Confident Learning-Based Domain Adaptation for Hyperspectral Image Classification

Zhuoqun Fang, Yuexin Yang, Zhaokui Li, Wei Li, *Senior Member, IEEE*, Yushi Chen, *Member, IEEE*, Li Ma, *Member, IEEE*, and Qian Du, *Fellow, IEEE*

***Abstract*— Cross-domain hyperspectral image classification is one of the major challenges in remote sensing, especially for target domain data without labels. Recently, deep learning approaches have demonstrated effectiveness in domain adaptation. However, most of them leverage unlabeled target data only from a statistical perspective but neglect the analysis at the instance level. For better statistical alignment, existing approaches employ the entire unevaluated target data in an unsupervised manner, which may introduce noise and limit the discriminability of the neural networks. In this article, we propose confident learning-based domain adaptation (CLDA) to address the problem from a new perspective of data manipulation. To this end, a novel framework is presented to combine domain adaptation with confident learning (CL), where the former reduces the interdomain discrepancy and generates pseudo-labels for the target instances, from which the latter selects high-confidence target samples. Specifically, the confident learning part evaluates the confidence of each pseudo-labeled target sample based on the assigned labels and the predicted probabilities. Then, high-confidence target samples are selected as training data to increase the discriminative capacity of the neural networks. In addition, the domain adaptation part and the confident learning part are trained alternately to progressively increase the proportion of high-confidence labels in the target domain, thus further improving the accuracy of classification. Experimental results on four datasets demonstrate that the proposed CLDA method outperforms the state-of-the-art domain adaptation approaches. Our source code is available at https://github.com/Li-ZK/CLDA-2022.**

***Index Terms*— Classification, confident learning (CL), domain adaptation, hyperspectral image (HSI).**

Manuscript received January 14, 2022; revised March 6, 2022; accepted April 6, 2022. Date of publication April 12, 2022; date of current version April 29, 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 62171295, Grant 61971164, Grant 61922013, and Grant 61771437. *(Corresponding author: Zhaokui Li.)*

Zhuoqun Fang is with the College of Artificial Intelligence, Shenyang Aerospace University, Shenyang 110136, China, and also with the Shenyang Institute of Computer Technology, Chinese Academy of Sciences, Shenyang 110168, China (e-mail: fangzhuoqun@sau.edu.cn).

Yuexin Yang and Zhaokui Li are with the School of Computer Science, Shenyang Aerospace University, Shenyang 110136, China (e-mail: yang1934301664@163.com; lzk@sau.edu.cn).

Wei Li is with the School of Information and Electronics, Beijing Institute of Technology, Beijing 100081, China (e-mail: liwei089@ieee.org).

Yushi Chen is with the School of Electronics and Information Engineering, Harbin Institute of Technology, Harbin 150001, China (e-mail: chenyushi@hit.edu.cn).

Li Ma is with the School of Mechanical Engineering and Electronic Information, China University of Geosciences, Wuhan 430074, China (e-mail: mali@cug.edu.cn).

Qian Du is with the Department of Electrical and Computer Engineering, Mississippi State University, Starkville, MS 39762 USA (e-mail: du@ece.msstate.edu).

Digital Object Identifier 10.1109/TGRS.2022.3166817

I. Introduction

HYPERSPECTRAL image (HSI) is an important type of data in remote sensing, which is widely used in applications related to earth observation, such as smart agriculture [1], [2] and environmental monitoring [3], [4]. The HSIs contain hundreds of spectral bands [5], [6] and, therefore, provide abundant information for accurate classification [7], [8]. However, labeling the remote sensing data is very expensive and time-consuming [9]. The lack of labeled samples poses a considerable challenge for HSI classification [10]. Moreover, spectral shift [11] exists between HSIs acquired at different times, or in different locations. Therefore, traditional classification models trained on one image cannot achieve satisfactory performance on the other with the spectral shift. According to the aforementioned facts, two natural and reasonable ideas arise, one of which is to provide high-quantity and high-quality data during the model training phase [12], [13], and the other is to reduce the distribution difference between two data [14]–[16].

In recent years, a large number of approaches have been proposed, among which domain adaptation is the most popular one [13], [17]. In the domain adaptation situation, the data with sufficient labels are referred to as the source domain and the data to be predicted with few or without labels are named as the target domain. The objective is to transfer knowledge from a source domain to improve the classification of a target domain. Domain adaptation approaches, aligning the marginal distribution, conditional distribution, or both between the two domains, can be mainly divided into three categories: instance-based, classifier-based, and feature-based approaches.

To partially tackle this problem, the instance-based approaches have attempted to decrease the interdomain discrepancy by reweighting the instances [18]. Schölkopf *et al.* [19] assigned weights to the source-domain instances, which were estimated by matching the means between the source data and the target data in a reproducing kernel Hilbert space. Sugiyama *et al.* [20] proposed an approach minimizing the Kullback–Leibler divergence with respect to the importance weights and incorporating a built-in model selection procedure. In the work by Yao and Doretto [21], both instances in the source and target domains were reweighted, and the weights were updated iteratively by a group of weak classifiers. However, these approaches are based on the assumption that domains differ only in marginal distributions, which cannot be met in most cases. In this article, we would construct an instance selection strategy that can reduce the difference in distributions between domains.

Another possible way to address the problem was the classifier-based approaches [22], which incorporated adaptation in the inference procedure. Mansour and Schain [23] trained a robust classifier by keeping the loss change small when removing part of the training samples, thus enabling the classifier to generalize well to the target domain. Zhong and Zhang [24] repeatedly trained a support vector machine (SVM) using the source data and the target data with pseudo-labels [25] assigned by the trained classifier. In the work by Wang *et al.* [26], the target data was self-labeled by a classifier to match the domains via a linear transformation, which in turn retrained the classifier and reassigned target labels until convergence. It can be seen that how the pseudo-labels are generated is important. Pseudo-labels for deep learning in computer vision was originally adopted by Zhang *et al.* [27], who discovered pseudo-labels using a hashing method to convert unsupervised models into supervised ones. Almost simultaneously in the society of geoscience and remote sensing, Wu and Prasad [28] incorporated pseudo-labels by a clustering method in semisupervised deep learning. However, it is a challenge to obtain high-quality target domain labels, and the quality of the pseudo-labeling is difficult to evaluate as there may have no label in the target domain.

Recently, feature-based approaches have been the most popular among the three types of domain adaptation methods, and deep learning methods fell into this category. Feature-based algorithms extract domain-invariant feature representations to achieve domain adaptation through feature matching [29]–[32] or adversarial learning [33]–[37]. Feature matching approaches reduce the difference between feature distributions by minimizing some domain dissimilarity metrics. In the work by Long *et al.* [29], a deep adaptation network (DAN) was presented to adapt features from convolutional layers by minimizing a multiple kernel maximum mean discrepancy (MK-MMD). Zhang *et al.* [38] proposed discriminative cooperative alignment (DCA) to reduce the geometric and statistical shift while preserving discriminant information. Wang *et al.* [39] projected features to an embedding space where a weighted MMD was minimized and the manifold structure was preserved. However, the extraction of discriminative features in these works is heavily dependent on the source domain data [38], [39] or is not considered at all [29], and the target domain data have little to contribute to the discriminative representation during the process of feature matching.

In contrast, adversarial learning approaches did not measure the distance of features between different domains, but rather trained the model with an adversarial learning strategy to produce domain-invariant features. Ganin *et al.* [33] introduced domain-adversarial neural networks (DANN) to promote the emergence of features using a domain classifier in an adversarial way. In the work by Deng *et al.* [4], Euclidean distance-based deep metric model with unsupervised domain adaptation (ED-DMM-UDA) trained a deep embedding model to generate a metric value between features and added it to an adversarial framework to adapt features distributions. Liu *et al.* [40] presented class-wise distribution adaptation (CDA) generating domain-invariant features by adopting a class-wise MMD and class-wise adversarial adaptation, and pseudo-labels of target data were used during model optimization. Unlike previous approaches conducting adversarial learning by constructing a domain discriminator, biclassifier domain adaptation methods [41], [42] utilized two distinct classifiers instead of the domain discriminator. However, adversarial learning will bring about deterioration of feature discriminability when the feature transferability is strengthened [42]. In other words, the lack of classifier determinacy in assigning confident labels on target samples affects feature discriminability.

To evaluate the label quality, a type of approach called confident learning (CL) [43] was investigated, which focused on label quality by characterizing and identifying label errors in datasets. Based on the principle of pruning noisy data, the confident learning methods estimated noise with probabilistic thresholds and ranked examples to train with confidence. Inspired by confident learning, we rank the pseudo-labels of target data according to their confidences to prune the noisy pseudo-labels.

In this article, we propose a novel adversarial domain adaptation framework in combination with confident learning, which can generate discriminative feature representation to meet the tasks of cross-domain classification. The objective of the proposed framework is to train the deep neural networks using high-quality labeled instances from the target domain. The novelty of our framework is that we present a powerful selection strategy for labeling target instances by using confident learning so that the target data can be applied as the source data for the optimization of the cross-entropy loss function. More specifically, pseudo-labels of target instances are recognized as noisy labels in confident learning, and the confidence of each label is evaluated. Once the confidences have been estimated, only instances with high-confidence labels are provided to model training of adversarial domain adaptation, which can preserve the feature discriminability. Compared with other recent work in this field, our proposed method has the following four contributions.

1) In confident learning-based domain adaptation (CLDA), the pseudo-labels of the target domain are regarded as noisy labels. As far as we know, it is the first time to combine domain adaptation and label denoising for cross-domain HSI classification, which can learn a more robust classification model for the target domain.
2) A novel biclassifier domain adaptation framework combined with confidence learning is designed to solve the problem of UDA, which enhances the discriminance of the model to the target domain while aligning the interdomain distribution.
3) A strategy of iteratively alternating between training domain adaptation and confident learning is adopted to give priority to target samples with high-confidence labels in training, which can progressively increase the proportion of high-confidence labels in the target domain and improve the accuracy of classification.
4) For the target instances with high-confidence labels, the cross-entropy loss and entropy regularization loss are

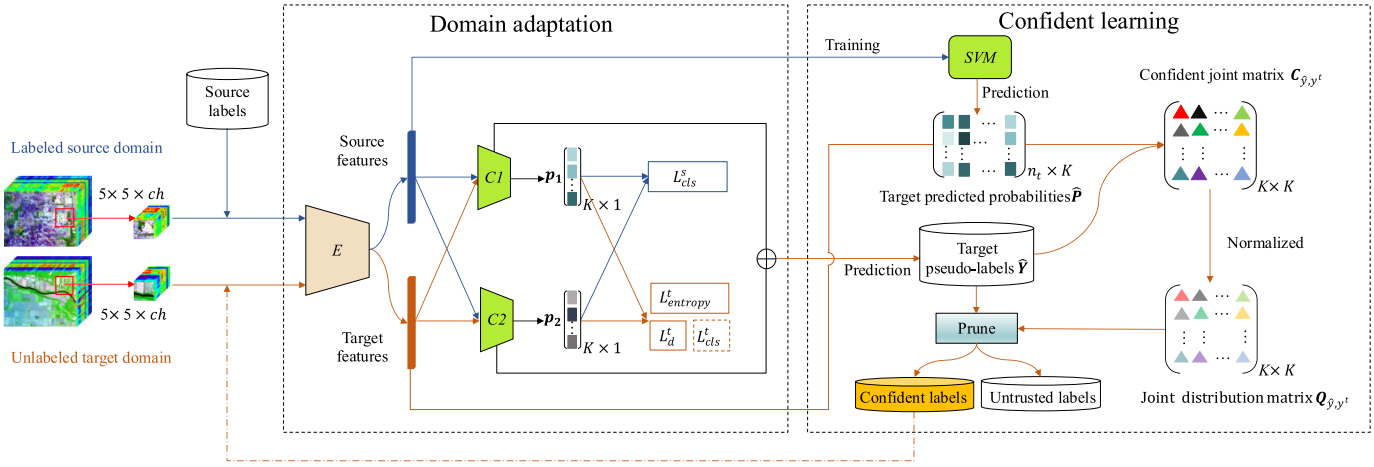

Fig. 1. Framework of the CLDA method.

carried out simultaneously, which improve the discriminability of target domain features.

The experimental results demonstrate that our method can outperform the previous state-of-the-art works concerning overall accuracy (OA), average accuracy (AA), and kappa coefficient (K) evaluated on different datasets.

The remainder of this article is organized as follows. The problem specification and notation are introduced in Section II, including confident learning and UDA. In Section III, the proposed CLDA method and the training strategy are presented in detail. The experimental results are discussed in Section IV, and the conclusion is drawn in Section V.

II. Definition of Problem

UDA [17] aims to use the relevant labeled data in the source domain to learn classifiers for the target domain where no label information is available. In this case, the dataset contains labeled source domain data and unlabeled target domain data. Specifically, the source domain has n_s samples that can be represented as $\{\boldsymbol{X_s}, \boldsymbol{Y_s}\} = \{\boldsymbol{x}_i^s, y_i^s\}_{i=1}^{n_s}$, where y_i^s is the corresponding label of x_i^s. Similarly, the target domain with n_t instances can be expressed as $\{\boldsymbol{X_t}, \boldsymbol{Y_t}\} = \{\boldsymbol{x}_i^t, y_i^t\}_{i=1}^{n_t}$. It is worth noting that the target domain samples do not have labels $\boldsymbol{Y_t}$. Notation used is summarized in Table I.

In this article, the biclassifier adversarial domain adaptation and confident learning are employed. In domain adaptation, the biclassifiers (*C1* and *C2*) output two probability vectors $\boldsymbol{p_1}$ and $\boldsymbol{p_2}$ ($\boldsymbol{p_1}$, $\boldsymbol{p_2} \in \mathbb{R}^{K\times 1}$) to optimize the loss function and update the neural networks, where K is the number of categories. Feature fusion classification takes the final layer features (before softmax) of the biclassifiers as input and selects the maximum element of the sum of the two features as the pseudo-label [44]. As a result, pseudo-labels $\widehat{\boldsymbol{Y}} = \{\hat{y}_i\}_{i=1}^{n_t}$ of all target instances are obtained. Confident learning is used to prune noisy data on labeled datasets and preserve the clean data for model training. In confident learning, an SVM classifier is adopted to estimate the predicted probabilities $\widehat{\boldsymbol{P}} = \{\hat{p}_{ij}\}_{i,j=1}^{i=n_t, j=K}$. Therefore, each target sample has a pseudo-label and a predicted probability. In this way, the confident joint matrix $\boldsymbol{C}_{\hat{y},y^t} \in \mathbb{R}^{K\times K}$ is achieved, and normalizing $\boldsymbol{C}_{\hat{y},y^t}$ can obtain the joint distribution matrix $\boldsymbol{Q}_{\hat{y},y^t} \in \mathbb{R}^{K\times K}$. The pseudo-labels with high-confidence are termed as confident labels $\widehat{\boldsymbol{Y}}_{\boldsymbol{c}} = \{\hat{y}_{c_i}\}_{i=1}^{n_c}$, where n_c is the number of confident labels, $\boldsymbol{X_{tc}} = \{\boldsymbol{x}_i^{tc}\}_{i=1}^{n_c}$ is the confident samples of the target domain, and the class weights $w_{\text{confident\_class}} \in \mathbb{R}^K$ are used to weight instances when updating the neural networks.

III. Proposed CLDA Method

The CLDA method, as illustrated in Fig. 1, consists of two main parts: domain adaptation and confident learning. Domain adaptation is employed to align the source and target domains. Confident learning attempts to evaluate pseudo-labels, thus selecting labels with high confidence. These labels and their corresponding samples are regarded as a new training set of target domain, which in turn is fed into the domain adaptation part. By iteratively alternating between training domain adaptation and confident learning, the CLDA method can progressively improve the classification accuracy.

Specifically, in the left part of Fig. 1, from the source domain (indicated by blue lines) and the target domain (indicated by orange lines), two HSI patches $I \in \mathbb{R}^{5\times 5\times ch}$, where ch and 5×5 denote the spectral bands and the spatial dimensions, are separately input into the feature extractor E. Then, the features extracted from both the source and target domains (indicated by black lines) are fed into the two classifiers *C1* and *C2*, and these two classifiers produce $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$, the predicted probabilities using the features. Thus, $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ can used to input into the loss function so that update E, *C1*, and *C2*. To predict pseudo-labels $\widehat{\boldsymbol{Y}}$ of the target instances, we take the last layer features of *C1* and *C2* (without softmax) and fuse these two features by summing them up. After that, the corresponding category of the maximum element of the fusion feature is selected as the pseudo-label. In the right part of Fig. 1, confident learning attempts to evaluate pseudo-labels $\widehat{\boldsymbol{Y}}$ and select the confident labels $\widehat{\boldsymbol{Y}}_{\boldsymbol{c}}$ from them. An SVM classifier is introduced to obtain the predicted probabilities $\widehat{\boldsymbol{P}}$ of the target domain samples. Based on this, the upper right matrix $\boldsymbol{C}_{\hat{y},y^t}$ is obtained by $\widehat{\boldsymbol{P}}$ and $\widehat{\boldsymbol{Y}}$. It is a counting matrix that counts the number of samples with a sufficiently large probability belonging to a class in each category. Then,

TABLE I
NOTATION USED IN CLDA

Notation	Definition	comment
ch	The HSI bands of the data set	
K	The number of the class, and $[K]$ denotes the set of K class labels	
n_s	The number of source domain data set	
n_t	The number of target domain data set	
n_c	The number of confident labels	
m	The discrete random variable $m \in [K]$	
n	The discrete random variable $n \in [K]$	
$\boldsymbol{X_s}$	The source domain data	
$\boldsymbol{Y_s}$	The source domain labels	
$\boldsymbol{x_i^s}$	The i^{th} pixel of source domain HSI with $ch-$bands	Domain adaptation
y_i^s	The source domain label corresponding to x_i^s	
$\boldsymbol{X_t}$	The target domain data	
$\boldsymbol{Y_t}$	The target domain unknown, true labels	
$\boldsymbol{x_i^t}$	The i^{th} target data of target domain	
y_i^t	The i^{th} unknown, true label corresponding to x_i^t	
$\boldsymbol{A}$	The prediction relevance matrix, $A \in \mathbb{R}^{K\times K}$	
A_{mn}	The product of predicted probabilities that *C1* classifies the sample into category m and *C2* classifies the sample into category n	
$\oplus$	Feature fusion classification (in Fig. 1): sum the output features (after the last layer and before softmax) of the bi-classifiers (*C1* and *C2*) and select the category label corresponding to the maximum element of the feature sum as the classification result	
$\boldsymbol{X_{tc}}$	The confident samples of the target domain	
$\widehat{\boldsymbol{Y}}_{\boldsymbol{c}}$	The confident labels of target domain (pseudo-labels with high-confidence)	
$\boldsymbol{x_i^{tc}}$	The i^{th}data of confident samples	
$\hat{y}_i^c$	The i^{th} confident label of target domain	
$\boldsymbol{p_1}$	The softmax outputs of classifier *C1*, $\boldsymbol{p_1} \in \mathbb{R}^{K\times 1}$	
$\boldsymbol{p_2}$	The softmax outputs of classifier *C2*, $\boldsymbol{p_2} \in \mathbb{R}^{K\times 1}$	
$\widehat{\boldsymbol{Y}}$	The target pseudo-labels	
$\hat{y}_i$	The target pseudo-label corresponding to $\boldsymbol{x_i^t}$	
$\widehat{\boldsymbol{P}}$	The target predicted probabilities by the SVM classifier, $\widehat{\boldsymbol{P}} \in \mathbb{R}^{K\times 1}$	
$\hat{p}_{ij}$	The target predicted probabilities of i^{th} sample belonging to j^{th} class	Confident learning
$\hat{p}(\hat{y}=j;\boldsymbol{x_t})$	Predicted probability of label $\hat{y}=j$ for data $\boldsymbol{x_t}$	
$\boldsymbol{X_{t(\hat{y}=i)}}$	Target data in $\boldsymbol{X_t}$ with pseudo-label i	
$\boldsymbol{X_{t(\hat{y}=i,y^t=j)}}$	Target data in $\boldsymbol{X_t}$ with pseudo-label i and true label j	
$\widehat{\boldsymbol{X}}_{\boldsymbol{t}(\hat{y}=i,y^t=j)}$	Estimate of target data in $\boldsymbol{X_t}$ with pseudo-label i and true label j	
$\boldsymbol{C}_{\hat{y},y^t}$	The confident joint matrix, $\boldsymbol{C}_{\hat{y},y^t} \in \mathbb{R}^{K\times K}$	
$\boldsymbol{Q}_{\hat{y},y^t}$	The joint distribution matrix $\boldsymbol{Q}_{\hat{y},y^t} \in \mathbb{R}^{K\times K}$, a normalized of $\boldsymbol{C}_{y^*,y^t}$	
$\widehat{\boldsymbol{Q}}_{\hat{y},y^t}$	Estimate of $\boldsymbol{Q}_{\hat{y},y^t}$	
t_j	The average for class j, used in $\boldsymbol{C}_{\hat{y},y^t}$ as threshold	
$w_{confident\_class}$	The class weights of clean samples, $w_{confident\_class} \in \mathbb{R}^K$	

$\boldsymbol{C}_{\hat{y},y^t}$ is normalized to obtain $\boldsymbol{Q}_{\hat{y},y^t}$, which is used to evaluate pseudo-labels. Through the pruning operation, pseudo-labels are divided into confident and untrusted labels. In turn, the target samples with the confident labels are selected as training data of the domain adaptation part.

A. Domain Adaptation

To solve the domain shift and mitigate side effects of the deterioration of feature discriminability, a biclassifier domain adaptation method is adopted to align the source and target domains. It is optimized by adversarial learning [41] between the feature extractor E and the biclassifiers *C1* and *C2*. In other words, the feature extractor is updated by minimizing the classifier discrepancy while biclassifiers are optimized by maximizing that discrepancy. Inspired by the biclassifier determinacy maximization [41], we use classifier determinacy disparity (CDD) to measure the classifier discrepancy. Through this adversarial learning, the biclassifier domain adaptation part can align distributions of source and target domains. The loss function of the CLDA method is defined as

$$L\left(\boldsymbol{X_s}, \boldsymbol{Y_s}, \boldsymbol{X_t}, \widehat{\boldsymbol{Y}}_{\boldsymbol{c}}, \boldsymbol{X_{tc}}\right) = L_{\text{cls}}^{s}(\boldsymbol{X_s}, \boldsymbol{Y_s}) + \gamma\, L_d^t(\boldsymbol{X_t}) + \alpha L_{\text{cls}}^t \times\left(\boldsymbol{X_{tc}}, \widehat{\boldsymbol{Y}}_{\boldsymbol{c}}\right) + \beta L_{\text{entropy}}^t(\boldsymbol{X_{tc}}) \quad (1)$$

where the first term represents the classification loss on the labeled source domain samples, the second term denotes the CDD loss on the unlabeled target domain samples, the third term expresses the classification loss on the target domain samples with confident labels, and the fourth term denotes the entropy regularization [40], [45] loss on the target domain samples. The notations γ, α, and β are the tradeoff hyperparameters. In the CLDA network, the feature extractor E and two classifiers (*C1* and *C2*) are trained by the following four losses.

1) Source Classification Loss: This is a cross-entropy loss. Labeled source domain samples are fed into feature extractor E to extract the features. The source features are used to train the classifiers *C1* and *C2*. Update the *E*, *C1*, and *C2* by minimizing the source classification loss

$$L^s_{\text{cls}}(\boldsymbol{X_s}, \boldsymbol{Y_s}) = \frac{1}{2n_s}\sum_{i=1}^{n_s}\sum_{j=1}^{2} L_{\text{cross-entropy}}\left(C_j\left(E\left(\boldsymbol{x}_i^{\boldsymbol{s}}\right)\right), y_i^s\right) \quad (2)$$

where $C(\cdot)$ is the output of the two classifiers and $C_j(E(x_i^s))$ represents the prediction value.

2) CDD Loss: In cooperation with biclassifiers, this loss formulates classifier discrepancy as the class relevance of distinct target predictions; meanwhile, implicitly imposed a constraint on the target feature discriminability. Specifically, two probability matrices $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ are obtained by two classifiers (*C1* and *C2*). Froze the parameters of E and update *C1* and *C2* by maximizing the discrepancy L_d^t between the two classifiers. It is defined as follows:

$$L_d^t(\boldsymbol{X_t}) = \frac{1}{n_t}\sum_{i=1}^{n_t} L_{\text{cdd}}\left(\boldsymbol{x}_i^{\boldsymbol{t}}\right) \quad (3)$$

$$L_{\text{cdd}} = \sum_{m,n=1}^{K} A_{mn} - \sum_{m=1}^{K} A_{mm} = \sum_{m\neq n}^{K} A_{mn} \quad (4)$$

where L_{cdd} is a CDD loss to measure the classifier discrepancy, $m, n \in \{1, 2, 3, \ldots, K\}$, and $\boldsymbol{A}$ is the prediction relevance matrix used to represent the discrepancy between the classifiers. It can be defined as $\boldsymbol{A} = \boldsymbol{p}_1\boldsymbol{p}_2^T$, where $\boldsymbol{p}_1, \boldsymbol{p}_2 \in \mathbb{R}^{K\times 1}$ are the softmax probabilities of the outputs of the two classifiers and satisfy $\sum_{k=1}^{K} p_i^k = 1, p_i^k \geq 0, \forall k = \{1, \ldots\ldots, K\}, i = \{1, 2\}$. So, $A_{mn} = p_1^m p_2^n$ is the product of probabilities that *C1* classifies the sample into category m and *C2* classifies the sample into category n. By adversarially optimizing the CDD loss, the classifier's determinacy and prediction diversity are simultaneously enhanced. In this way, the interdomain discrepancy is also decreased.

3) Target Classification Loss: When the target domain has confident labels, they are used to further improve the discriminability of the model. Let L^t_{cls} be the classification loss of target domain with confident labels, which can be expressed as

$$L^t_{\text{cls}}\left(\boldsymbol{X_{tc}}, \widehat{\boldsymbol{Y}}_{\boldsymbol{c}}\right) = \frac{1}{2n_{tc}}\sum_{i=1}^{n_{tc}}\sum_{j=1}^{2} L_{\text{cross-entropy}}\left(C_j\left(E\left(\boldsymbol{x}_i^{tc}\right)\right), \hat{y}_{c_i}\right) \quad (5)$$

where $C_j(E(x_i^{tc})$ represents the prediction value. The weight of the cross-entropy loss is set to be $\omega_{\text{confident\_class}}$.

4) Entropy Regularization: For the target domain, the labels of the confident samples selected by confident learning may be inaccurate; therefore, the low-density separation between classes is guaranteed by the constraint on the predicted probabilities of the target domain samples [43]. As for the probabilistic prediction results in the target domain, it is expected that the probabilistic prediction output has a peak distribution rather than a smooth distribution

$$L^t_{\text{entropy}}\left(C_j\left(E\left(x_i^t\right)\right), \hat{y}_{c_i}\right) = -\frac{1}{N}\sum_{x_t^i\in X_t}\sum_{k=1}^{K} p_i^k \log p_i^k \quad (6)$$

where p_i^k is the mean of probabilities predicted by the two classifiers in the target domain. The entropy regularization loss also ensures discriminability of the features by utilizing target instances after data cleaning by the confident learning part.

B. Confident Learning

Confident learning [43] takes two inputs: the noisy pseudo-labels $\widehat{\boldsymbol{Y}}$ and the predicted probabilities $\widehat{\boldsymbol{P}}$. On the one hand, all the target domain samples are classified by *C1* and *C2* to obtain pseudo-labels $\widehat{\boldsymbol{Y}}$, and $\widehat{\boldsymbol{Y}}$ is recognized as the noisy label input of confident learning. Specifically, the sum of the final layer feature vector of *C1* and *C2* is calculated and the category concerning the maximum probability is chosen as the pseudo-label of that sample. On the other hand, an SVM classifier is adopted to estimate predicted probabilities $\widehat{\boldsymbol{P}}$ of target samples so that the classifications according to $\widehat{\boldsymbol{P}}$ are different with $\widehat{\boldsymbol{Y}}$. To this end, we train the SVM classifier using source domain data. In other words, the source features extracted by E are fed into the SVM, and the related labels are only used to update the parameters of the SVM classifier rather than that of the feature extractor E during this training phase. As a result, each target sample $\boldsymbol{x}_i^t$ can evaluate its confidence belonging to a category j, which can be denoted by $\hat{p}_{ij}$ or $\hat{p}(\hat{y} = j; \boldsymbol{x_t})$. Since the target domain samples have no ground truth labels $\boldsymbol{Y_t}$, here, the results predicted by SVM are regarded as credible and close to the true labels.

The process of confident learning can be divided into two steps: 1) estimate the joint distribution matrix $\boldsymbol{Q}_{\hat{y},y^t}$ of pseudo-labels and true labels (unknown, estimated by $\widehat{\boldsymbol{P}}$) to evaluate the confidence of the pseudo-labels and 2) find and prune the untrusted labels based on their confidence.

First, a confident joint matrix $\boldsymbol{C}_{\hat{y},y^t}$ is obtained from the $\widehat{\boldsymbol{Y}}$ and $\widehat{\boldsymbol{P}}$ to partition and count label errors. Specifically, for the target domain, $\boldsymbol{X_t}_{(\hat{y}=i,y^t=j)}$ represents samples $\boldsymbol{x_t}$ with pseudo-label i that actually have true label j. Here, $\widehat{\boldsymbol{X}}_{\boldsymbol{t}(\hat{y}=i,y^t=j)}$ is used to estimate $\boldsymbol{X_t}_{(\hat{y}=i,y^t=j)}$. It means the set of samples $\boldsymbol{x}_t$ labeled $\hat{y} = i$ with enough predicted probabilities $\hat{p}(\hat{y} = j; \boldsymbol{x_t})$ for class $y^t = j$, determined by a threshold t_j. Here, t_j is a per-class threshold equivalent to the mean of the probabilities of each class. The automatic thresholding method makes confident learning robust to class-imbalance and heterogeneous class probability distributions. Thus, the confident joint matrix can be defined as follows:

$$\boldsymbol{C}_{\hat{y},y^t}[i][j] := |\widehat{\boldsymbol{X}}_{\boldsymbol{t}(\hat{y}=i,y^t=j)}| \quad \text{where}$$
$$\hat{\boldsymbol{X}}_{\boldsymbol{t}(\hat{y}=i,y^t=j)} := \left\{\boldsymbol{x_t}\in \boldsymbol{X}_{t(\hat{y}=i)} : \hat{p}(\hat{y} = j; \boldsymbol{x_t}) \geq t_j\right.$$

$$j = \underset{l\in\{1,\ldots,k\}:\hat{p}(\hat{y}=l;\boldsymbol{x_t})\geq t_l}{\operatorname{argmax}} \hat{p}(\hat{y}=l;\boldsymbol{x_t})\Big\} \quad (7)$$

$$t_j = \frac{1}{|\boldsymbol{X_{t(\hat{y}=j)}}|}\sum_{\boldsymbol{x_t}\in\boldsymbol{X_{t(\hat{y}=j)}}} \hat{p}(\hat{y}=j;\boldsymbol{x_t}). \quad (8)$$

Normalize $\boldsymbol{C}_{\hat{y},y^t}$ to obtain the estimate of joint distribution matrix $\boldsymbol{Q}_{\hat{y},y^t}$

$$\widehat{\boldsymbol{Q}}_{\hat{y}=i,y^t=j} = \frac{\frac{\boldsymbol{C}_{\hat{y}=i,y^t=j}}{\sum_{j\in\{1,\ldots,K\}}\boldsymbol{C}_{\hat{y}=i,y^t=j}}\cdot|\boldsymbol{X_t}(\hat{y}=i)|}{\sum_{i,j\in\{1,\ldots,K\}}\left(\frac{\boldsymbol{C}_{\hat{y}=i,y^t=j}}{\sum_{j\in\{1,\ldots,K\}}\boldsymbol{C}_{\hat{y}=i,y^t=j}}\cdot|\boldsymbol{X_t}(\hat{y}=i)|\right)}. \quad (9)$$

It is the joint distribution of pseudo-labels and true labels.

Second, following estimation of $\boldsymbol{Q}_{\hat{y},y^t}$, the prune by class (PBC) method [43] is adopted to clean the target pseudo-labels. For each class $i\in\{1,\ldots,K\}$, we first sort the samples (with the pseudo-label i) according to their self-confidence $\hat{p}(\hat{y}=i;\boldsymbol{x_t})$, and then select the $n_t\cdot\sum_{j\in\{1,\ldots,K\}:j\neq i}(\widehat{\boldsymbol{Q}}_{\hat{y}=i,y^t=j}[i])$ examples with the lowest self-confidence $\hat{p}(\hat{y}=i;\boldsymbol{x_t})$ as the untrusted labels and prune them. In this way, the confident labels $\widehat{Y}_c\in\widehat{Y}$ are obtained. Since the untrusted labels are filtered out class-wisely, the number of confident labels between categories may be different. To balance the class distribution, we calculate the class weights $w_{\text{confident\_class}}$

$$w_{\text{confident\_class}} = \frac{\widehat{\boldsymbol{Q}}_{\hat{y}}[i]}{\widehat{\boldsymbol{Q}}_{\hat{y},y^t}[i][i]}. \quad (10)$$

As a result, confident labels and samples $\{\boldsymbol{X_{tc}},\widehat{\boldsymbol{Y}}_{\boldsymbol{c}}\}$ are fed into the feature extractor E to replace all target domain samples X_t, and the class weights $w_{confident\_class}$ are used to reweight the target supervised loss function in each class. In this way, these target samples and their confident labels $\{\boldsymbol{X_{tc}},\widehat{\boldsymbol{Y}}_{\boldsymbol{c}}\}$ can be used to update the neural networks of the domain adaptation part in the same way as the labeled source data.

C. Implementation Details

To make full use of confident target domain samples and improve classification performance in the target domain, the training process of the CLDA method includes two parts, domain adaptation and confident learning. In the early stages of training, there are no confident labels in the target domain. Therefore, we first train the feature extractor E and biclassifiers using labeled source data $\boldsymbol{X_s},\boldsymbol{Y_s}$ and unlabeled target data $\boldsymbol{X_t}$. After *TRAIN_NUM* epochs, biclassifiers are able to provide pseudo-labels to target samples. Through confident learning, confident labels and their related samples $\{\boldsymbol{X_{tc}},\hat{Y}_{\boldsymbol{c}}\}$, instead of the whole target data $\boldsymbol{X_t}$, are selected to add to model training. Once achieving confident labels of target domain, we can iteratively alternate between training domain adaptation and confident learning. The total training procedure is detailed in Algorithm 1.

On the one hand, the adversarial training process of domain adaptation is divided into three steps. First, using labeled source domain samples to update the feature extractor (E) and two classifiers (*C1* and *C2*), which improves the discriminability of the network to the source domain samples.

Algorithm 1 The Algorithm of CLDA for UDA

Input: Source domain samples $\boldsymbol{X_s},\boldsymbol{Y_s}$, Target domain samples $\boldsymbol{X_t}$, trade-off parameter α,β and γ, the total epoch number *NUM*, the epoch number to start confident learning *TRAIN_NUM*.

Output: The parameters $\theta_e,\theta_{c1},\theta_{c2}$ of feature extractor E and two classifiers *C1*, *C2*.

begin

Randomly initialize $\theta_e,\theta_{c1},\theta_{c2}$, $\alpha=\beta=0$.

FOR epoch in *{1, ..., NUM}* **do**

Domain adaptation

If epoch $\geq TRAIN\_\_NUM$, set the values of α,β

Update $\theta_e,\theta_{c1},\theta_{c2}$ under Eq. (11).

Update θ_{c1},θ_{c2} through Eq. (12),

Update θ_e through Eq. (13),

Confident learning

If epoch % *TRAIN_NUM* == 0

Obtain the target pseudo labels $\widehat{Y}$ and target predicted probabilities $\widehat{P}$.

obtain the confident joint matrix $\boldsymbol{C}_{\hat{y},y^t}$ according to Eq. (1).

obtain the joint distribution matrix $\boldsymbol{Q}_{\hat{y},y^t}$ according to Eq. (3)

Find the untrusted labels.

Prune the untrusted labels to obtain the confident labels.

Use target domain samples with confident labels and source domain samples to update E, *C1* and *C2*.

END FOR

When the target domain has confident labels, add the cross-entropy loss and entropy regularization loss to make the network more discriminant to target domain samples. It can be expressed as

$$\min_{\theta_e,\theta_{c1},\theta_{c2}} L^s_{\text{cls}}(\boldsymbol{X_s},\boldsymbol{Y_s}) + \alpha L^t_{\text{cls}}\left(\boldsymbol{X_{tc}},\widehat{\boldsymbol{Y}}_{\boldsymbol{c}}\right) + \beta L^t_{\text{entropy}}(\boldsymbol{X_{tc}}) \quad (11)$$

where $\theta_e,\theta_{c1},\theta_{c2}$ are the parameters of feature extractor and two classifiers, and $\alpha=\beta=0$ before the confident learning begins. Second, using target domain samples to update the classifiers and preserve the accuracy on source domain samples, the formula becomes

$$\min_{\theta_{c1},\theta_{c2}} L^s_{\text{cls}}(\boldsymbol{X_s},\boldsymbol{Y_s}) - L^t_d(\boldsymbol{X_{tc}}). \quad (12)$$

Third, using unlabeled target domain samples to update the feature extractor, which can be expressed as follows:

$$\min_{\theta_e} L^t_d(\boldsymbol{X_{tc}}). \quad (13)$$

It is worth noting that feature fusion classification results $\widehat{\boldsymbol{Y}}$ do not participate in gradient back propagation. On the other hand, confident learning is used to find more accurate target domain samples with confident labels to further supervise the training of the classifier and improve the transferability of the model. All target domain samples are predicted to obtain pseudo-labels $\widehat{\boldsymbol{Y}}$ by final layer features of *C1* and *C2*. Then, the SVM classifier is trained by source domain features

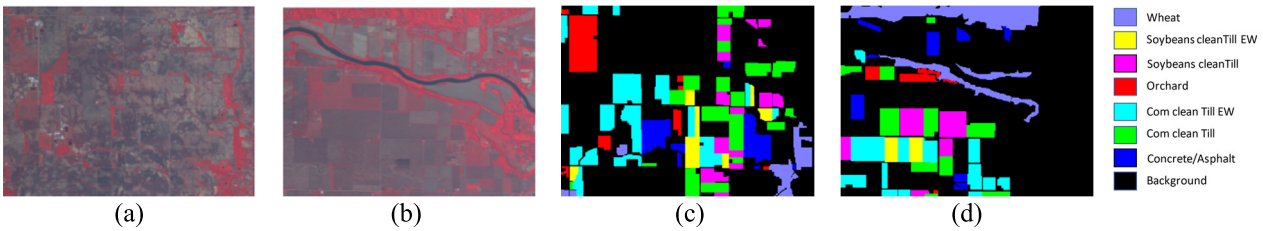

Fig. 2. Indiana image: (a) source false-color image; (b) target false-color image; (c) source ground-truth map; and (d) target ground-truth map.

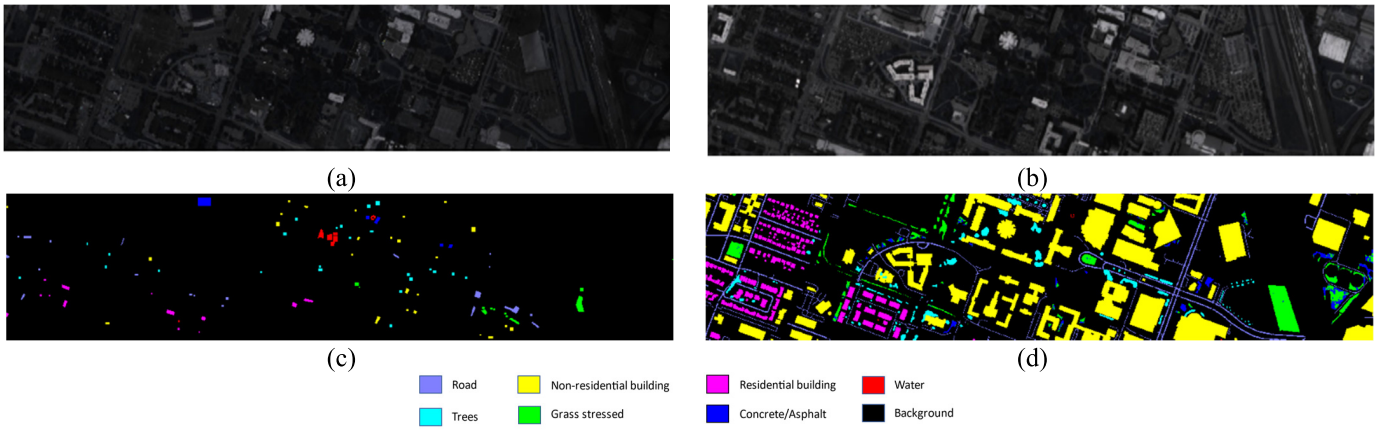

Fig. 3. Houston image: (a) Houston 2013 false-color image; (b) Houston 2018 false-color image; (c) Houston 2013 ground-truth map; and (d) Houston 2018 ground-truth map.

and used to predict target samples to obtain the prediction probabilities $\widehat{\boldsymbol{P}}$. Counting the sample number using $\widehat{\boldsymbol{Y}}$ and $\widehat{\boldsymbol{P}}$ in each category to obtain the matrix $\boldsymbol{C}_{\hat{y},y^t}$. Normalize $\boldsymbol{C}_{\hat{y},y^t}$ to obtain the joint distribution matrix $\boldsymbol{Q}_{\hat{y},y^t}$. $\boldsymbol{Q}_{\hat{y},y^t}$ can calculate the class weight and filter out the untrusted labels to obtain the confident labels for the next epoch of training.

IV. Experiments and Discussion

A. Data Description

To evaluate the effectiveness of the proposed method, four publicly available HSI datasets are used, including Indiana, Houston, Pavia, and Shanghai–Hangzhou.

1) Indiana: The Indiana dataset was gathered by AVIRIS from Northwest Tippecanoe Country, IN, USA, in 1992. The image spatial resolution is 20 m. As in [46], two nonoverlapping subsets are divided into source and target domains. Both the domains have 400×300 pixels and 220 bands. They share seven vegetation classes. The detailed information of the samples is shown in Table II. Fig. 2 shows their false-color images and ground-truth maps.

2) Houston: The Houston datasets were captured at the University of Houston campus, Houston, TX, USA, with different sensors and years, including Houston 2013 (source domain) and Houston 2018 (target domain). Houston 2013 contains 349×1905 pixels and has 144 spectral bands. Its spatial resolution is 2.5 m. The Houston 2018 contains 209×955 pixels and has 48 spectral bands. Its spatial resolution is 1 m. They have the same wavelength range 380–1050 nm. Specifically, on the Houston 2013 dataset, we chose 209×955 overlapping areas and 48 spectral bands, which are corresponding to Houston 2018. Both of them contain seven classes, and the detailed information of the samples is shown in Table III. Fig. 3 shows their false-color images and ground-truth maps.

TABLE II
Land Cover Classes and the Numbers of Samples in the Indiana Dataset

Class		Number of samples	
	Name	Source scene	Target scene
C1	Concrete/Asphalt	4867	2942
C2	Corn cleanTill	9822	6029
C3	Corn cleanTill EW	11414	7999
C4	Orchard	5106	1562
C5	Soybeans cleanTill	4731	4792
C6	Soybeans cleanTill EW	2996	1638
C7	Wheat	3223	10739
Total		42159	35701

TABLE III
Land Cover Classes and the Numbers of Samples in the Houston Dataset

Class		Number of samples	
	Name	Houston 2013 (Source)	Houston 2018 (Target)
C1	Grass healthy	345	1353
C2	Grass stressed	365	4888
C3	Trees	365	2766
C4	Water	285	22
C5	Residential buildings	319	5347
C6	Non-residential buildings	408	32459
C7	Road	443	6365
Total		2530	53200

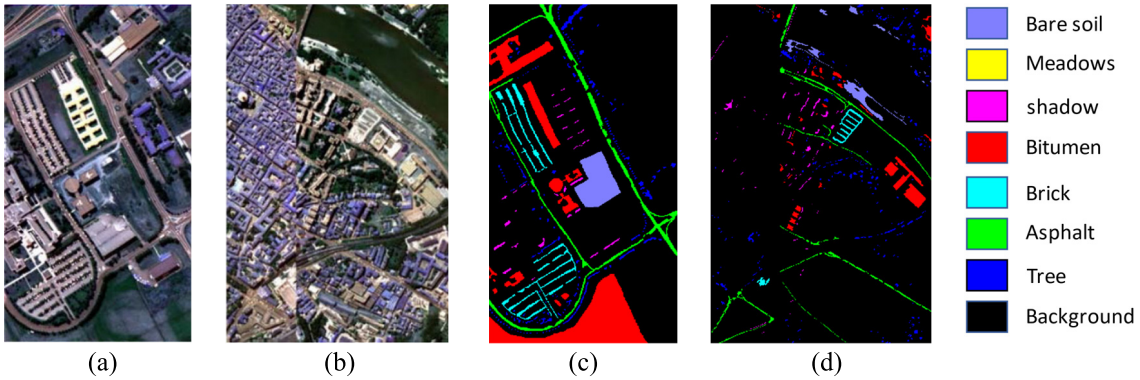

Fig. 4. Pavia image: (a) UP false-color image; (b) PC false-color image; (c) UP ground-truth map; and (d) PC ground-truth map.

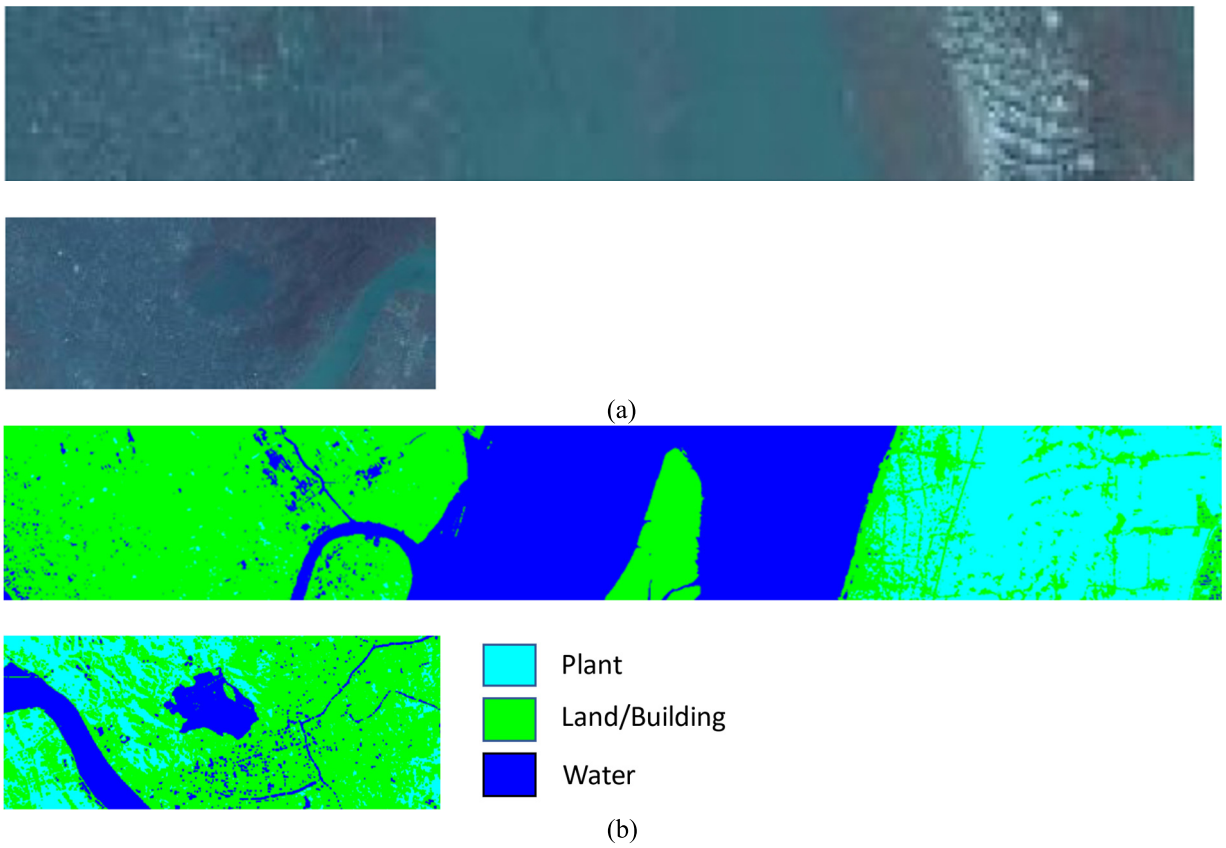

Fig. 5. Shanghai–Hangzhou image: (Top) Shanghai and (Bottom) Hangzhou. (a) False-color image; (b) Ground-truth map.

3) Pavia: The Pavia datasets include the University of Pavia (UP, source domain) and Pavia Center (PC, target domain). Both were collected by ROSIS sensors in Pavia in northern Italy. The UP has 610×610 pixels and 103 bands. The PC has 1096×1096 pixels and 102 bands. Specifically, because some samples in these two domains have no information, they can be discarded before analysis. Therefore, the UP contains 610×315 pixels and the PC contains 1096×715 pixels. For the UP, the last band was removed to ensure the same number of spectral bands as PC. In particular, after removing the last band of the UP, the spectral wavelength range of both datasets is 0.430–0.834. They all have seven same categories, and the detailed information of the samples is shown in Table IV. Fig. 4 shows their false-color images and ground-truth maps.

4) Shanghai–Hangzhou: Shanghai (source domain) and Hangzhou (target domain) datasets were obtained by the EO-1 Hyperion hyperspectral sensor, which leaves 198 bands and removes the bad band. The Shanghai has 1600×230 pixels and Hangzhou has 590×230 pixels. Their shared classes are 3, including water, ground/buildings, and plants. The detailed information of the samples is shown in Table V. Fig. 5 shows their false-color images and ground-truth maps.

TABLE IV
LAND COVER CLASSES AND THE NUMBERS OF SAMPLES IN THE PAVIA DATASET

Class		Number of samples	
	Name	UP (Source)	PC (Target)
C1	Tree	3064	7598
C2	Asphalt	6631	9248
C3	Brick	3682	2685
C4	Bitumen	1330	7287
C5	shadow	947	2863
C6	Meadow	18649	3090
C7	Bare soil	5029	6584
Total		39332	39355

TABLE V
LAND COVER CLASSES AND THE NUMBERS OF SAMPLES IN THE SHANGHAI–HANGZHOU DATASET

Class		Number of samples	
	Name	Shanghai (Source)	Hangzhou (Target)
C1	Water	123123	18043
C2	Land/Building	161689	77450
C3	Plant	83188	40207
Total		368000	135700

B. Experimental Setup

Five comparative methods are used to illustrate the effectiveness of our experiment, including transductive SVM (TSVM [47]), DAN [29], DANN [33], ED-DMM-UDA [4], and CDA [40]. In all experiments, 180 samples are taken from each category in the source domain to ensure the fairness of the experiment. For DANN and DAN methods, before network training, the training samples are standardized so that source domain and target domain samples obey the standard normal distribution N (0,1).

1) For TSVM, an SVM classifier is trained using labeled source samples, and all unlabeled target domain samples are predicted. Then, the target domain samples with pseudo-labels are added to the training sets to retrain the SVM classifier. Each iteration includes relabeling of target domain samples and retraining of classifier SVM. The kernel function is linear. C is the penalty term coefficient of SVM classifier. It can be chosen from [1,1.5,10,100]. Cl and Cu represent the penalty factors for labeled samples (source domain) and unlabeled samples (target domain), which are used to adjust the weights of different samples. The Cl is 1.5, and the Cu is 0.001.
2) For DAN, the feature extractor uses resnet50, the last layer uses MMD adaptation, and the kernel is 5. For all datasets, the iteration is 1000, the batch size is 36, and the learning rate is $lr = lr_0/(1+\alpha i)^p$, where α is 10, p is 0.75, and lr_0 is 0.001. The optimizer is stochastic gradient descent (SGD), the moment is 0.9, and the weight decay is 0.0005. For the Shanghai–Hangzhou dataset, the input size is 1×1, and for the other datasets, the input size is 5×5.

TABLE VI
STRUCTURE OF OUR NETWORK

	Layer Name	OUTPUT SHAPE	Filter Size	Padding	BN	Relu
E	INPUT	$5 \times 5 \times ch$	N/A	N	N	N
	Conv 1	$5 \times 5 \times 200$	$1 \times 1{,}200$	N	Y	Y
	Conv 2	$5 \times 5 \times 200$	$1 \times 1{,}200$	N	Y	Y
	Conv3	$5 \times 5 \times 200$	$1 \times 1{,}200$	N	Y	Y
	Conv 4	$5 \times 5 \times 200$	$1 \times 1{,}200$	N	Y	Y
	AvgPool	$1 \times 1 \times 200$	5	N	N	N
	Flatten	200	N/A	N	N	N
	FC	128	N/A	N	N	N
C1 *C2*	FC1	128	N/A	N	Y	Y
	FC2	64	N/A	N	Y	Y
	FC3	NUM_CLASS	N/A	N	N	N

3) For DANN, resnet18 is used as the structure of feature extractor, and the domain classifier consists of three fully connected layers to distinguish the source domain and target domain. For all datasets, the learning rate is 0.01, the optimizer is SGD, the moment is 0.9, and the weight decay is 0.0005. The epoch is 1000. The input sizes of the four datasets are the same as DAN.
4) For ED-DMM-UDA, the experimental setup is the same as in [4]. For Shanghai–Hangzhou, the input size is 1×1 and others are 5×5. The setting of all hyperparameters is consistent with that in this article. For Pavia and Houston datasets, the input size is 5×5, the source domain is trained 1000 epochs to make it sufficiently clustered, and the target domain is trained 20 epochs.
5) For CDA, different datasets have different parameters. The ratio of $\lambda 1$ and $\lambda 2$ is 20:1. For parameter β, Houston, Indian, and Shanghai–Hangzhou are 0.35, and Pavia is 0.25. Other parameter settings are consistent with those in this article.
6) The CLDA method is trained via SGD optimizer, the moment is 0.9, and the weight decay is 0.0005. The learning rate is 0.01. The total number of training epoch *NUM* is set to 100. The batch is 36. The *TRAIN_NUM* is 20. The confidence operation is four times, and it is repeated every 20 epochs to update network parameters. And α is 0.1, β is 0.01, and γ is 0.01. It is worth noting that the Shanghai–Hangzhou and Houston datasets obey the standard normal distribution N (0,1), and the Indiana and Pavia datasets do not process. We classify the test data using the output of feature fusion classification during the test phase. The details of the network structure are shown in Table VI. For the SVM classifier, the radial basis function RBF kernel is adopted and other parameters use the default settings.

In order to validate the proposed method, OA, AA, and K are used for evaluation.

C. Experimental Results

Tables VII–X shows OA, AA, and K in different comparison methods for four different datasets, as well as the classification accuracy of each class. All experiments were performed ten

TABLE VII

CLASSIFICATION RESULTS (VALUES ± STANDARD DEVIATION) ON THE INDIANA DATASET (180 TRAINING SAMPLES PER CLASS)

Class	TSVM	DAN	DANN	ED-DMM-UDA	CDA	CLDA
1	**59.25**	7.56	2.08	0.20	2.05	0.75
2	0.55	41.06	52.90	**62.83**	43.39	60.84
3	16.44	34.18	32.01	38.54	**49.49**	37.78
4	58.54	69.53	62.72	84.70	65.64	**96.61**
5	0.87	27.88	25.62	41.76	14.09	52.21
6	0.66	24.81	41.17	1.28	15.43	0.18
7	96.73	94.19	94.04	94.05	**98.36**	95.55
OA	40.46 ±2.32	51.47 ±0.03	52.64 ±0.04	56.92 ±0.03	53.64 ±0.18	**58.79 ±1.77**
AA	33.41 ±1.90	42.74 ±2.79	44.36 ± 3.89	46.19 ± 0.12	41.21 ±1.75	**49.13 ±1.74**
K×100	29.31 ±1.87	40.76 ±0.01	45.61 ± 0.01	45.61 ± 0.02	40.76 ±0.05	**49.12 ±2.01**

TABLE VIII

CLASSIFICATION RESULTS (VALUES ± STANDARD DEVIATION) ON THE HOUSTON DATASET (180 TRAINING SAMPLES PER CLASS)

Class	TSVM	DAN	DANN	ED-DMM-UDA	CDA	CLDA
1	**99.70**	74.69	47.82	68.14	65.04	64.97
2	27.04	73.49	**88.76**	85.45	86.32	88.04
3	24.33	**72.55**	57.74	57.86	66.92	71.73
4	92.73	**95.00**	36.36	79.55	77.27	93.16
5	81.36	79.43	**98.20**	97.77	98.02	96.13
6	49.27	54.55	51.35	56.62	60.16	**61.71**
7	62.39	64.56	75.40	**75.97**	73.46	77.71
OA	52.03 ±0.32	61.45 ±2.18	62.61 ±2.10	66.09 ±1.83	68.44 ±0.77	**70.12 ±2.43**
AA	62.40 ±1.30	73.47 ±1.74	65.09 ±3.54	74.48 ±1.66	75.31 ±1.58	**79.07 ±0.38**
K×100	34.29 ±0.80	47.08 ±2.36	49.33 ±1.98	53.10 ±2.18	55.65 ±0.77	**57.75 ±2.49**

TABLE IX

CLASSIFICATION RESULTS (VALUES ± STANDARD DEVIATION) ON THE PAVIA DATASET (180 TRAINING SAMPLES PER CLASS)

Class	TSVM	DAN	DANN	ED-DMM-UDA	CDA	CLDA
1	**98.55**	98.31	98.46	98.21	97.27	96.68
2	96.73	96.56	96.21	96.63	98.54	**99.82**
3	69.96	70.98	68.72	72.40	**93.70**	80.12
4	32.19	58.11	75.67	67.94	82.07	**84.46**
5	**100**	**100**	99.99	**100**	**100**	**100.00**
6	23.02	24.31	23.05	25.46	68.97	**91.23**
7	86.39	85.00	86.67	83.92	81.06	**90.48**
OA	76.03 ±0.02	80.68 ±0.02	83.90 ±0.05	82.50 ±0.50	89.78 ±3.03	**92.80 ±0.67**
AA	72.41 ±2.22	76.18 ±1.62	78.40 ±0.47	77.80 ±0.56	88.80 ±5.05	**91.83 ±1.09**
K×100	70.70 ±3.05	76.47 ±2.34	80.41 ±0.50	78.73 ±0.54	87.71 ±3.66	**91.32 ±0.81**

times to eliminate the effect of random sampling, and the mean of them is taken as the final classification result.

For the test part, the target domain data are sent into feature extractor E, and the extracted features are sent into two classifiers $C1$ and $C2$, respectively. The features output at the last layer of the classifiers are added up, and the index of the maximum value in each line is taken, that is, the prediction label of each sample is obtained. We can see the following from the results.

TABLE X

CLASSIFICATION RESULTS (VALUES ± STANDARD DEVIATION) ON THE SHANGHAI–HANGZHOU DATASET (180 TRAINING SAMPLES PER CLASS)

Class	TSVM	DAN	DANN	ED-DMM-UDA	CDA	CLDA
1	97.50	95.46	94.14	99.59	99.83	**99.94**
2	64.31	84.48	85.66	84.20	85.64	**88.49**
3	66.42	74.86	80.12	86.53	86.38	**92.01**
OA	82.43 ±1.97	83.09 ±1.36	85.15 ±3.16	86.94 ±1.09	87.74 ±2.56	**91.05 ±1.64**
AA	82.90 ±3.60	84.94 ±2.16	86.64 ±3.25	90.11 ±0.73	90.61 ±2.04	**93.48 ±1.33**
K×100	69.28 ±3.83	70.78 ±2.79	74.69 ±4.90	78.02 ±1.55	79.25 ±4.24	**84.79 ±2.68**

1) The classification result of the deep learning algorithm is better than that of the traditional algorithm (TSVM). Taking Houston as an example, DAN algorithm is 9.42% higher than TSVM method in OA. The deep learning method obtains discriminative features by establishing a hierarchical structure network, which has better classification performance.
2) For domain alignment, only using MMD (DAN) or discriminant (DANN) is lower than the result of adding clustering (ED-DMM-UDA) or pseudo-label (CDA) at the same time. The former only closes the margin distribution of the two domains without considering the specific class-wise information. Whereas, for ED-DMM-UDA, the source domain features are first clustered, and then, the target domain is trained to form the same clustering as the source domain to close the category. For CDA, the source domain is fully trained, and then, the target domain is predicted to obtain the prediction probability. This result is used between each category and the alignment between classes. Therefore, they also consider discriminability while ensuring transferability. The results of the Houston dataset in Table VIII, ED-DMM-UDA, are 4.64% higher than DAN and 3.48% higher than DANN. The results of CDA are 6.99% higher than DAN and 5.83% higher than DANN.
3) Considering the category level alignment, the results of CDA are superior to the results of ED-DMM-UDA in some datasets (Pavia, Houston, Shanghai–Hangzhou). Both of the methods first use source domain data to train the network, and then, CDA predicts the target domain to obtain the prediction probability. ED-DMM-UDA encourages the target features to form clusters similar to the source features. According to the Pavia dataset in Table IX, CDA results are 7.28% higher than ED-DMM-UDA. It can be seen from Table III that in some classes, the number of samples in the source and target domains is quite different. Therefore, the clustering method cannot achieve good performance.

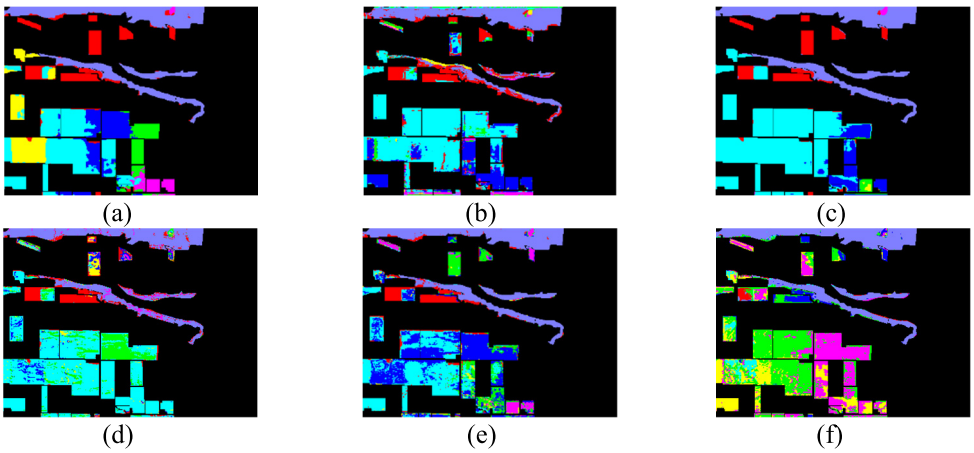

Fig. 6. Classification map for Indiana with different methods: (a) TSVM; (b) DAN; (c) DANN; (d) ED-DMM-UDA; (e) CDA; and (f) CLDA.

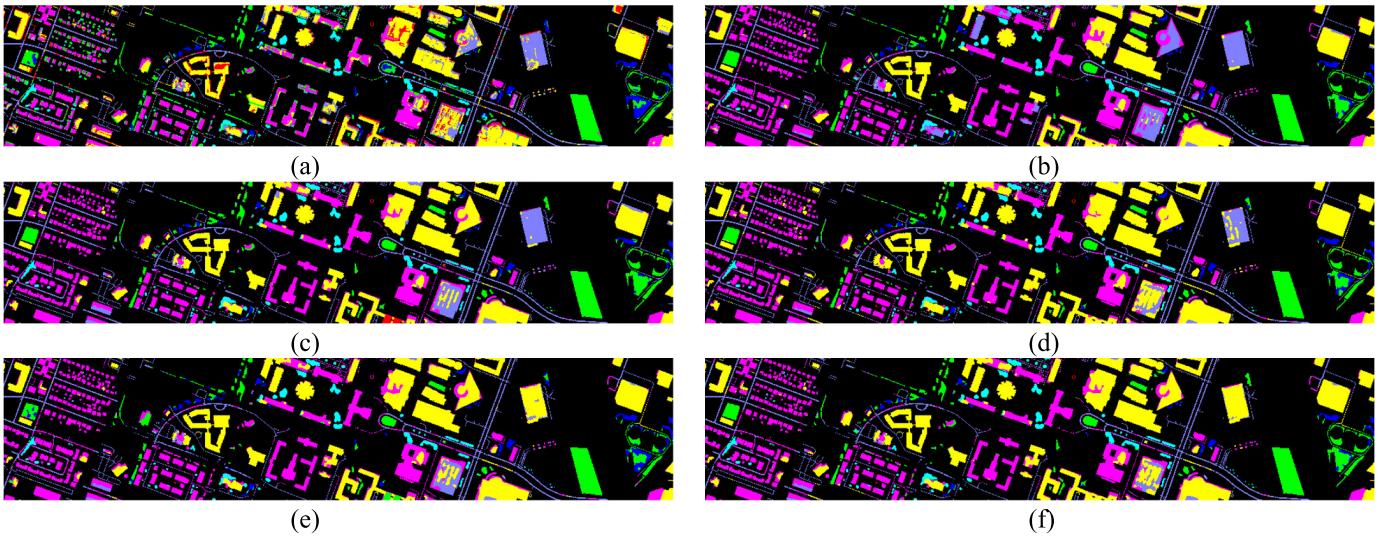

Fig. 7. Classification map for Houston 2018 with different methods: (a) TSVM; (b) DAN; (c) DANN; (d) ED-DMM-UDA; (e) CDA; and (f) CLDA.

And the prediction probability of target domain can further supervise the training of the networks and make it have better classification performance in the target domain.

4) Considering pseudo-labels, our method is better than CDA. It can be seen from Shanghai–Hangzhou dataset in Table X, our result is 3.31% higher than CDA. For CDA, after the network is trained with the source domain, the target domain is predicted, and then, the predicted result is used for interclass alignment operation. However, the predicted result of the target domain may not be accurate, even if entropy regularization is used to constrain it. Therefore, in our method, pseudo-labels are further processed. Through confident learning, more accurate samples are selected for training to improve the classification performance.

Table VIII shows the detailed results on the Houston dataset. It can be seen that the class-wise alignment (CDA) is better than MMD (DAN) and domain discriminator (DANN) alignment, indicating that considering conditional distribution and marginal distribution together can better enhance transferability and ensure discrimination. Compared with CDA, our method's OA improved by 1.48%, AA improved by 3.76%, and K improved by 2.1%, further indicating that the accuracy of the pseudo-label of the target domain used in networks training should be guaranteed while considering the category information. In addition to using performance measures to evaluate experimental performance, Figs. 6–9 also show the classification prediction maps for the target domain. It can be seen that compared with other methods, the proposed CLDA method can correctly classify more samples, so the mapping maps are more consistent with the ground-truth maps.

D. Ablation Experiments

In order to further prove the validity of confident learning and entropy regularization, we adopt three methods: 1) only use two classifiers (B); 2) use two classifiers and confident learning (B + C); and 3) use two classifiers, confident learning, and entropy regularization (B + C + E). As shown in Table XI, confident learning is useful for pseudo-label

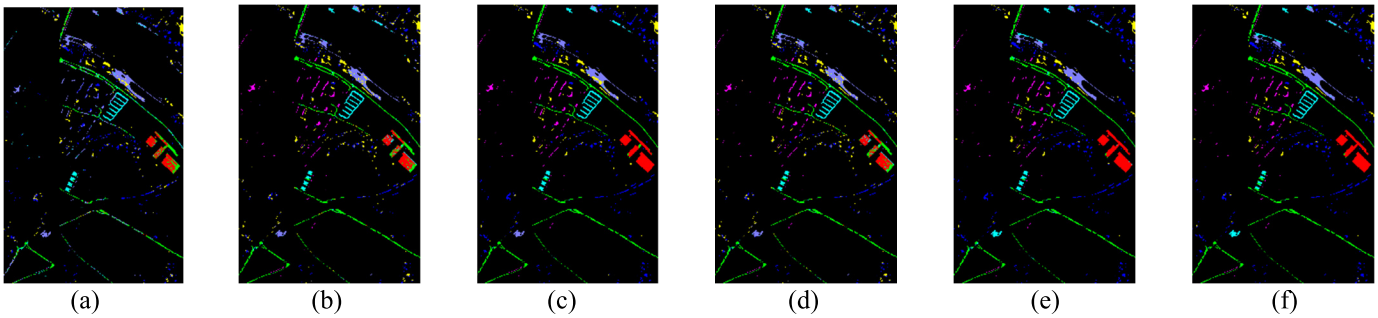

Fig. 8. Classification map for PC with different methods: (a) TSVM; (b) DAN; (c) DANN; (d) ED-DMM-UDA; (e) CDA; and (f) CLDA.

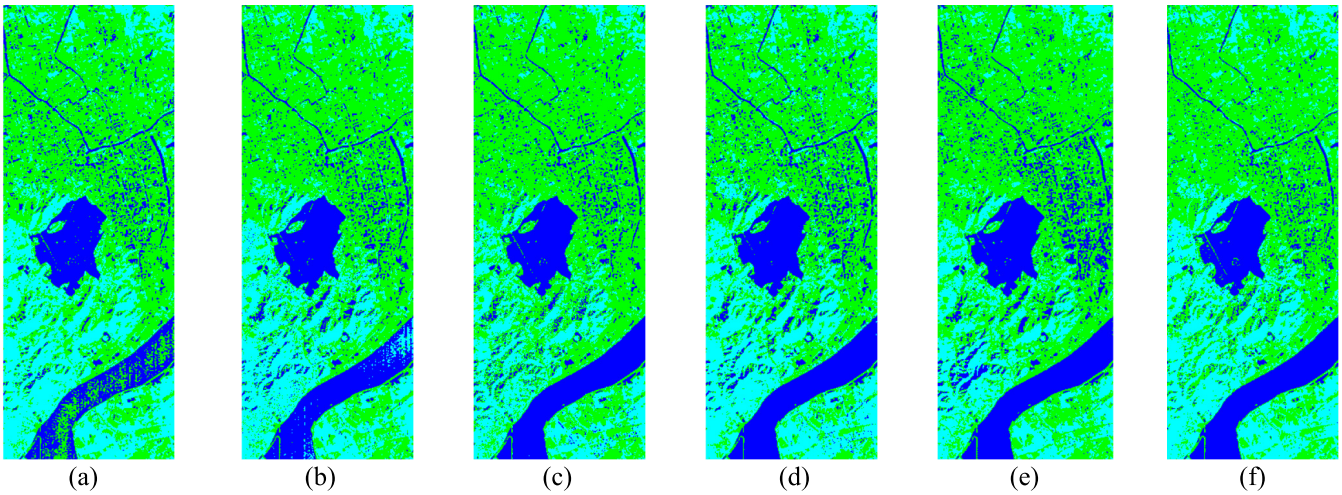

Fig. 9. Classification map for Hangzhou with different methods: (a) TSVM; (b) DAN; (c) DANN; (d) ED-DMM-UDA; (e) CDA; (f) CLDA.

TABLE XI
CLASSIFICATION RESULTS (%) FOR DIFFERENT METHODS WITH TWO CLASSIFIERS (B), CONFIDENT LEARNING (C), AND ENTROPY REGULARIZATION (E)

methods	Indiana	Houston	Pavia	Shanghai-Hangzhou
B	56.51	67.5	91.46	87.17
B+C	58.12	69.56	92.58	90.84
B+C+E	58.79	70.12	92.8	91.05

processing. Meanwhile, entropy regularization is added to further constrain the accuracy of pseudo-labels.

E. Computational Complexity

In order to compare the computational efficiency of all methods, the computational complexity of the proposed method is evaluated by analyzing the training time, testing time, the number of parameters, and the number of floating-point operations (FLOPS). The workstation used in the experiment is configured with an Intel Xeon processor (3.7 GHz), 64 GB of memory, and an Nvidia TITAN RTX 24-GB graphics card. The open-source software framework is Pytorch. Table XII shows the training and testing time, FLOPS, and network parameters of the above method on the target domain in the four datasets. Our method can achieve better results while the training time was not significantly different.

F. Sensitivity Analysis of Parameters in CLDA Network

There are six parameters, i.e., *TRAIN_NUM*, *NUM*, α, β, γ, and patch_size, in the CLDA network, where *TRAIN_NUM* is the epoch number of beginning confident learning, *NUM* is the total epoch number, α controls the weight of target classification loss, β denotes the weight of CDD loss, and γ expresses the weight of entropy loss. Take the Houston dataset for example to select these parameters.

For *TRAIN_NUM*, the loss values of Houston 2013 can be shown in Fig. 10(a). It was tested only using domain adaptation, and the total epoch is 100. Through the Houston 2013 of training set and validation set (the remaining samples), loss values can be seen that when the epoch is equal to 20, the two losses tended to be flat; thus, the *TRAIN_NUM* can be selected 20.

For *NUM*, it is not only the number of training but also related to the choice of the number of confident learning. We counted the percentage of confident samples that were truly correct after each confident learning. The percentage results are shown in Fig. 10(b). The training epoch is 200, confident learning is conducted every 20 epochs, and the times equal 1 which means epoch equals 20. When the time is 4 (epoch is equal to 80), the percentage can be the best.

TABLE XII
COMPUTATION TIME (S), FLOPS, AND PARAMETERS ON EACH DATASET WITH DIFFERENT METHODS

	Methods	TSVM	DAN	DANN	ED-DMM-UDA	CDA	CLDA
UP-PC	Training time	783.52	800.73	27.90	329.72	226.33	288.68
	Testing time	1.45	12.94	4.17	8.16	0.25	1.33
	FLOPS	-	27166983	16341960	3633402	20000000	3633983
	#params	-	23832839	13329224	181074	90260	182558
IP	Training time	6124.28	636.90	134.55743	860.84	224.85	156.78
	Testing time	2.34	18.74	3.98	8.70687	0.60	4.79
	FLOPS	-	30497415	19672392	4223402	30000000	4223983
	#params	-	24202887	13699272	204674	119760	206158
Houston13-18	Training time	1994.87	786.65	27.64	1907.54	227.43	293.4306
	Testing time	1.20	17.02	5.34	12.98	0.36	1.72
	FLOPS	-	25642887	14817864	3363402	20000000	3363983
	#params	-	23663495	13159880	170274	76760	171758
Shanghai-Hangzhou	Training time	546.92	512.75	32.28	1019.63	14172432	287.41
	Testing time	7.34	68.28	14.24	18.68	1.12	8.50744
	FLOPS	-	24148483	13634756	201402	30000000	4113983
	#params	-	24125699	13628228	200274	113253	201758

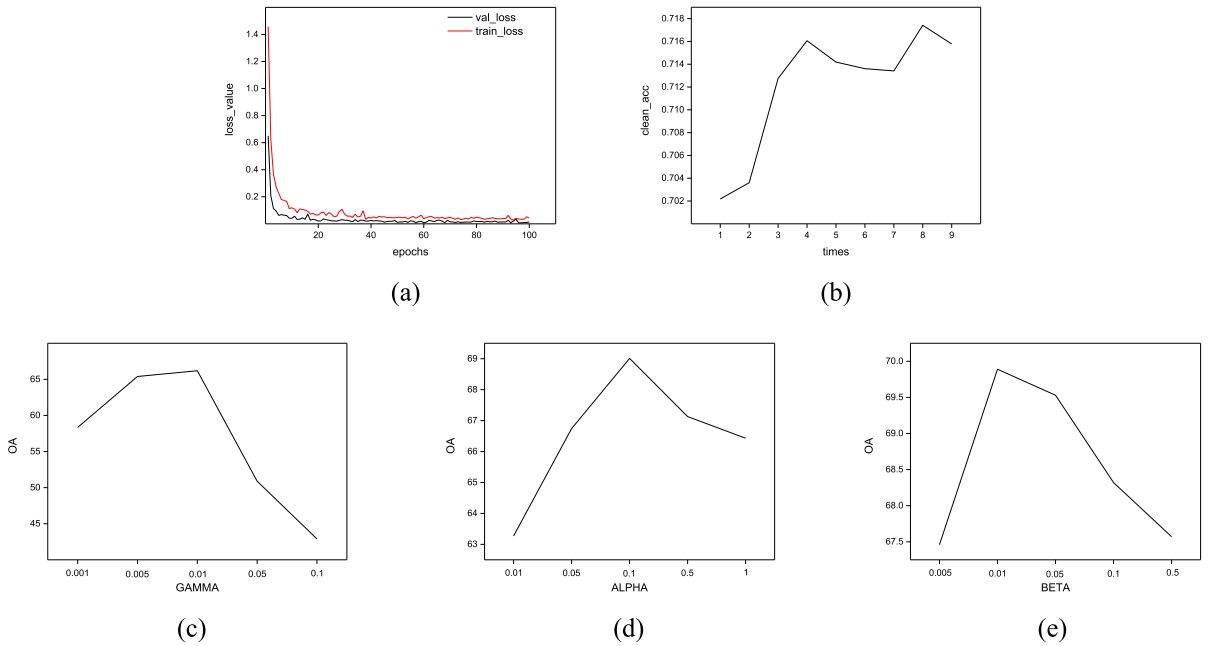

Fig. 10. Sensitivity analysis of parameters using Houston dataset in CLDA method: (a) parameter *TRAIN_NUM*; (b) parameter *NUM*; (c) parameter γ; (d) parameter α; and (e) parameter β.

For γ, five different values (0.001, 0.005, 0.01, 0.05, and 0.1) were tested with α and β equaling to zero. The classification results on Houston 2018 are shown in Fig. 10(c). When γ is equal to 0.01, the classification accuracy is higher.

For α, five different values (0.01, 0.05, 0.1, 0.5, and 1) were tested with γ equal to 0.01 and β equal to 0. The classification results on Houston 2018 are shown in Fig. 10(d). When α is equal to 0.1, the result can be higher.

For β, five different values (0.005, 0.01, 0.05, 0.1, and 0.5) were tested with γ equal to 0.01 and α equal to 0.1. The results on Houston 2018 can be seen in Fig. 10(e). When β is 0.01, the result is higher.

To verify the effect of different patch sizes of input samples on the different classification methods, we also select patches of size 1×1, 3×3, 5×5, and 7×7 for each method from the four datasets. From the results shown in Fig. 11, we can

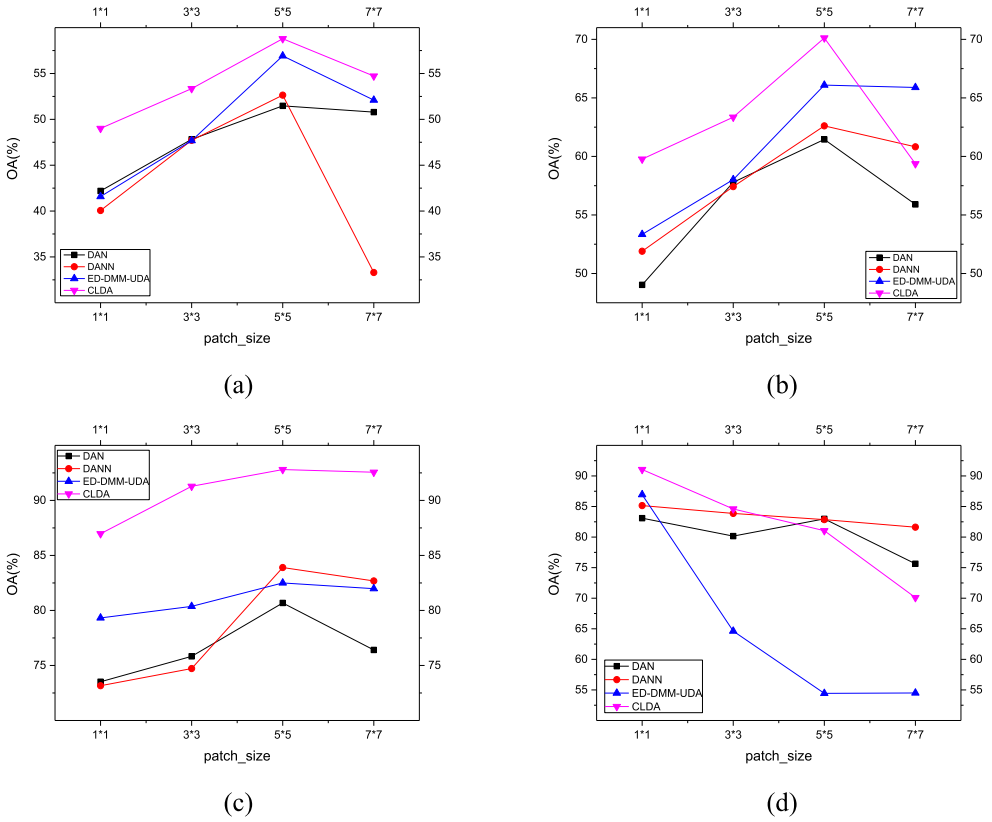

Fig. 11. Sensitivity analysis of patch size in different methods: (a) Indiana; (b) Houston; (c) Pavia; and (d) Shanghai–Hangzhou.

find that the OA of the classification methods hits its peak with a patch of size 5×5 in Indiana, Houston, and Pavia, but the highest OA in Shanghai–Hangzhou is achieved with a patch of size 1×1. The results demonstrate that the setup of patch size in the above experiments is appropriate, having a favorable impact on the performance of different classification methods.

V. Conclusion

In this article, we proposed a UDA framework combining adversarial learning with confident learning. It can achieve discriminative feature representation using high-confidence target labels and reduce the interdomain discrepancy. Since our selection strategy prunes low-confidence labeled instances of target domain, this can be seen as instance-based approaches, which further benefit the feature-based approach. Therefore, this kind of instance-based approach can be integrated with other domain adaptation methods. In the future work, since high-confidence target pseudo-labels exist, more supervised optimization strategies will be explored on the target domain data to perform domain adaptation.

References

[1] J. Liu *et al.*, "Estimating the forage neutral detergent fiber content of Alpine grassland in the Tibetan plateau using hyperspectral data and machine learning algorithms," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–17, 2022, doi: 10.1109/TGRS.2021.3105482.

[2] B. Luo, C. Yang, J. Chanussot, and L. Zhang, "Crop yield estimation based on unsupervised linear unmixing of multidate hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 1, pp. 162–173, Jan. 2013, doi: 10.1109/TGRS.2012.2198826.

[3] P. Ghamisi, J. Plaza, Y. Chen, J. Li, and A. J. Plaza, "Advanced spectral classifiers for hyperspectral images: A review," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 1, pp. 8–32, Mar. 2017, doi: 10.1109/MGRS.2016.2616418.

[4] 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, doi: 10.1109/TGRS.2019.2946318.

[5] P. Ghamisi *et al.*, "Advances in hyperspectral image and signal processing: A comprehensive overview of the state of the art," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 4, pp. 37–78, Dec. 2017, doi: 10.1109/MGRS.2017.2762087.

[6] S.-E. Qian, "Hyperspectral satellites, evolution, and development history," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 7032–7056, 2021, doi: 10.1109/JSTARS.2021.3090256.

[7] J. Jiang, J. Ma, Z. Wang, C. Chen, and X. Liu, "Hyperspectral image classification in the presence of noisy labels," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 2, pp. 851–865, Feb. 2018, doi: 10.1109/tgrs.2018.2861992.

[8] K. Safari, S. Prasad, and D. Labate, "A multiscale deep learning approach for high-resolution hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 18, no. 1, pp. 167–171, Jan. 2021, doi: 10.1109/lgrs.2020.2966987.

[9] S. Zhang, X. Kang, P. Duan, B. Sun, and S. Li, "Polygon structure-guided hyperspectral image classification with single sample for strong geometric characteristics scenes," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2021, Art. no. 5511212, doi: 10.1109/TGRS.2021.3094582.

[10] J. M. Bioucas-Dias, A. Plaza, G. Camps-Valls, P. Scheunders, N. M. Nasrabadi, and J. Chanussot, "Hyperspectral remote sensing data analysis and future challenges," *IEEE Geosci. Remote Sens. Mag.*, vol. 1, no. 2, pp. 6–36, Jun. 2013, doi: 10.1109/MGRS.2013.2244672.

[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, doi: 10.1109/MGRS.2016.2548504.

[12] F. Zhuang *et al.*, "A comprehensive survey on transfer learning," *Proc. IEEE*, vol. 109, no. 1, pp. 43–76, Jan. 2021, doi: 10.1109/JPROC.2020.3004555.

[13] J. Wang, C. Lan, C. Liu, Y. Ouyang, and T. Qin, "Generalizing to unseen domains: A survey on domain generalization," in *Proc. 13th Int. Joint Conf. Artif. Intell.*, Aug. 2021, pp. 4627–4635, doi: 10.24963/IJCAI.2021/628.

[14] J. T. Peng, W. Sun, L. Ma, and Q. Du, "Discriminative transfer joint matching for domain adaptation in hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 6, pp. 972–976, Jun. 2019, doi: 10.1109/LGRS.2018.2889789.

[15] L. Ma, C. Luo, J. Peng, and Q. Du, "Unsupervised manifold alignment for cross-domain classification of remote sensing images," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 10, pp. 1650–1654, Oct. 2019, doi: 10.1109/LGRS.2019.2902615.

[16] L. Ma, M. M. Crawford, L. Zhu, and Y. Liu, "Centroid and covariance alignment-based domain adaptation for unsupervised classification of remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 4, pp. 2305–2323, Apr. 2019, doi: 10.1109/TGRS.2018.2872850.

[17] W. M. Kouw and M. Loog, "A review of domain adaptation without target labels," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 3, pp. 766–785, Mar. 2021, doi: 10.1109/TPAMI.2019.2945942.

[18] J. Jiang and C. Zhai, "Instance weighting for domain adaptation in NLP," in *Proc. ACL*, Prague, Czech Republic, Jun. 2007, pp. 264–271. [Online]. Available: https://aclanthology.org/P07-1034

[19] B. Schölkopf, J. Platt, and T. Hofmann, "Correcting sample selection bias by unlabeled data," in *Proc. NeurIPS*, 2007, pp. 601–608.

[20] M. Sugiyama, S. Nakajima, H. Kashima, P. von Bünau, and M. Kawanabe, "Direct importance estimation with model selection and its application to covariate shift adaptation," in *Proc. NeurIPS*, Red Hook, NY, USA, 2007, pp. 1433–1440.

[21] Y. Yao and G. Doretto, "Boosting for transfer learning with multiple sources," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 1855–1862, doi: 10.1109/CVPR.2010.5539857.

[22] H. Wei, L. Ma, Y. Liu, and Q. Du, "Combining multiple classifiers for domain adaptation of remote sensing image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 1832–1847, 2021, doi: 10.1109/JSTARS.2021.3049527.

[23] Y. Mansour and M. Schain, "Robust domain adaptation," *Ann. Math. Artif. Intell.*, vol. 71, no. 4, pp. 365–380, Aug. 2014.

[24] S. Zhong and Y. Zhang, "An iterative training sample updating approach for domain adaptation in hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 18, no. 10, pp. 1821–1825, Oct. 2021, doi: 10.1109/LGRS.2020.3007021.

[25] 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, doi: 10.1109/ICCV.2013.274.

[26] J. Wang, Y. Chen, S. Hao, W. Feng, and Z. Shen, "Balanced distribution adaptation for transfer learning," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2017, pp. 1129–1134, doi: 10.1109/ICDM.2017.150.

[27] H. Zhang, L. Liu, Y. Long, and L. Shao, "Unsupervised deep hashing with pseudo labels for scalable image retrieval," *IEEE Trans. Image Process.*, vol. 27, no. 4, pp. 1626–1638, Apr. 2018, doi: 10.1109/TIP.2017.2781422.

[28] H. Wu and S. Prasad, "Semi-supervised deep learning using pseudo labels for hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1259–1270, Mar. 2018, doi: 10.1109/TIP.2017.2772836.

[29] M. Long, Y. Cao, J. Wang, and M. I. Jordan, "Learning transferable features with deep adaptation networks," in *Proc. ICML*, Lille, France, 2015, pp. 97–105.

[30] C. Yu, C. Liu, M. Song, and C.-I. Chang, "Unsupervised domain adaptation with content-wise alignment for hyperspectral imagery classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022, doi: 10.1109/LGRS.2021.3126594.

[31] W. Wang, L. Ma, M. Chen, and Q. Du, "Joint correlation alignment-based graph neural network for domain adaptation of multitemporal hyperspectral remote sensing images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 3170–3184, 2021, doi: 10.1109/JSTARS.2021.3063460.

[32] Y. Zhang, W. Li, M. Zhang, Y. Qu, R. Tao, and H. Qi, "Topological structure and semantic information transfer network for cross-scene hyperspectral image classification," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–14, 2021, doi: 10.1109/TNNLS.2021.3109872.

[33] Y. Ganin *et al.*, "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2030–2096, 2016.

[34] H. Wang, Y. Cheng, C. L. P. Chen, and X. Wang, "Hyperspectral image classification based on domain adversarial broad adaptation network," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–13, 2022, doi: 10.1109/TGRS.2021.3128162.

[35] N. Makkar, L. Yang, and S. Prasad, "Adversarial learning based discriminative domain adaptation for geospatial image analysis," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 150–162, 2022, doi: 10.1109/JSTARS.2021.3132259.

[36] 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, doi: 10.1109/TGRS.2020.3015357.

[37] C. Yu, C. Liu, H. Yu, M. Song, and C.-I. Chang, "Unsupervised domain adaptation with dense-based compaction for hyperspectral imagery," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 12287–12299, 2021, doi: 10.1109/JSTARS.2021.3128932.

[38] Y. Zhang, W. Li, R. Tao, J. Peng, Q. Du, and Z. Cai, "Cross-scene hyperspectral image classification with discriminative cooperative alignment," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 1, pp. 9646–9660, Nov. 2021, doi: 10.1109/TGRS.2020.3046756.

[39] Z. Wang, B. Du, Q. Shi, and W. Tu, "Domain adaptation with discriminative distribution and manifold embedding for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 7, pp. 1155–1159, Jul. 2019, doi: 10.1109/LGRS.2018.2889967.

[40] 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, doi: 10.1109/TGRS.2020.2997863.

[41] S. Li, F. Lv, B. Xie, C. H. Liu, J. Liang, and C. Qin, "Bi-classifier determinacy maximization for unsupervised domain adaptation," in *Proc. AAAI*, 2021, pp. 8455–8464.

[42] X. Chen, S. Wang, M. Long, and J. Wang, "Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation," in *Proc. ICML*, vol. 97, Jun. 2019, pp. 1081–1090. [Online]. Available: https://proceedings.mlr.press/v97/chen19i.html

[43] C. Northcutt, L. Jiang, and I. Chuang, "Confident learning: Estimating uncertainty in dataset labels," *J. Artif. Intell. Res.*, vol. 70, pp. 1373–1411, Apr. 2021, doi: 10.1613/jair.1.12125.

[44] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada, "Maximum classifier discrepancy for unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3723–3732, doi: 10.1109/CVPR.2018.00392.

[45] Y. Grandvalet and Y. Bengio, "Semi-supervised learning by entropy minimization," in *Proc. NeurIPS*, Cambridge, MA, USA, 2004, pp. 529–536.

[46] M. Ye, Y. Qian, J. Zhou, and Y. Y. Tang, "Dictionary learning-based feature-level domain adaptation for cross-scene hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 3, pp. 1544–1562, Mar. 2017, doi: 10.1109/TGRS.2016.2627042.

[47] T. Joachims, "Transductive inference for text classification using support vector machines," in *Proc. ICML*, San Francisco, CA, USA, 1999, pp. 200–209.

Zhuoqun Fang received the Ph.D. degree in pattern recognition from Northeastern University, Shenyang, China, in 2019.

He is currently a Lecturer with the College of Artificial Intelligence, Shenyang Aerospace University, Shenyang, and a Post-Doctoral Researcher with the Shenyang Institute of Computer Technology, Chinese Academy of Sciences, Shenyang. His current research interests include hyperspectral image analysis, reflectance estimation, scene understanding, image processing, and computer vision.

Yuexin Yang received the B.E. degree from the Hubei University of Education, Wuhan, China, in 2019. She is currently pursuing the master's degree with the School of Computer Science, Shenyang Aerospace University, Shenyang, China.

Her main research interests are hyperspectral image processing, computer vision, and image understanding.

Zhaokui Li received the M.S. degree in computer application from Liaoning University, Shenyang, China, in 2003, and the Ph.D. degree in computer software and theory from Wuhan University, Wuhan, China, in 2014.

He is currently a Professor with the School of Computer, Shenyang Aerospace University, Shenyang. His research interests include hyperspectral image analysis, computer vision, and machine learning.

Wei Li (Senior Member, IEEE) received the B.E. degree in telecommunications engineering from Xidian University, Xi'an, China, in 2007, the M.S. degree in information science and technology from Sun Yat-sen University, Guangzhou, China, in 2009, and the Ph.D. degree in electrical and computer engineering from Mississippi State University, Starkville, MS, USA, in 2012.

Subsequently, he spent one year as a Post-Doctoral Researcher at the University of California at Davis, Davis, CA, USA. He is currently a Professor with the School of Information and Electronics, Beijing Institute of Technology, Beijing, China. His research interests include hyperspectral image analysis, pattern recognition, and data compression.

Dr. Li is an Associate Editor of the IEEE SIGNAL PROCESSING LETTERS (SPL) and the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING (JSTARS).

Yushi Chen (Member, IEEE) received the Ph.D. degree from the Harbin Institute of Technology, Harbin, China, in 2008.

He is currently an Associate Professor with the School of Electronics and Information Engineering, Harbin Institute of Technology. His research interests include remote sensing data processing and machine learning.

Li Ma (Member, IEEE) received the B.S. and M.S. degrees from Shandong University, Jinan, China, in 2004 and 2006, respectively, and the Ph.D. degree in pattern recognition and intelligent system from the Huazhong University of Science and Technology, Wuhan, China, in 2011.

From 2008 to 2010, she was a Visiting Scholar with Purdue University, West Lafayette, IN, USA. She also visited Mississippi State University, Starkville, MS, USA, for five months in 2018. She is currently an Associate Professor with the School of Mechanical Engineering and Electronic Information, China University of Geosciences, Wuhan. Her research interests include hyperspectral data analysis, pattern recognition, and remote sensing applications.

Qian Du (Fellow, IEEE) received the Ph.D. degree in electrical engineering from the University of Maryland, Baltimore, MD, USA, in 2000.

She is currently the Bobby Shackouls Professor with the Department of Electrical and Computer Engineering, Mississippi State University, Starkville, MS, USA. Her research interests include hyperspectral remote sensing image analysis and applications, pattern classification, data compression, and neural networks.

Dr. Du is a fellow of the SPIE-International Society for Optics and Photonics. She received the 2010 Best Reviewer Award from the IEEE Geoscience and Remote Sensing Society. She was a Co-Chair of the Data Fusion Technical Committee of the IEEE Geoscience and Remote Sensing Society from 2009 to 2013, and the Chair of the Remote Sensing and Mapping Technical Committee of the International Association for Pattern Recognition from 2010 to 2014. She has served as an Associate Editor for the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING (JSTARS), the *Journal of Applied Remote Sensing*, and the IEEE SIGNAL PROCESSING LETTERS. Since 2016, she has been the Editor-in-Chief of the IEEE JSTARS. She is the General Chair of the 4th IEEE GRSS Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing, Shanghai, in 2012.