend; 2) as  $\gamma$  increases, OA first rises and then decreases; and 3) when  $\lambda = 0.1$  and  $\gamma = 0.3$ , OA reaches the highest. Next, we analyze the influence of parameters of BLS on OA. From Fig. 2(c) and (d), it can be observed that: 1) with the increase in  $G^M$ , OA takes the tendency of rising up first and then dropping down; 2) OA is not sensitive to



Fig. 3. Classification maps (BOT5-BOT6). (a) Ground truth. (b) BLS. (c) TCA. (d) JDA. (e) DANN. (f) DAN. (g) DCORAL. (h) DABL. (i) CNN. (j) SRNN. (k) SBLS. (l) MMD\_RECON. (m) DAAN. (n) DABAN.

parameter  $d^M$ ; 3) when  $G^M = 40$  and  $d^M = 40$ , OA reaches the highest; 4) as  $d^E$  increases, OA first rises and then falls; and 5) OA reaches the maximum when the number of ENs is equal to 4000.

It should be pointed out that, for simplicity, the parameter settings on other BOT data pairs are consistent with those on BOT5-BOT6. In the similar way, we can settle the parameters on KSC data pairs as follows:  $\alpha_1 = 0.1$ ,  $\alpha_2 = 0.1$ ,  $\lambda = 0.1$ ,  $\gamma = 0.5$ ,  $G^M = 10$ ,  $d^M = 60$ , and  $d^E = 1000$ . In addition, for H-P and P-H data pairs, the parameter settings are the same as BOT data pairs.

### C. Comparative Experiments

In order to verify the effectiveness and superiority of the proposed DABAN, the following 13 classification methods are selected for comparison:

- 1) *Supervised Learning Methods*: BLS [34], CNN [7], and scalable recurrent neural network (SRNN) [45].
- 2) *Semisupervised Learning Method*: SBLS [40].
- 3) *Traditional Transfer Learning Methods*: TCA [14] and joint distribution adaptation (JDA) [44].
- 4) *Deep Transfer Learning Methods*: DANN [25], DAN [18], DCORAL [20], and MMD\_RECON [24].
- 5) *Broad Transfer Learning Method*: DABL [42].
- 6) *DABAN Without CABN*: DAAN.

A total of ten data pairs are exploited for the experiment, i.e., BOT5-BOT6, BOT6-BOT5, BOT5-BOT7, BOT7-BOT5, BOT6-BOT7, BOT7-BOT6, KSC1-KSC3, KSC3-KSC1, H-P, and P-H. In each data pair, the first dataset represents the source domain, and the second one represents the target domain. To ensure the fairness of comparison, all experiments are done using the PyTorch for Python 3.6 on the computer with Intel Core i7-6850K CPU, 16 GB of

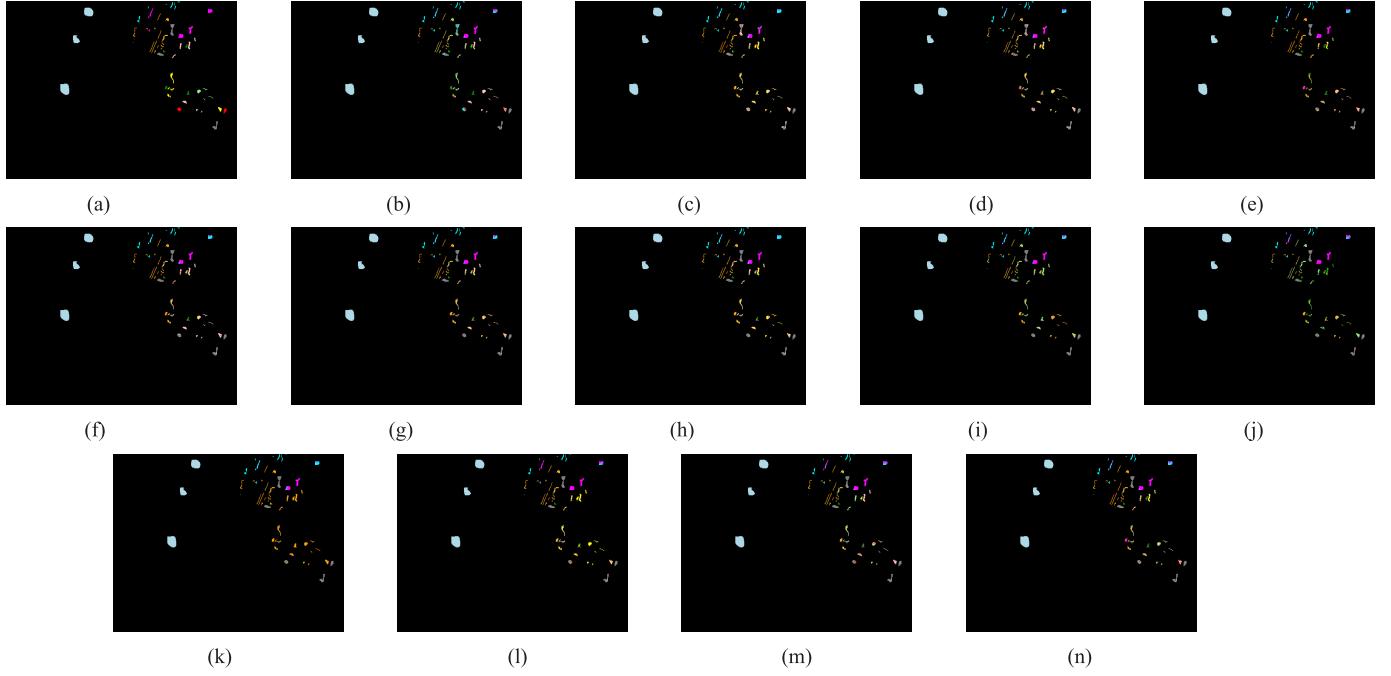


Fig. 4. Classification maps (KSC1-KSC3). (a) Ground truth. (b) BLS. (c) TCA. (d) JDA. (e) DANN. (f) DAN. (g) DCORAL. (h) DABL. (i) CNN. (j) SRNN. (k) SBLS. (l) MMD\_RECON. (m) DAAN. (n) DABAN.

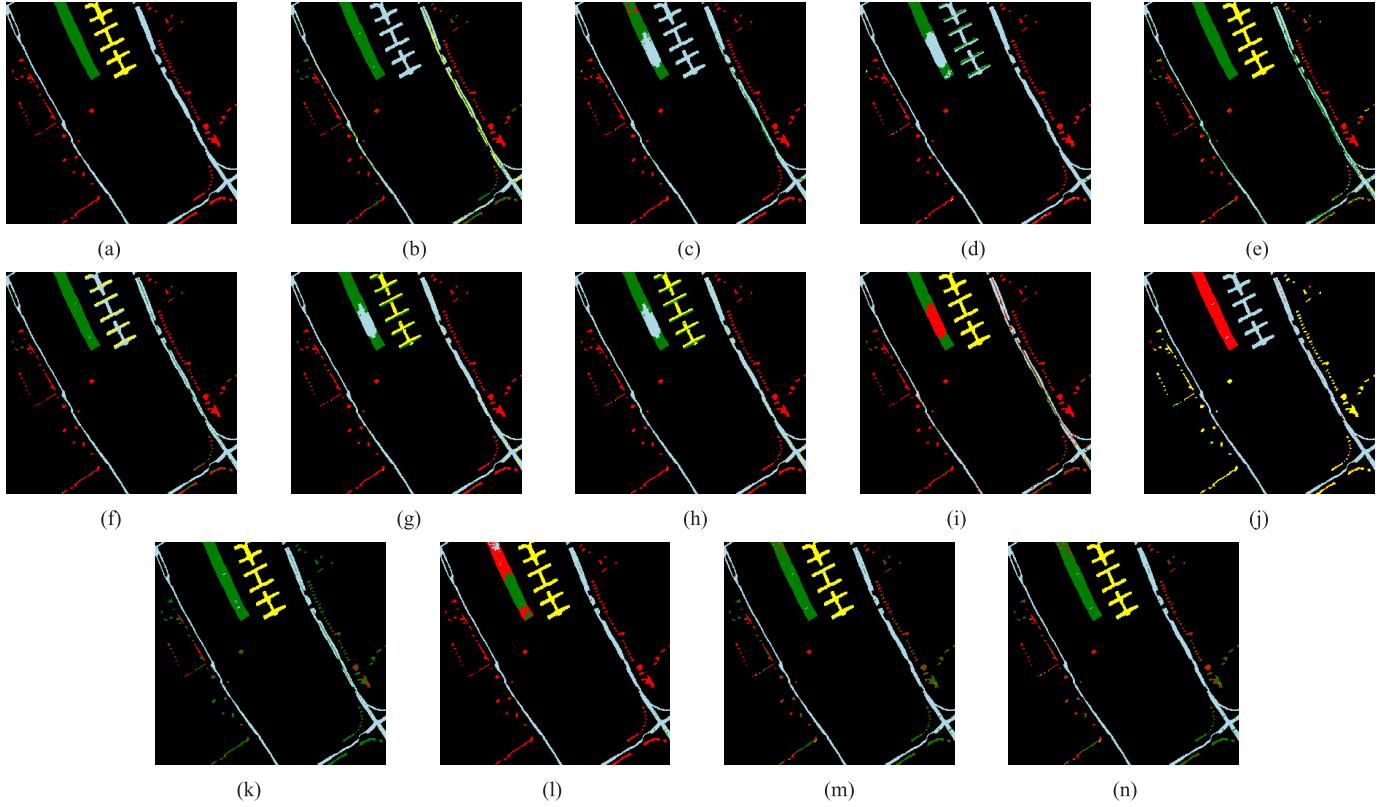


Fig. 5. Classification maps (H-P). (a) Ground truth. (b) BLS. (c) TCA. (d) JDA. (e) DANN. (f) DAN. (g) DCORAL. (h) DABL. (i) CNN. (j) SRNN. (k) SBLS. (l) MMD\_RECON. (m) DAAN. (n) DABAN.

RAM, and Nvidia GeForce GTX 1080 Ti. Three evaluating indexes, including OA (%), kappa coefficient, and consumed time (s), are used. Each experiment is performed

ten times independently, and then the average value is taken. Tables III–V show the classification results of all methods on ten data pairs, and Figs. 3–5 visibly

TABLE III  
COMPARISON OF CLASSIFICATION PERFORMANCE (OA, %)

| Datasets   |     | BLS   | TCA   | JDA   | DANN  | DAN   | DCORAL | DABL  | CNN   | SRNN  | SBLS  | MMD-RECON | DAAN  | DABAN        |
|------------|-----|-------|-------|-------|-------|-------|--------|-------|-------|-------|-------|-----------|-------|--------------|
| BOT        | 5-6 | 73.14 | 81.37 | 82.65 | 87.74 | 88.23 | 88.37  | 87.18 | 80.81 | 81.61 | 81.27 | 90.35     | 91.29 | <b>92.44</b> |
|            | 6-5 | 64.23 | 66.66 | 71.60 | 85.63 | 86.66 | 86.73  | 85.01 | 72.82 | 80.02 | 71.96 | 90.25     | 90.58 | <b>91.41</b> |
|            | 5-7 | 57.17 | 61.22 | 66.66 | 76.05 | 76.11 | 75.55  | 70.10 | 63.79 | 73.48 | 66.14 | 79.79     | 79.29 | <b>81.09</b> |
|            | 7-5 | 64.03 | 65.39 | 68.21 | 76.81 | 76.41 | 78.90  | 81.47 | 76.14 | 81.46 | 71.96 | 83.24     | 83.28 | <b>83.91</b> |
|            | 6-7 | 66.44 | 70.86 | 71.60 | 78.97 | 81.07 | 80.37  | 83.34 | 76.12 | 78.74 | 78.57 | 83.63     | 84.69 | <b>85.29</b> |
|            | 7-6 | 74.96 | 76.31 | 78.65 | 86.38 | 85.82 | 87.67  | 88.44 | 77.60 | 77.61 | 79.90 | 88.96     | 89.69 | <b>90.53</b> |
| KSC        | 1-3 | 62.40 | 65.23 | 66.66 | 72.88 | 71.04 | 70.94  | 72.04 | 67.12 | 65.48 | 67.44 | 73.53     | 73.53 | <b>74.63</b> |
|            | 3-1 | 58.56 | 69.61 | 71.44 | 70.06 | 68.97 | 70.90  | 70.37 | 69.45 | 71.40 | 70.14 | 72.78     | 72.81 | <b>73.47</b> |
| H-P<br>P-H | H-P | 65.82 | 69.53 | 71.95 | 83.11 | 82.71 | 84.54  | 84.54 | 78.77 | 80.89 | 80.71 | 86.06     | 87.32 | <b>88.02</b> |
|            | P-H | 67.79 | 69.96 | 72.48 | 81.89 | 83.12 | 85.32  | 84.43 | 77.44 | 80.03 | 81.05 | 87.33     | 88.03 | <b>89.14</b> |

TABLE IV  
COMPARISON OF CLASSIFICATION PERFORMANCE (KAPPA)

| Datasets   |     | BLS    | TCA    | JDA    | DANN   | DAN    | DCORAL | DABL   | CNN    | SRNN   | SBLS   | MMD-RECON | DAAN   | DABAN         |
|------------|-----|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|-----------|--------|---------------|
| BOT        | 5-6 | 0.6965 | 0.7902 | 0.8046 | 0.8618 | 0.8674 | 0.8688 | 0.8554 | 0.7834 | 0.7925 | 0.7923 | 0.8913    | 0.9018 | <b>0.9148</b> |
|            | 6-5 | 0.5970 | 0.6250 | 0.6793 | 0.8375 | 0.8494 | 0.8501 | 0.8304 | 0.7282 | 0.7784 | 0.6989 | 0.8896    | 0.8935 | <b>0.9028</b> |
|            | 5-7 | 0.5168 | 0.5601 | 0.6227 | 0.7296 | 0.7301 | 0.7243 | 0.6601 | 0.6146 | 0.7004 | 0.6412 | 0.7719    | 0.7660 | <b>0.7870</b> |
|            | 7-5 | 0.5897 | 0.6105 | 0.6421 | 0.7382 | 0.7338 | 0.7620 | 0.7914 | 0.7467 | 0.7928 | 0.6963 | 0.8106    | 0.8110 | <b>0.8180</b> |
|            | 6-7 | 0.6212 | 0.6713 | 0.6793 | 0.7619 | 0.7860 | 0.7788 | 0.8116 | 0.7612 | 0.7662 | 0.7648 | 0.8146    | 0.8280 | <b>0.8320</b> |
|            | 7-6 | 0.7167 | 0.7329 | 0.7591 | 0.8464 | 0.8402 | 0.8611 | 0.8696 | 0.7476 | 0.7477 | 0.7732 | 0.8754    | 0.8840 | <b>0.8930</b> |
| KSC        | 1-3 | 0.5428 | 0.5752 | 0.5932 | 0.6939 | 0.6441 | 0.6441 | 0.6561 | 0.5976 | 0.6335 | 0.6001 | 0.6745    | 0.6750 | <b>0.6820</b> |
|            | 3-1 | 0.5291 | 0.6482 | 0.6738 | 0.6520 | 0.6352 | 0.6604 | 0.6538 | 0.6701 | 0.6738 | 0.6732 | 0.6796    | 0.6800 | <b>0.6920</b> |
| H-P<br>P-H | H-P | 0.5053 | 0.5917 | 0.6245 | 0.7687 | 0.7480 | 0.7784 | 0.7785 | 0.7121 | 0.7543 | 0.7438 | 0.8058    | 0.8199 | <b>0.8300</b> |
|            | P-H | 0.5680 | 0.5993 | 0.6308 | 0.7586 | 0.7742 | 0.8039 | 0.7915 | 0.6984 | 0.7315 | 0.6836 | 0.8308    | 0.8402 | <b>0.8551</b> |

TABLE V  
COMPARISON OF CLASSIFICATION PERFORMANCE (CONSUMED TIME, S)

| Datasets   |     | BLS   | TCA  | JDA    | DANN    | DAN     | DCORAL  | DABL  | CNN    | SRNN   | SBLS   | MMD-RECON | DAAN    | DABAN   |
|------------|-----|-------|------|--------|---------|---------|---------|-------|--------|--------|--------|-----------|---------|---------|
| BOT        | 5-6 | 9.35  | 4.28 | 85.36  | 2036.91 | 3199.41 | 2213.87 | 16.27 | 465.24 | 203.79 | 729.16 | 4392.00   | 2245.17 | 2318.10 |
|            | 6-5 | 8.61  | 3.83 | 85.91  | 2357.73 | 3215.68 | 2223.65 | 18.47 | 366.92 | 256.97 | 613.40 | 4435.56   | 2475.58 | 2544.94 |
|            | 5-7 | 9.84  | 5.79 | 128.84 | 2362.31 | 3291.09 | 2214.83 | 18.06 | 481.14 | 253.62 | 793.67 | 4523.57   | 2583.66 | 2652.15 |
|            | 7-5 | 11.51 | 5.76 | 129.20 | 1707.78 | 3255.29 | 1611.24 | 18.97 | 492.57 | 225.79 | 635.48 | 3791.03   | 1694.97 | 1757.31 |
|            | 6-7 | 9.67  | 5.80 | 125.77 | 2359.95 | 3193.57 | 2219.06 | 21.13 | 341.41 | 265.86 | 607.79 | 4256.56   | 2349.55 | 2417.11 |
|            | 7-6 | 11.28 | 6.10 | 127.85 | 1617.63 | 3197.29 | 1609.43 | 19.97 | 360.72 | 238.45 | 649.36 | 3615.84   | 1723.88 | 1789.31 |
| KSC        | 1-3 | 10.74 | 5.24 | 134.03 | 2368.97 | 3195.04 | 2211.44 | 20.55 | 460.11 | 269.12 | 896.13 | 4591.24   | 2468.55 | 2543.05 |
|            | 3-1 | 10.21 | 5.55 | 132.08 | 2371.33 | 3206.31 | 2213.64 | 20.74 | 446.56 | 300.47 | 716.58 | 4627.45   | 2582.50 | 2659.28 |
| H-P<br>P-H | H-P | 14.45 | 7.24 | 163.69 | 4872.38 | 6587.57 | 2569.35 | 49.44 | 378.16 | 264.18 | 671.49 | 8325.82   | 5826.24 | 5920.56 |
|            | P-H | 13.65 | 6.94 | 154.24 | 4798.32 | 6387.88 | 2548.10 | 46.19 | 349.72 | 277.28 | 721.34 | 7649.34   | 4834.96 | 4925.24 |

show the classification maps on BOT5-BOT6, KSC1-KSC3, and H-P.

It can be observed from Tables III–V and Figs. 3–5 that the following holds.

- 1) The OAs and kappa coefficients achieved by all methods on BOT5-BOT6 are higher than BOT6-BOT5. Similarly, BOT7-BOT5 is higher than BOT5-BOT7, and BOT7-BOT6 is higher than BOT6-BOT7. In other words, for BOT5, BOT6, and BOT7 originating from the same HSI dataset, the classification accuracy is higher when the sample size of the source domain is larger than that of the target domain. This is because a large

amount of source-domain samples can ensure sufficient training of the classification model, thus achieving high classification accuracy on the target domain.

- 2) Both TCA and DAN aim to reduce the marginal distribution discrepancy of the two domains. Compared with TCA, DAN has higher OAs and kappa coefficients on all data pairs. The reason is that DAN can extract deeper and more discriminative features of HSI.
- 3) Compared with TCA, JDA achieves higher OAs and kappa coefficients on all data pairs; this is because JDA not only aligns the marginal distributions but also aligns the conditional distributions of two domains.

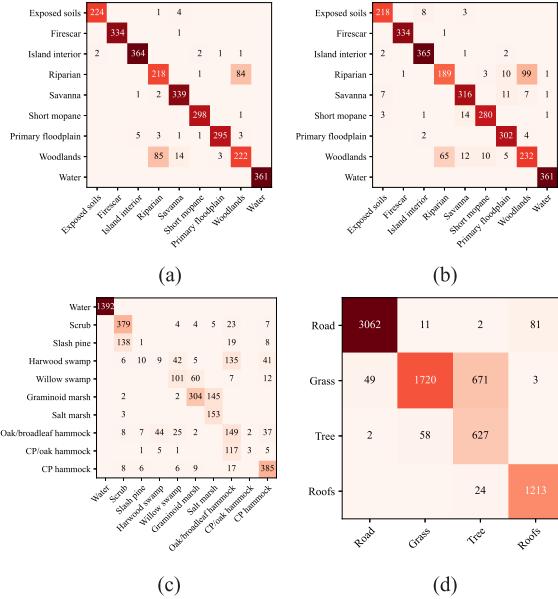


Fig. 6. Confusion matrix. (a) BOT5-BOT6. (b) BOT7-BOT6. (c) KSC1-KSC3. (d) H-P.

- 4) Both DANN and DAAN belong to DANNs. DAAN outperforms DANN on OA and kappa coefficient. This is because DANN only aligns the source- and target-domain distributions via an adversarial adaptation, while DAAN reduces the cross-domain discrepancy simultaneously in three different manners.
- 5) Among the 13 classification methods, DABAN achieved the highest OA and kappa coefficients on all data pairs with a little sacrifice of the consumed time. The reasons are analyzed as follows. First, DABAN is able to extract deep and discriminative features from the original HSI data. Second, DABAN performs a hierarchical and comprehensive alignment of distributions between the source and target domains. Finally, the width expansion is performed on domain-invariant features to further enhance the feature representation ability.

Fig. 6 shows the confusion matrices of DABAN on four HSI data pairs, including BOT5-BOT6, BOT7-BOT6, KSC1-KSC3, and H-P. It can be seen from Fig. 6 that the following holds.

- 1) Among all the surface objects from the Botswana dataset, Riparian and Woodlands are the most difficult to be recognized. Specifically, 84 Riparian and 85 Woodlands are misclassified as each other on BOT5-BOT6, while 99 Riparian and 65 Woodlands are misclassified as each other on BOT7-BOT6. This is because the spectral features of Riparian and Woodlands are relatively similar, which provides the biggest hurdle to train DABAN.
- 2) For the KSC1-KSC3 data pair, the classification accuracies of Slash pine, Harwood swamp, and CP/oak hammock are very low. The main reasons are analyzed as follows. On the one hand, the class imbalance phenomenon makes the classification accuracies of these surface objects with a small number of samples lower.

On the other hand, these classes are comprised of mixtures. Spectral changes and mixed spectral features make domain adaptation in these surface objects even more difficult.

- 3) For the H-P data pair, many surface objects of grass and tree are misclassified as each other because of their similar spectral features.

The original features and the features extracted by DABAN were visualized to obtain Fig. 7. It can be observed that: 1) compared with the original features on the target HSI domain, the covariance differences and the distance of centroids between the source and target domains are reduced and 2) for surface objects whose features have been extracted by DABAN, the surface objects from the same class are much closer, while the surface objects from different classes are far away from each other. Thus, the classification accuracy of DABAN is improved.

Finally, we investigated the alignment effects of several surface objects from BOT5-BOT6. It is easily seen from Fig. 8 that, compared with BLS without transfer learning ability, DABAN makes the features of the same class from the source and target domains are aligned much better, demonstrating the perfect domain adaptation ability of DABAN.

#### D. Influence of Different Band Combinations

In order to show the influence of different band combinations on the performance of DABAN, we took BOT5-BOT6 as an example and selected 64 bands from the source and target domains, respectively, according to three different manners to conduct experiments. The specific band selection manners are given as follows.

- 1) The first band selection manner is denoted as SS. We select  $a$  and  $b$  bands from the source and target domains with  $\lfloor N_b/N_d \rfloor$  and  $\lfloor N_b/N_d \rfloor + 1$  as the interval number, respectively [46], where  $N_b$  represents the number of original bands,  $N_d$  is the number of selected bands, and  $\lfloor \cdot \rfloor$  represents the floor operation. Then, the following equation can be obtained:

$$\begin{cases} a + b = N_d \\ \lfloor N_b/N_d \rfloor * a + (\lfloor N_b/N_d \rfloor + 1) * b = N_b. \end{cases} \quad (26)$$

- 2) The second band selection manner is called HSR. The first 32 bands of two domains are selected according to the method presented in [46], and the last 32 bands are randomly selected from the remaining bands that have not been selected.
- 3) The third band selection manner is denoted as SR. The 64 bands in the source domain are selected according to the method in [46], and the 64 bands in the target domain are all randomly selected.

According to the above three band selection manners, we use DABAN to classify the target-domain HSI. Each experiment was repeated ten times, and the average value was recorded. It can be seen from Table VI that: 1) when the manner of SS is used to select bands, both DAAN and DABAN obtain the highest OA and kappa coefficient, while

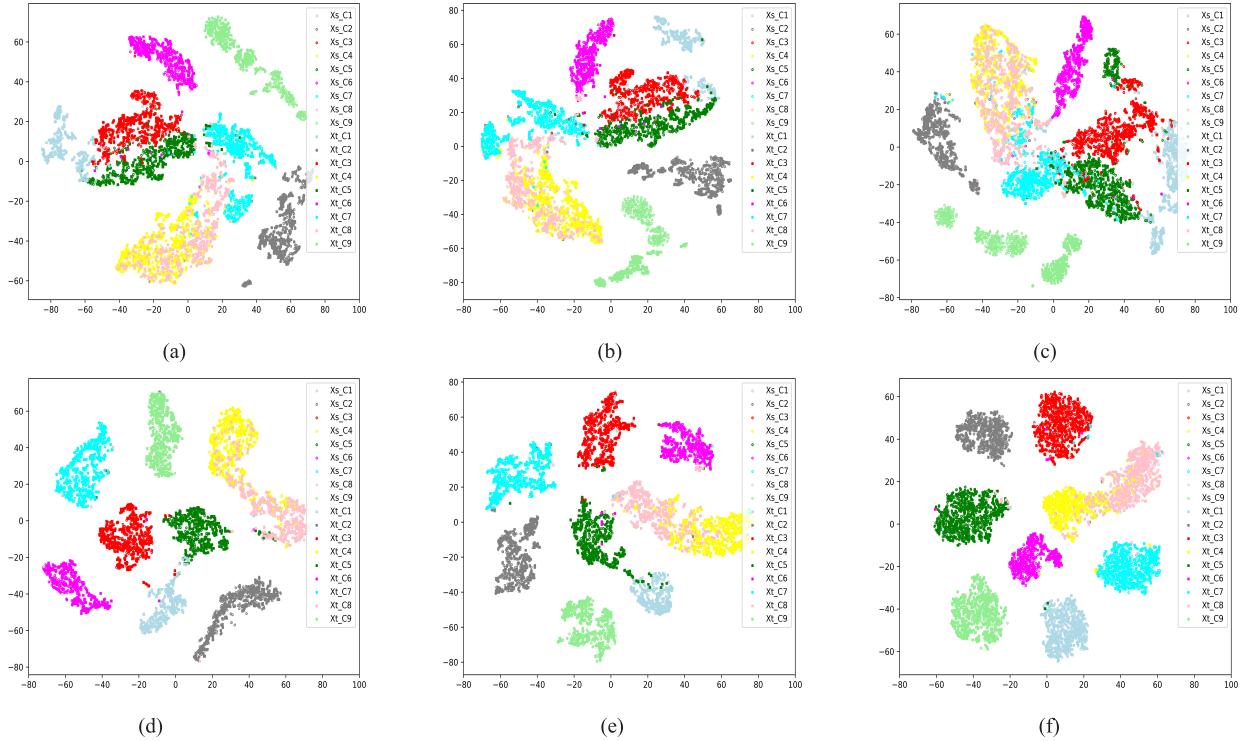


Fig. 7. t-SNE visualization of features. (a) Features of the original HSI (BOT5-BOT6). (b) Features of the original HSI (BOT6-BOT5). (c) Features of the original HSI (BOT7-BOT6). (d) Features extracted by DABAN (BOT5-BOT6). (e) Features extracted by DABAN (BOT6-BOT5). (f) Features extracted by DABAN (BOT7-BOT6).

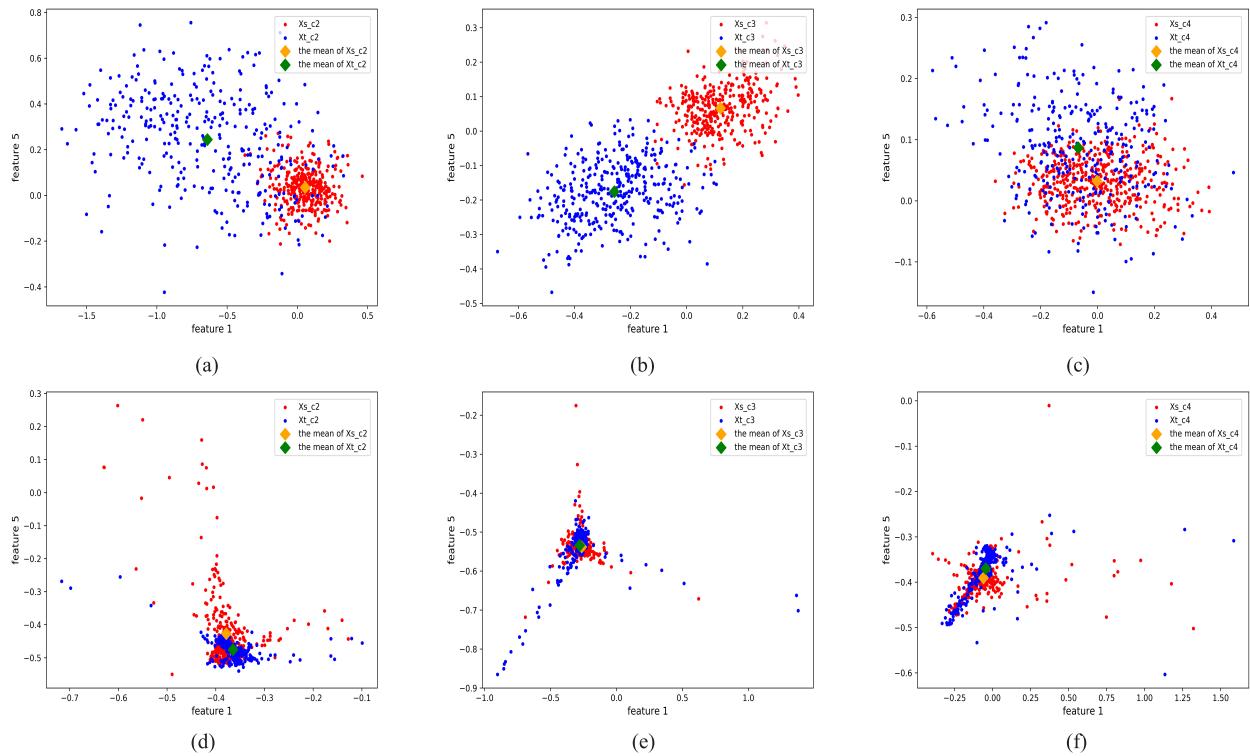


Fig. 8. Alignment performance of the BOT5-BOT6 data. (a) Class 2 with BLS. (b) Class 3 with BLS. (c) Class 4 with BLS. (d) Class 2 with DABAN. (e) Class 3 with DABAN. (f) Class 4 with DABAN.

the smallest consumed time and 2) when the manner of SR is used to select bands, both DAAN and DABAN show the

poorest classification performance. This is because when SS is used to select bands for the source and target domains, these

TABLE VI  
COMPARISON OF CLASSIFICATION PERFORMANCE

|          | DAAN          |        |        | DABAN         |        |        |
|----------|---------------|--------|--------|---------------|--------|--------|
|          | SS            | HSR    | SR     | SS            | HSR    | SR     |
| OA (%)   | <b>89.58</b>  | 87.53  | 84.88  | <b>90.42</b>  | 88.08  | 85.89  |
| Kappa    | <b>0.8826</b> | 0.8594 | 0.8297 | <b>0.892</b>  | 0.8657 | 0.8409 |
| Time (s) | <b>1673.3</b> | 1693.7 | 1712.7 | <b>1737.7</b> | 1767.2 | 1794.9 |

two domains will get a larger number of the same bands, which is helpful for the training of DAAN and DABAN.

#### IV. CONCLUSION

With the continuous development of remote sensing technology, more and more HSIs are captured by sensors. HSI classification, as one of the basic and key technologies in the field of remote sensing, can use HSIs to predict the class of each pixel on the ground. However, labeling these pixels requires expensive costs and a lot of labor. Therefore, a DABAN-based HSI classification method is proposed. The main advantages of DABAN can be summarized as follows: 1) the classification of HSIs in the target domain can be achieved by only using the labeled samples from a relevant source domain; 2) by combining the ideas of adversarial learning and transfer learning, DABAN can align the distributions of two domains comprehensively and hierarchically, and transfer the knowledge from the labeled source domain to the unlabeled target domain; and 3) in combination with the broad learning method, DABAN can expand the width of discriminative domain-invariant features, which further enhances the representation ability of features and, thus, improves the classification ability.

#### REFERENCES

- [1] L. Zhang, L. Zhang, and B. Du, "Deep learning for remote sensing data: A technical tutorial on the state of the art," *IEEE Geosci. Remote Sens. Mag.*, vol. 4, no. 2, pp. 22–40, Jun. 2016.
- [2] W. Li, C. Chen, H. Su, and Q. Du, "Local binary patterns and extreme learning machine for hyperspectral imagery classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 7, pp. 3681–3693, Jul. 2015.
- [3] L. Ma, M. M. Crawford, and J. Tian, "Local manifold learning-based k-nearest-neighbor for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 11, pp. 4099–4109, Nov. 2010.
- [4] Y. Ding, S. Pan, and Y. Chong, "Robust spatial-spectral block-diagonal structure representation with fuzzy class probability for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 3, pp. 1747–1762, Mar. 2020.
- [5] F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 8, pp. 1778–1790, Aug. 2004.
- [6] Y. Chen, N. M. Nasrabadi, and T. D. Tran, "Hyperspectral image classification using dictionary-based sparse representation," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 10, pp. 3973–3985, Oct. 2011.
- [7] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, "Deep feature extraction and classification of hyperspectral images based on convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 10, pp. 6232–6251, Oct. 2016.
- [8] M. S. Aydemir and G. Bilgin, "Semisupervised hyperspectral image classification using deep features," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 9, pp. 3615–3622, Sep. 2019.
- [9] C. Liu, J. Li, and L. He, "Superpixel-based semisupervised active learning for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 1, pp. 357–370, Jan. 2019.
- [10] A. Qin, Z. Shang, J. Tian, Y. Wang, T. Zhang, and Y. Y. Tang, "Spectral-spatial graph convolutional networks for semisupervised hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 2, pp. 241–245, Sep. 2019.
- [11] D. Tuia, C. Persello, and L. Bruzzone, "Domain adaptation for the classification of remote sensing data: An overview of recent advances," *IEEE Geosci. Remote Sens. Mag.*, vol. 4, no. 2, pp. 41–57, Jun. 2016.
- [12] X. Li, L. Liang, B. Du, and L. Zhang, "On gleaned knowledge from cross domains by sparse subspace correlation analysis for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 6, pp. 3204–3220, Jun. 2019.
- [13] M. Long, H. Zhu, J. Wang, and M. Jordan, "Unsupervised domain adaptation with residual transfer networks," in *Proc. Conf. Neural Inf. Process. Syst.*, 2016, pp. 136–144.
- [14] 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.
- [15] J. Blitzer, R. McDonald, and F. Pereira, "Domain adaptation with structural correspondence learning," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2006, pp. 22–23.
- [16] 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.
- [17] X. Zhou and S. Prasad, "Domain adaptation for robust classification of disparate hyperspectral images," *IEEE Trans. Comput. Imag.*, vol. 3, no. 4, pp. 822–836, Dec. 2017.
- [18] M. Long, Y. Cao, Z. Cao, J. Wang, and M. I. Jordan, "Transferable representation learning with deep adaptation networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 12, pp. 3071–3085, Dec. 2019.
- [19] Z. Wang, B. Du, and Y. Guo, "Domain adaptation with neural embedding matching," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 7, pp. 2387–2397, Jul. 2020.
- [20] B. Sun and K. Saenko, "Deep CORAL: Correlation alignment for deep domain adaptation," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 443–450.
- [21] E. Riz, B. Demir, and L. Bruzzone, "Domain adaptation based on deep denoising auto-encoders for classification of remote sensing images," *Proc. SPIE*, vol. 10004, Oct. 2016, Art. no. 100040K.
- [22] X. Zhou and S. Prasad, "Deep feature alignment neural networks for domain adaptation of hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 10, pp. 5863–5872, Oct. 2018.
- [23] B. Deng, S. Jia, and D. Shi, "Deep metric learning-based feature embedding for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 2, pp. 1422–1435, Feb. 2020.
- [24] W. Li, W. Wei, L. Zhang, C. Wang, and Y. Zhang, "Unsupervised deep domain adaptation for hyperspectral image classification," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2019, pp. 1–4.
- [25] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by back-propagation," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1180–1189.
- [26] S. Cicek and S. Soatto, "Unsupervised domain adaptation via regularized conditional alignment," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1416–1425.
- [27] Z. Pei, Z. Cao, M. Long, and J. Wang, "Multi-adversarial domain adaptation," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 3934–3941.
- [28] G. Kang, L. Jiang, Y. Yang, and A. G. Hauptmann, "Contrastive adaptation network for unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4893–4902.
- [29] V. K. Kurmi, S. Kumar, and V. P. Namboodiri, "Attending to discriminative certainty for domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 491–500.
- [30] A. Elshamli, G. W. Taylor, A. Berg, and S. Areibi, "Domain adaptation using representation learning for the classification of remote sensing images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 9, pp. 4198–4209, Sep. 2017.
- [31] Z. Liu, L. Ma, and Q. Du, "Class-wise distribution adaptation for unsupervised classification of hyperspectral remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 1, pp. 508–521, Jan. 2021.
- [32] X. Wang, Y. Li, and Y. Cheng, "Hyperspectral image classification based on unsupervised heterogeneous domain adaptation CycleGAN," *Chin. J. Electron.*, vol. 29, no. 4, pp. 608–614, Jul. 2020.
- [33] X. Ma, X. Mou, J. Wang, X. Liu, J. Geng, and H. Wang, "Cross-dataset hyperspectral image classification based on adversarial domain adaptation," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 5, pp. 4179–4190, May 2021.

- [34] C. L. P. Chen and Z. Liu, "Broad learning system: An effective and efficient incremental learning system without the need for deep architecture," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 1, pp. 10–24, Jan. 2018.
- [35] F. Chu, T. Liang, C. L. P. Chen, X. Wang, and X. Ma, "Weighted broad learning system and its application in nonlinear industrial process modeling," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 8, pp. 3017–3031, Aug. 2020.
- [36] L. Guo, R. Li, and B. Jiang, "A cascade broad neural network for concrete structural crack damage automated classification," *IEEE Trans. Ind. Informat.*, vol. 17, no. 4, pp. 2737–2742, Apr. 2021.
- [37] J. Feng, Y. Yao, S. Lu, and Y. Liu, "Domain knowledge-based deep-broad learning framework for fault diagnosis," *IEEE Trans. Ind. Electron.*, vol. 68, no. 4, pp. 3454–3464, Apr. 2021.
- [38] M. Xu, M. Han, C. L. P. Chen, and T. Qiu, "Recurrent broad learning systems for time series prediction," *IEEE Trans. Cybern.*, vol. 50, no. 4, pp. 1405–1417, Apr. 2020.
- [39] T. Zhang, X. Wang, X. Xu, and C. L. P. Chen, "GCB-net: Graph convolutional broad network and its application in emotion recognition," *IEEE Trans. Affect. Comput.*, early access, Aug. 27, 2019, doi: 10.1109/TAFFC.2019.2937768.
- [40] Y. Kong, X. Wang, Y. Cheng, and C. L. P. Chen, "Hyperspectral imagery classification based on semi-supervised broad learning system," *Remote Sens.*, vol. 10, no. 5, p. 685, Apr. 2018.
- [41] Y. Kong, Y. Cheng, C. L. P. Chen, and X. Wang, "Hyperspectral image clustering based on unsupervised broad learning," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 11, pp. 1741–1745, Nov. 2019.
- [42] H. Wang, X. Wang, C. L. P. Chen, and Y. Cheng, "Hyperspectral image classification based on domain adaptation broad learning," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 3006–3018, Jun. 2020.
- [43] D. L. Donoho and J. M. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," *Biometrika*, vol. 81, no. 3, pp. 425–455, 1994.
- [44] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer feature learning with joint distribution adaptation," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2200–2207.
- [45] M. E. Paoletti, J. M. Haut, J. Plaza, and A. Plaza, "Scalable recurrent neural network for hyperspectral image classification," *J. Supercomput.*, vol. 76, pp. 8866–8882, Feb. 2020.
- [46] Y. Liu, L. Gao, C. Xiao, Y. Qu, K. Zheng, and A. Marinoni, "Hyperspectral image classification based on a shuffled group convolutional neural network with transfer learning," *Remote Sens.*, vol. 12, no. 11, p. 1780, Jun. 2020.



**Haoyu Wang** received the M.S. degree from the China University of Mining and Technology, Xuzhou, China, in 2017, where he is currently pursuing the Ph.D. degree with the School of Information and Control Engineering.

His main research interests include hyperspectral image analysis and transfer learning.



**Yuhu Cheng** (Member, IEEE) received the Ph.D. degree in control science and technology from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2005.

He is currently a Professor with the School of Information and Control Engineering, China University of Mining and Technology, Xuzhou, China. His main research interests include machine learning and intelligent systems.



**C. L. Philip Chen** (Fellow, IEEE) received the M.S. degree in electrical engineering from the University of Michigan, Ann Arbor, MI, USA, in 1985, and the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, IN, USA, in 1988.

He is currently the Dean of the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China, and a Chair Professor with the Department of Computer and Information Science, Faculty of Science and Technology, University of Macau, Macau. His research interests include computational intelligence, systems, and cybernetics.

Dr. Chen is also a fellow of the American Association for the Advancement of Science (AAAS), the Chinese Association of Automation, and the Hong Kong Institution of Engineers (HKIE). He is also the Chair of the TC 9.1 Economic and Business Systems of the International Federation of Automatic Control (IFAC). He is also an Accreditation Board of Engineering and Technology Education (ABET) Program Evaluator for Computer Engineering, Electrical Engineering, and Software Engineering Programs. He was the President of the IEEE Systems, Man, and Cybernetics Society from 2012 to 2013. He has been the Editor-in-Chief of the *IEEE TRANSACTIONS ON CYBERNETICS* since 2020 and an Associate Editor of several *IEEE TRANSACTIONS*.



**Xuesong Wang** (Member, IEEE) received the Ph.D. degree in control science and technology from the China University of Mining and Technology, Xuzhou, China, in 2002.

She is currently the Dean of the Engineering Research Center of Intelligent Control for Underground Space, Ministry of Education, the Dean of the Xuzhou Key Laboratory of Artificial Intelligence and Big Data, and a Professor with the School of Information and Control Engineering, China University of Mining and Technology. Her main research interests include reinforcement learning, machine learning, and pattern recognition.

Dr. Wang is also an Associate Editor of *IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS*, *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS*, *International Journal of Machine Learning and Cybernetics*, *Acta Automatica Sinica*, and *Acta Electronica Sinica*.