Convolutional Transformer-Based Few-Shot Learning for Cross-Domain Hyperspectral Image Classification

Yishu Peng, *Member, IEEE*, Yaru Liu, *Student Member, IEEE*, Bing Tu, *Member, IEEE*, and Yuwen Zhang, *Student Member, IEEE*

***Abstract*—In cross-domain hyperspectral image (HSI) classification, the labeled samples of the target domain are very limited, and it is a worthy attention to obtain sufficient class information from the source domain to categorize the target domain classes (both the same and new unseen classes). This article investigates this problem by employing few-shot learning (FSL) in a meta-learning paradigm. However, most existing cross-domain FSL methods extract statistical features based on convolutional neural networks (CNNs), which typically only consider the local spatial information among features, while ignoring the global information. To make up for these shortcomings, this article proposes novel convolutional transformer-based few-shot learning (CTFSL). Specifically, FSL is first performed in the classes of source and target domains simultaneously to build the consistent scenario. Then, a domain aligner is set up to map the source and target domains to the same dimensions. In addition, a convolutional transformer (CT) network is utilized to extract local-global features. Finally, a domain discriminator is executed subsequently that can not only reduce domain shift but also distinguish from which domain a feature originates. Experiments on three widely used hyperspectral image datasets indicate that the proposed CTFSL method is superior to the state-of-the-art cross-domain FSL methods and several typical HSI classification methods in terms of classification accuracy.**

***Index Terms*—Convolutional transformer (CT), cross-domain, few-shot learning (FSL), hyperspectral image (HSI), scene consistency.**

I. Introduction

HYPERSPECTRAL images (HSIs) are 3-D data cubes with 1-D spectral information in addition to the general 2-D spatial image [1], [2], [3] that integrate the characteristics of image and spectra. HSIs contain abundant spectral and spatial information [4], [5], which have been applied in land-use and land-cover classification and have gained increasing attention [6], [7], [8], [9]. In HSI classification, it is sufficient to labeled samples in the same scene such that a scene can be classified correctly. However, achieving labeling process is difficult for a newly collected HSI.

Cross-domain HSI classification was proposed for resolving the problem of difficult classification due to the scarcity of ground-cover labels [10], [11], [12], [13]. This aims to use the similarity of covering features between multiple HSIs to form classification and recognition criteria from an HSI with sufficient labeled pixels for model training and learning, which is called the source domain or source scene. Then, the model is used to identify and classify another HSI with similar scenes called the target domain or target scene that is seriously lacking in labeled pixels or even without available labeled pixels.

Inevitably, difficulties and challenges in cross-scene HSI classification tasks followed. Restricted by factors such as sensor differences, imaging time, location, and atmospheric environment, the acquired HSI has heterogeneity [14], [15], [16], [17]. Therefore, solving the distribution differences of the source and target domains is the key to cross-scene HSI classification, which is the domain adaptation problem. In recent years, a series of HSI classification approaches have been presented to achieve cross-scene learning tasks and solve domain adaptation problems, which can be roughly defined into two types: heterogeneity of feature distribution and heterogeneity of feature space.

The former refers to the HSIs collected by the same optical sensor under different angles, times, locations, etc., causing heterogeneity in the feature distribution between the same land covers in different scenes, which is manifested by the same number of spectral bands but the spectral curves may differ in the same class. The latter refers to the restriction of the parameters of the optical sensor, which leads to feature space heterogeneity between source and target domain HSIs; this manifests that not only the spectral bands are different in number, but the spectral curves of the identical class in different scenes would also be significantly different.

To address cross-scene classification, from the heterogeneity of feature distribution-based perspective, some works are operated to explore the similarity between the source and target domains, thus, solving the spectral offset problem. Deng et al. [18] proposed a feature embedding model based on deep metric

Manuscript received 4 November 2022; revised 21 December 2022; accepted 31 December 2022. Date of publication 5 January 2023; date of current version 20 January 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 61977022 and Grant 62271200, in part by the Foundation of Department of Water Resources of Hunan Province under Grant XSKJ2021000-12, Grant XSKJ2021000-13, and Grant XSKJ2022068-48, in part by the Natural Science Foundation of Hunan Province under Grant 2021JJ40226, in part by the Foundation of Education Bureau of Hunan Province under Grant 21B0590, Grant 21B0595, and Grant 20B062, and in part by the Scientific Research Fund of Education Department of Hunan Province under Grant 19A200. *(Corresponding author: Bing Tu.)*

Yishu Peng, Yaru Liu, and Yuwen Zhang are with the School of Information Science and Engineering, Hunan Institute of Science and Technology, Yueyang 414000, China (e-mail: lovepys@hnist.edu.cn; liuyarua@foxmail.com; yuwen_zhang@vip.hnist.edu.cn).

Bing Tu is with the School of Information Science and Engineering, Hunan Institute of Science and Technology, Yueyang 414000, China, and also with the Guangxi Key Laboratory of Cryptography and Information Security, Guilin University of Electronic Technology, Guilin 541000, China (e-mail: tubing@hnist.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2023.3234302

learning, which applies the features learned from the source scene to the target scene with an unsupervised domain adaptation technique. A maximum mean difference (MMD)-based graph optimal transmission (GOT) was proposed to align the distribution discrepancy of the source and target domains [19]. An unsupervised domain adaptation method was accomplished for cross-scene HSI classification by utilizing an integrated framework with spectral-spatial feature dense compaction [20]. The unsupervised domain adaptation method for feature learning does not demand labeled data in the target scene, but it requires having a small enough discrepancy between the source and target scenes. Although the heterogeneity of feature distribution-based methods enable to decrease data migration between two domains, they usually require that the target categories are the same as the source and can not classify the new unseen categories.

From the perspective of heterogeneity of feature space, Liu et al. [21] introduced spectral shift mitigation to simultaneously minimize the amplitude shift between source and target domains as well as the spectral variation for the target scene. Despite the great similarities in the data between the source and target domains, the classes between the two scenes may differ and new classes need to be considered. Recently, few-shot learning (FSL) [22], [23], [24] has been used to address the above problem, the goal of which is to classify a target class data given just a small number of labeled samples from each class. Li et al. [25] proposed a deep cross-domain few-shot learning (DCFSL) method for cross-scene classification of HSIs in the case of less labeled data, which overcomes domain shift by learning a domain-adaptive feature embedded space through a 3-D-CNN-based deep residual network from two mapping layers of the source and target sceneries that are used for ensuring that the inputs to the embedded feature extractor share equal dimensions. In addition, DCFSL makes it possible to perform domain distribution alignment by the domain discriminator. Zhang et al. [26] developed a dual graph cross-domain few-shot learning (DG-CFSL) method to mitigate the impact of domain transitions. DG-CFSL designs intradomain distribution extraction block (IDE-block) to carry out domain alignment using nonlocal spatial information which has powerful corresponding properties.

The foregoing FSL approaches enable increased classification accuracy with limited labels; they commonly extract features using a convolutional neural network (CNN) that have obtained significant results for cross-scene HSI classification. However, it is difficult for CNN to capture the sequence attributes of spectral features due to the limitations of its network backbone. In addition, the receptive field of CNN is limited which may easily cause the missing information in the down-sampling layer, and it needs to expand the convolution kernel to expand the receptive field, which causes dimensional disaster. A transformer network [27], [28] can be utilized to overcome the above issues because it can capture the sequence attributes of spectral features. Meanwhile, vision transformer (ViT) [29] has been proposed to apply a transformer in image classification, Chen et al. [30] developed a multistage vision transformer model to form pyramid feature extraction. Wu et al. [31] introduced spectrally enhanced and densely connected transformer model to capture local contextual and semantic features. Feng et al. [32] developed a novel spectral transformer with dynamic spatial sampling and gaussian positional embedding to take full advantage of the flexible nature of spatial sampling, to emphasize the importance of the central image element for HSI cube classification, and to improve the adaptability. Peng et al. [33] proposed a spatial–spectral transformer with cross-attention, which is composed of a dual-branch structures with spatial and spectral sequence. However, it tends to overlook some local information that may be important for HSI classification. To enhance information utilization and extract more discriminative features, we combine the CNN and transformer module and propose a convolutional transformer-based few-shot learning (CTFSL) structure for cross-domain HSI classification. Specifically, two FSLs are first executed simultaneously for the source and target domains. After mapping two domains' bands to the same dimensions through the distribution aligner, a feature extractor based on a convolutional transformer (CT) network is utilized to learn spectral-spatial features, which can both expand interclass distances and reduce innerclass distances. Furthermore, a domain discriminator is employed to tackle the domain separability problems that can not only classify the same target domain classes as source domain classes but also classify new unseen classes.

The major contributions presented in this article are grouped as follows.

1) A CTFSL framework is proposed where a novel FSL method is developed to solve classes that are scarcely represented, and an FSL loss is defined to avoid overfitting to underrepresented classes.
2) The CT network is designed by composing the convolutional neural network and a vision transformer, which achieves more effective feature embedding and extracts both local detail and global information for HSI patches.
3) An adversarial loss is introduced using domain discriminator based on FCN to match the prediction between two domains and optimize the proposed network model for a cross-domain task.
4) It can be observed that CTFSL can achieve better classification results than other cross-domain FSL methods on practical application.

The rest of this article is organized as follows. Section II briefly describes some relevant concepts. Section III explicitly explains the full details of the proposed CTFSL for cross-scene HSI classification. Section IV shows experimental results to demonstrate the superior performance of CTFSL. Finally, Section V concludes this article.

II. Related Work

This section introduces several relevant concepts to better explain the proposed CTFSL.

A. Domain Adaptation

In cross-scene HSI classification, domain adaption aims to transfer data knowledge from the source domain to the target domain by mapping the data features of two domains into the same feature space [34], [35]. Domain adaptation can solve the distribution discrepancy between the source and target domains by learning domain-invariant features. Domain adaption may be described in two forms: unsupervised domain adaptation [36], [20], [37] and supervised domain adaptation [38], [39], [40]. In domain adaptation, the source domain has rich learning information. Unsupervised domain adaptation refers to the target domain without labeled samples, while supervised domain adaptation means that the target domain has a few labeled samples. Our method leans toward supervised domain adaptation and proposes cross-scene few-shot domain adaptation.

B. Cross-Scene Few-Shot Learning

FSL is one type of meta-learning [41], [42] that processes images given only a small number of labeled samples [43]; FSL aims to construct a consistent scene of a source and target domain based on an FSL method through meta-learning [44], [45], [46]. In cross-scene HSI classification, FSL is usually defined as a K-way N-shot task [47] (i.e., N labeled samples of K unique classes) and N is very small, e.g., 1 or 5 [48]. First, two HSI datasets are given: the source dataset $\boldsymbol{X_s} \in \mathbb{R}^{S_D}$ and the target dataset $\boldsymbol{X_t} \in \mathbb{R}^{T_D}$, where $\boldsymbol{X_t}$ contains two parts D_f with labeled few-shot data and D_t with unlabeled test data, i.e., $\boldsymbol{X_t} = D_f \cup D_t$. Then, the numbers of categories in the source and target domains are marked with C_s and C_t separately. Generally, to guarantee diversity in the training samples, we set $C_s > C_t$, which is beneficial for meta-learning [49], [50].

In our method, we take the source data $\boldsymbol{X_s} \in \mathbb{R}^{S_D}$ and the target labeled few-shot data D_f as the training set for feature extraction and the target unlabeled data D_t as the test set for model evaluation. The FSL model operates on the task-based learning tactic in both the source and target domains, where each task is one single iteration of training. During every iteration, taking the FSL on the source dataset $\boldsymbol{X_s}$ as instance. A support set is first formed with C classes and K samples per class are randomly selected from $\boldsymbol{X_s}$. Therefore, the support set is expressed as $S = \{(x_i, y_i)\}_{i=1}^{C\times K}$. Analogously, a query set $Q = \{(x_j, y_j)\}_{j=1}^{C\times N}$ consists of N samples randomly selected from the identical C classes that are unique from the elements of the support set. It is note worthy that the sample labels of the query set are considered as unknown. In experiments, we usually set K significantly smaller than N which can simulate the practical few-shot classification scenarios. Summarizing, a C-way K-shot N-query FSL work is formed for the source dataset. The target FSL is similar to the source data.

C. Vision Transformer

After the publication of the vision transformer (ViT) [29], it has been broadly used in various tasks of computer vision due to its excellent performance such as HSI classification [51], [52], [53], [30]. ViT is derived from the structure of the original transformer [54], [55], [56] and is easy to transplant into different tasks. The original transformer, which is a typical encoder–decoder model, is proposed for natural language processing. Therefore, the transformer consists of two parts: the encoding and decoding components. The encoding component is composed of multiple encoder layers, each of which is made up of two sublayers: self-attention and feed-forward network [57]. Likewise, the decoding component also consists of a stack of decoder layers, but decoder inserts a third sublayer per layer, encoder–decoder attention, in addition the two sublayers of the encoder. The transformer is entirely based on self-attention mechanisms, which can realize input parameter sharing by the global contextual information.

Inspired by the tremendous achievements of the original transformers, ViT is an extension in the field of image classification. The original transformer only accepted sequential inputs (i.e., the input of the original transformer is 1-D embeddings). Therefore, the input image in ViT is first divided into a series of nonoverlapped fixed-size patches (i.e., 2-D patches) that are then projected into patch embeddings (i.e., flatten the 2-D patches into a 1-D image sequence). Finally, send the patch embeddings of the image into the transformer to extract features. Fig. 1 illustrates the Encoder structure of ViT and Fig. 2 illustrates the multihead self-attention model. “Q, K and V” in Fig. 2 are the new sequence vectors generated by linear projection, and the

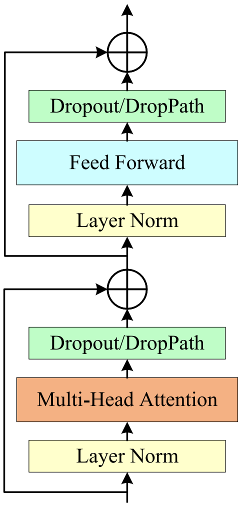


Fig. 1. Structure of vision transformer encoder.

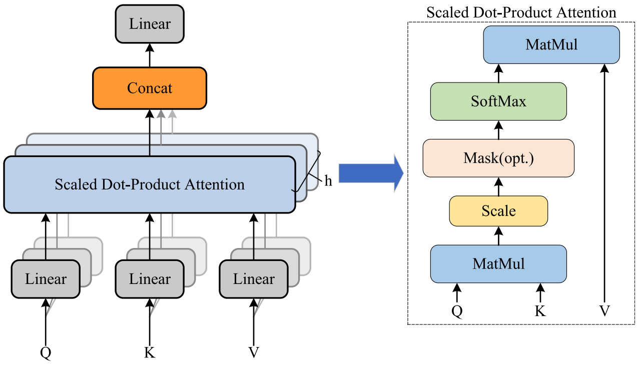


Fig. 2. Model of the multihead attention.

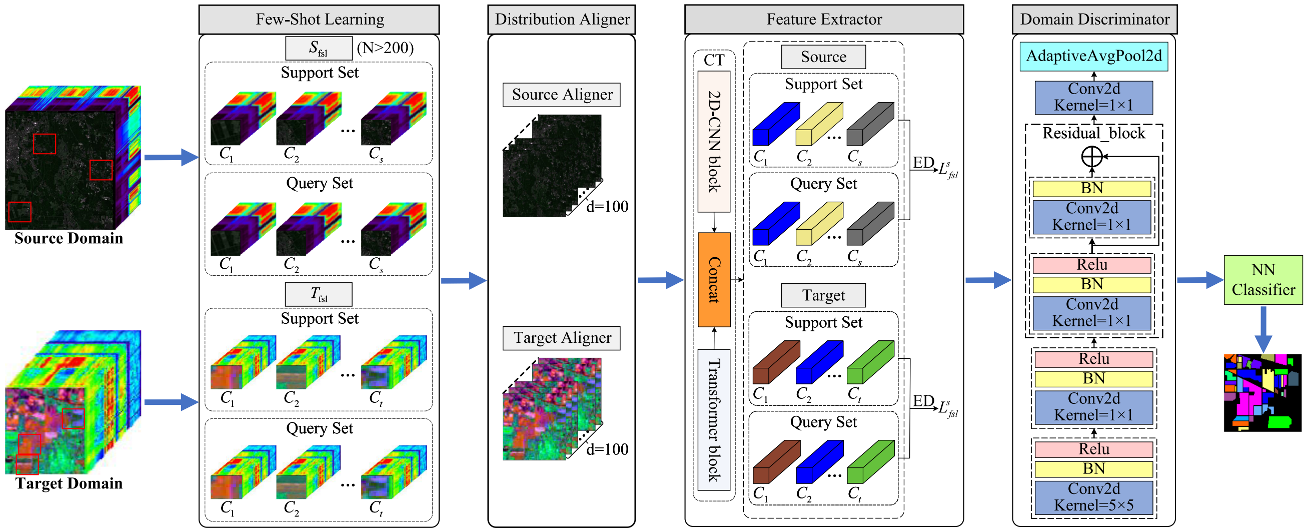


Fig. 3. Schematic of the proposed CTFSL classification method, including few-shot learning, distribution aligner, feature extractor, and domain discriminator.

self-attention can be calculated as follows:

$$\text{Attention}\,(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

where d_k is the dimension of K. The attention weights obtained from the dot product of Q and K are responsible for calculating the attention scores between each pair of different vectors that determine the level of attention given to the other data when encoding the data at the current location. $\sqrt{d_k}$ and Softmax normalize the attention scores to enhance the gradient stability to improve the training, and subsequently convert the scores into probabilities. Finally, according to the probability magnitude, each value vector is multiplied with the sum of the probabilities to assign attention weights to it and produce the final output vector.

III. Method

This section introduces the convolutional transformer-based few-shot learning (CTFSL) network for cross-scene HSI classification. Fig. 3 displays the structure diagram of the suggested CTFSL, which contains four parts: few-shot learning (FSL), distribution aligner, feature extractor, and domain discriminator. Specifically, executing FSL in both the source and target categories concurrently. Then, a distribution aligner is used before the feature extractor to map the source and target domains into an identical dimensions. Next, the feature extractor maps features from two domains into a scene-consistency metric space. The domain discriminator predicts the domain to which a feature belongs and achieves the distinguishability of the two domain classes.

A. Few-Shot Learning

Given the source domain data $\boldsymbol{X_s} \in \mathbb{R}^{S_D}$ having C_s classes and the target domain data $\boldsymbol{X_t} \in \mathbb{R}^{T_D}$ having C_t classes separately, the proposed CTFSL network has two FSL tasks: the source FSL task S_{fsl} and target FSL task T_{fsl}. Two kinds of FSL are executed in the classes with both the source and target domains simultaneously by episodes, enabling scene consistency between the source and target domain data and building cross-scene classification model.

1) Source FSL: In the source FSL S_{fsl} task, selecting C classes from the source classes C_s to form an episode. In the source episode, source data $\boldsymbol{X_s}$ is divided into a support set $S_s = \{(x_i^s, y_i^s)\}_{i=1}^{C\times K}$ and a query set $Q_s = \{(x_j^s, y_j^s))\}_{j=1}^{C\times N}$. Specifically, C categories are randomly selected from $\boldsymbol{X_s}$, with K samples from each category, forming a support set. Moreover, a query set is formed by randomly selecting N samples from the same C classes that are distinct to those in the support set. After that, the distribution aligner is first applied for dimensionality reduction of all samples in the support and query sets, after which the embedding characteristics are obtained by the feature extractor. FSL is executed by comparing the similarity of the embedded features between the query and support sets per category. The class prototype for a support sample x_i^s in the support set S_s is

$$c_k = \frac{1}{|S_s^k|}\sum_{(x_i^s, y_i^s)\in S_s^k} f_\varphi(x_i^s) \tag{2}$$

where S_s^k is the set belonging to class k in the support set, $|S_s^k|$ is the number of samples in S_s^k, x_i^s denotes a support set sample for which the label is y_i^s, and f_φ indicates the feature extractor with argument φ. A query sample x_j^s in Q_s has the category distributivity computed by the Bregman divergences (i.e., the Euclidean distance) based on a softmax function

$$P_\varphi(y_j^s = k|x_j^s \in Q_s) = \frac{\exp(-ED(\,f_\varphi(x_j^s), c_k))}{\sum_{k=1}^{C}\exp(-ED(\,f_\varphi(x_j^s), c_k))} \tag{3}$$

where x_j^s represents a support set sample for which the label is y_j^s, $ED(\bullet)$ denotes a Euclidean distance function, C denotes the amount of distinct categories per episode. The source FSL loss of $x^s_j \in Q_s$ is calculated into the negative log-probability of its corresponding truth category by cross-entropy loss

$$L^s_{fsl} = -\log P_\varphi(y^s_j = k|x^s_j \in Q_s) = ED(f_\varphi(x^s_j), c_k) + \log\sum_{k=1}^{C}\exp(-ED(f_\varphi(x^s_j), c_k)). \tag{4}$$

2) Target FSL: Similar to the source FSL task, C classes are selected from target classes C_t to form an episode in the target FSL. In the target episode, target data $\boldsymbol{X_t}$ is similarly divided into a support set $S_t = \{(x^t_i, y^t_i)\}^{C\times K}_{i=1}$ and a query set $Q_t = \{(x^t_j, y^t_j)\}^{C\times N}_{j=1}$. Notice the support set samples are selected from labeled data D_f with only a few samples. Therefore, the class prototype for a support sample x^t_i in the support set S_t is

$$c_k = \frac{1}{S_t}\sum_{(x^t_i, y^t_i)\in S_t} f_\varphi(x^t_i). \tag{5}$$

The class predicted probability for a query sample x^t_j in Q_t expressed as

$$P_\varphi(y^t_j = k|x^t_j \in Q_t) = \frac{\exp(-ED(f_\varphi(x^t_j), c_k))}{\sum_{k=1}^{C}\exp(-ED(f_\varphi(x^t_j), c_k))}. \tag{6}$$

The target FSL loss of $x^t_j \in Q_t$ is given by

$$L^t_{fsl} = -\log P_\varphi(y^t_j = k|x^t_j \in Q_t) = ED(f_\varphi(x^t_j), c_k) + \log\sum_{k=1}^{C}\exp(-ED(f_\varphi(x^t_j), c_k)). \tag{7}$$

B. Distribution Aligner

The heterogeneity of feature distribution between the source and target domains resulted in inconsistent spectral resolutions of the samples. Thus, a distribution aligner is employed for mapping the source (the Chikusei dataset with 128 bands) and target domains (e.g., the Indian Pines dataset with 200 bands) to the same dimension d. The distribution aligner is implemented via 2-D CNN. First, we ensure the rationality of the selected band by selecting 9×9 neighborhoods to be the input spatial dimensions. Thus, assuming that $I \in R^{9\times 9\times b}$ is the input of the HSI cube where b means the bands amount, the result obtained from the distribution aligner as

$$\boldsymbol{I_A} = \boldsymbol{I} \times \boldsymbol{A} \tag{8}$$

where $\boldsymbol{I_A} \in R^{9\times 9\times 100}$ is the aligned dataset, and $\boldsymbol{A} \in R^{b\times 100}$ is the function of the distribution aligner. $b \times 100$ denotes learnable parameters in the alignment. There are 128×100 parameters for $\boldsymbol{X_s}$, and 200×100 parameters for $\boldsymbol{X_t}$.

C. Feature Extractor

The feature extractor works for extracting the spatial-spectral embedding features and mapping them to a scene-consistency metric space. The feature extractor is based on a convolutional transformer (CT) network, which effectively combines the convolutional neural network (CNN) with a vision transformer (ViT) structure and can extract both the local and global features for using the spatial and spectral information sufficiently. The feature extractor mainly consists of two subblocks, Fig. 3 shows the architecture of the feature extractor (see the feature extractor module). Specifically, Fig. 4 shows the CT module of the feature extractor.

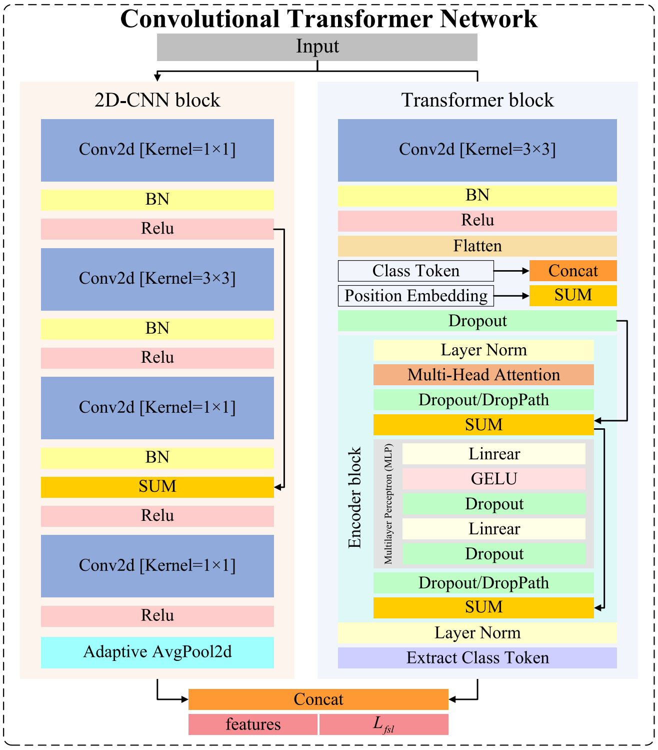


Fig. 4. CT module of the feature extractor.

The input to the feature extractor is the output $\boldsymbol{I_A} \in R^{9\times 9\times 100}$ from the distribution aligner. In our method, the input patch $\boldsymbol{I_A}$ is fed into the CT module which consisted of a CNN block and a ViT block. The CNN block extracts local features f_c from $\boldsymbol{I_A}$ and the ViT block is utilized to extract global features f_v. Then, we combined the local and global features to form the feature representation f of the feature extractor

$$f = \text{concat}(f_c, f_v). \tag{9}$$

D. Domain Discriminator

To reduce domain shift as inspired by [40], a domain discriminator is explored with adversarial loss to predict the domain to which a feature belongs. The domain discriminator is built on a fully convolutional network (FCN) that contains a convolutional layer with a 5×5 kernel as a filter, a convolutional layer with a 1×1 kernel, a residual block, followed by a final convolutional layer with a 1×1 kernel. Except for the last layer, each convolutional layer is followed by a batch normalization (BN) and a rectified linear unit (ReLU) nonlinear activation function.

Fig. 3 shows the architecture of the domain discriminator (see the domain discriminator module).

Our goal is classifying whether the features come from the source or target domain. On the domain discriminator, we define an adversarial loss function L_D to resolve the imbalance among classes while the loss L_D should be minimized

$$L_D = -\sum_{i \in \boldsymbol{I_A}} \log D\left(f_\theta\left(x_i^s\right)\right) + \log\left(1 - D\left(f_\theta\left(x_i^t\right)\right)\right) \quad (10)$$

where $D(\cdot)$ and $1 - D(\cdot)$ are the probabilities of a sample i belonging to the source and target domains predicted by the domain discriminator, respectively. f_θ denotes the features from the feature extractor with parameter θ, x_i^s and x_i^t are samples from the source and target domains (i.e., $x_i^s \in \boldsymbol{X_s}$, $x_i^t \in \boldsymbol{X_t}$), respectively.

Thus, the source domain's total loss function as

$$L^s = L_{fsl}^s + L_D. \quad (11)$$

Likewise, the target domain's total loss function as

$$L^t = L_{fsl}^t + L_D. \quad (12)$$

Finally, the nearest neighbor (NN) method is utilized to classify unlabeled samples in the target domain during the testing phase and then generate their classification maps to evaluate the effectiveness of CTFSL.

IV. Experimental Results

The experiments are performed using software platform Pycharm on a 12th Gen Intel Core$^{\text{TM}}$ i9-12900KF processor equipped with NVidia GeForce$^{\text{TM}}$ RTX 3090 Ti and 64 GB of RAM, and all codes executed on Python 3.7.

A. Experimental Data

The proposed CTFSL approach for cross-domain HSI classification is performed employing four public HSI datasets, namely, the Chikusei, Indian Pines, University of Pavia, and Salinas datasets.

1) Source Domain: The source domain dataset utilizes the Chikusei dataset. The Chikusei dataset was gathered over agricultural and urban areas in Chikusei, Ibaraki, Japan by a Headwall Hyperspec-VNIR-C imaging sensor, on July 29, 2014 [58]. It comprises 128 spectral bands with a spectrum of 363–1018 nm, comprises 2517 × 2335 pixels in which each has a spatial resolution of 2.5 m and comprises 19 unique land-cover categories. Fig. 5(a)–(c) presents the false-color image, the matching ground-truth map and the matching color card of the Chikusei. The classes of the Chikusei dataset and the corresponding sample numbers are shown in Table I.

2) Target Domain: The Indian Pines, University of Pavia, and Salinas datasets are applied as target domains. The Indian Pines dataset was acquired over the agricultural Indian Pine test site in North-western Indiana by an AVIRIS sensor in June 1992. It comprises 200 spectral bands with a spectrum of 400–2500 nm, comprises 145 × 145 pixels in which each has a spatial resolution of 20 m and it comprises 16 unique land-cover categories. Fig. 6(a)–(c) presents the false-color image, the matching ground-truth map and the matching color card of the Indian Pines. The classes of the Indian Pines dataset and the corresponding sample numbers are shown in Table II.

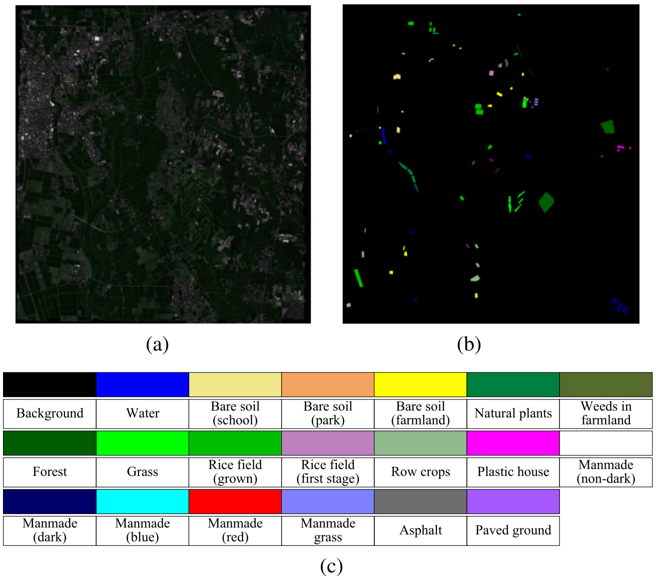


Fig. 5. Chikusei dataset. (a) False-color image. (b) Groundtruth map. (c) Color coding.

TABLE I
Class, Name, and Number of Samples on Chikusei Dataset

Class	Name	Samples
1	Water	2845
2	Bare soil (school)	2859
3	Bare soil (park)	286
4	Bare soil (farmland)	4852
5	Natural plants	4297
6	Weeds in farmland	1108
7	Forest	20516
8	Grass	6515
9	Rice field (grown)	13369
10	Rice field (first stage)	1268
11	Row crops	5961
12	Plastic house	2193
13	Manmade (non-dark)	1220
14	Manmade (dark)	7664
15	Manmade (blue)	431
16	Manmade (red)	222
17	Manmade grass	1040
18	Asphalt	801
19	Paved ground	145
	Total	77,592

The University of Pavia dataset was acquired over Pavia, Nothern Italy utilizing the ROSIS sensor in a flight campaign. It comprises 103 spectral bands with a spectrum of 430–860 nm, comprises 610 × 340 pixels in which each has a spatial resolution of 1.3 m and it comprises nine unique land-cover categories. Fig. 7(a)–(c) presents the false-color image, the matching ground-truth map and the matching color card of the University of Pavia. The classes of the University of Pavia dataset and the corresponding sample numbers are shown in Table III.

The Salinas dataset was gathered over Salinas Valley, California using AVIRIS sensor. It comprises 204 spectral bands

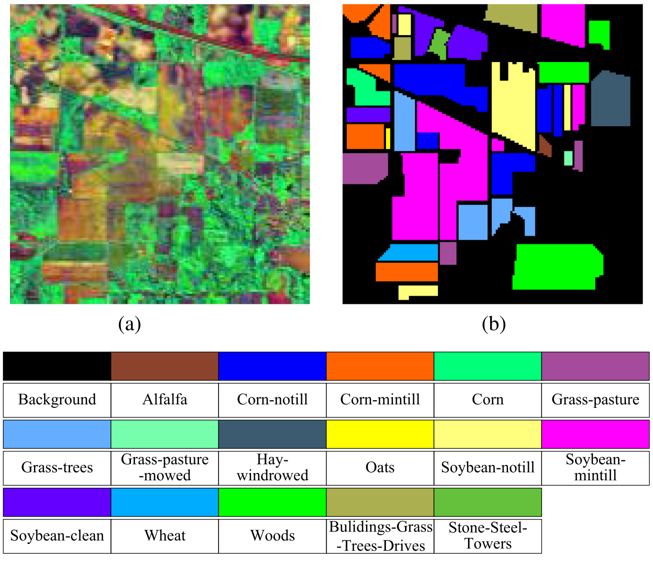


Fig. 6. Indian Pines dataset. (a) False-color image. (b) Groundtruth map. (c) Color coding.

TABLE II
CLASS, NAME, AND NUMBER OF SAMPLES ON INDIAN PINES DATASET

Class	Name	Samples
1	Alfalfa	46
2	Corn-notill	1428
3	Corn-mintill	830
4	Corn	237
5	Grass-pasture	483
6	Grass-trees	730
7	Grass-pasture-mowed	28
8	Hay-windrowed	478
9	Oats	20
10	Soybean-notill	972
11	Soybean-mintill	2455
12	Soybean-clean	593
13	Wheat	205
14	Woods	1265
15	Buildings-Grass-Trees-Drives	386
16	Stone-Steel-Towers	93
	Total	10,249

with a spectrum of 400–2500 nm, comprises 512 × 217 pixels in which each has a spatial resolution of 3.7 m and it comprises 16 unique land-cover categories. Fig. 8 presents the false-color graph, matching ground-truth map, and matching color card of Salinas. The classes of the Salinas dataset and the corresponding sample numbers are shown in Table IV.

B. Experimental Setup

The input of the proposed CTFSL is chosen from set with patch size $\{5\times5,\ 7\times7,\ 9\times9,\ 11\times11,\ 13\times13,\ 15\times15\}$. From Fig. 9, for all experiments on different target domains, we observe that with increasing input patch size, the classification accuracy also increases, but it will decrease after increasing beyond a certain extent and it approximately obeys the Gaussian distribution. Therefore, taking this into account, our method sets

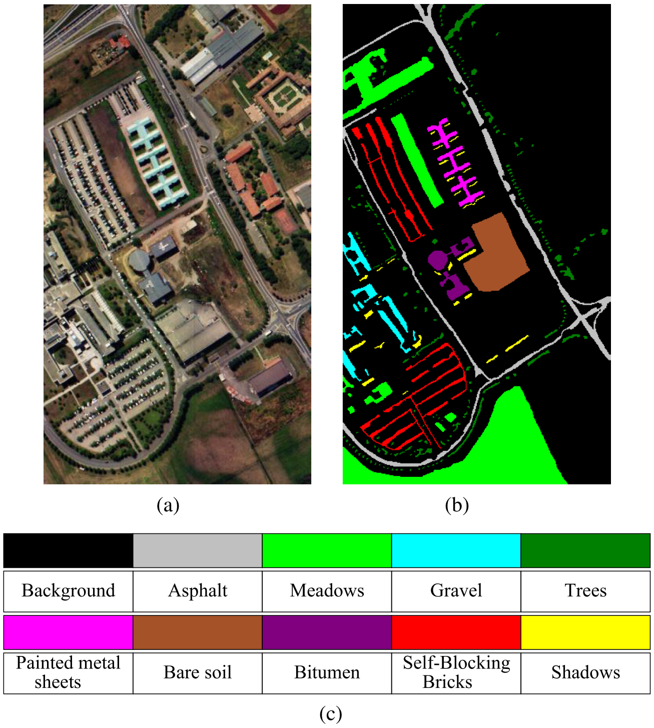


Fig. 7. University of Pavia dataset. (a) False-color image. (b) Groundtruth map. (c) Color coding.

TABLE III
CLASS, NAME, AND NUMBER OF SAMPLES ON UNIVERSITY OF PAVIA DATASET

Class	Name	Samples
1	Asphalt	6631
2	Meadows	18649
3	Gravel	2099
4	Trees	3064
5	Painted metal sheets	1345
6	Bare Soil	5029
7	Bitumen	1330
8	Self-Blocking Bricks	3682
9	Shadows	947
	Total	42,776

TABLE IV
CLASS, NAME, AND NUMBER OF SAMPLES ON SALINAS DATASET

Class	Name	Samples
1	Brocoli_green_weeds_1	2009
2	Brocoli_green_weeds_2	3726
3	Fallow	1976
4	Fallow_rough_plow	1394
5	Fallow_smooth	2678
6	Stubble	3959
7	Celery	3579
8	Grapes_untrained	11271
9	Soil_vinyard_develop	6203
10	Corn_senesced_green_weeds	3278
11	Lettuce_romaine_4wk	1068
12	Lettuce_romaine_5wk	1927
13	Lettuce_romaine_6wk	916
14	Lettuce_romaine_7wk	1070
15	Vinyard_untrained	7268
16	Vinyard_vertical_trellis	1807
	Total	54,129

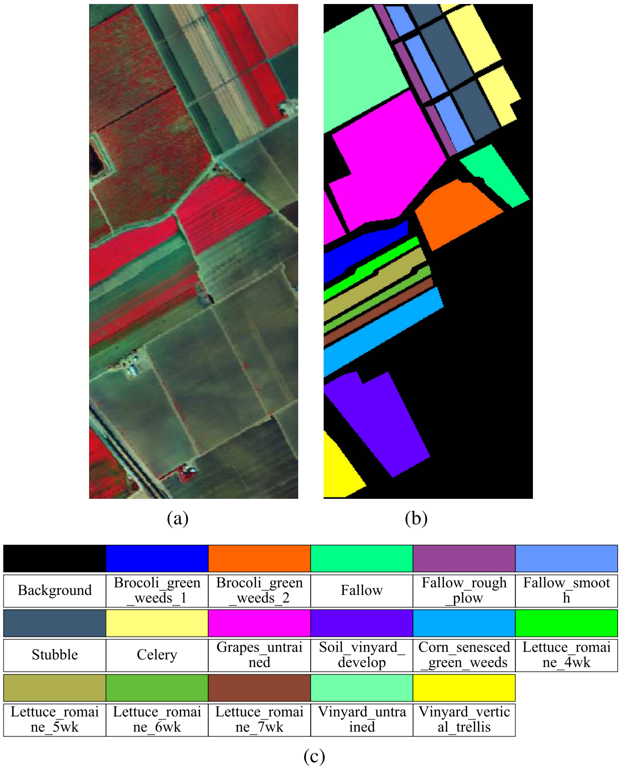


Fig. 8. Salinas dataset. (a) False-color image. (b) Groundtruth map. (c) Color coding.

the input patch size to 9×9. The CTFSL method is trained via an Adaptive Moment Estimation (Adam) optimizer. The training iterations are setup as 10 000 and the learning rate as 1e-3. In the episodic training phase, each episode represents a C-way K-shot mission. C indicates the count of categories and sets it as the class number in the target domain (i.e., setting the University of Pavia dataset as 9, the Indian Pines and Salinas datasets as 16). K indicates samples number per class within support set S and is always set to one regardless of source or target FSL. In addition, samples number per class within query set Q is N_Q, and N_Q is setup to 19 to evaluate the learned classifier. Furthermore, 200 labeled samples selected arbitrarily from each category of the source domain to acquire transferred knowledge. Finally, classification was based on a K-nearest neighbor (KNN) classifier, and the number of nearest neighbors is set to 1.

To verify the validity of the proposed CTFSL, we compared our method to some classical and cutting edge cross-scene methods used for HSI classification, containing typical KNN [59], which is one of the simplest classification methods, a common kernel-learning method SVM [9], 3-D-CNN [2], DFSL+NN [60], DFSL+SVM [60], and DCFSL [25].

To guarantee the equity of the above-mentioned approaches, five labeled samples per target domain category were first chosen for training within all control experiments. Then, adding random Gaussian noise to augment the data. The remaining entries in the target domain are regarded as the testing data. In addition, for cross-domain methods, learning portable information by 200 labeled samples selected randomly from each source domain class (DFSL+SVM, DFSL+NN, and DCFSL).

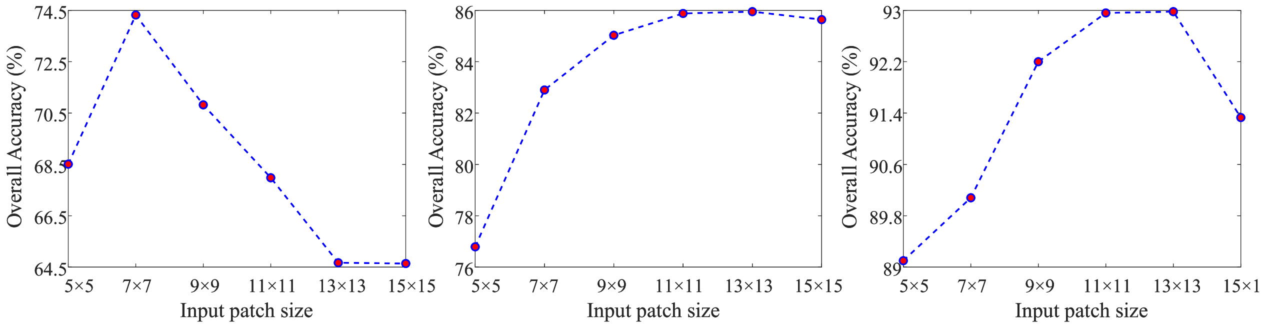
To assess the classification effects of different approaches objectively, we adopted three widely used quality indicators, the overall accuracy (OA), the average accuracy (AA), and the kappa coefficient. The training samples chosen at random for all experiments, for which reason ten repetitions were performed for eliminating the influences, and thus obtaining the means and standard deviations of OA, AA, and Kappa. In addition, the values reported for each metric were computed by taking the average of the outcomes derived from ten repetitive experiments with arbitrarily chosen training samples.

C. Comparison of Different Methods

Comparing the proposed CTFSL method with three typical classification methods (KNN, SVM, and 3-D-CNN), and three FSL classification approaches (DFSL+SVM, DFSL+NN, and DCFSL) show our method's advantages and efficiency. For the supervised methods (KNN, SVM, and 3-D-CNN), training the classifier can only choose a few-shot data from the target domain. The reason why source domain samples cannot be used as a training set in these methods is that they demand the same training as the test categories. In particular, KNN calculates the Euclidean distances between test and training samples of distinct categories, and obtains the class to which the test sample belongs by comparing the average of the smallest Euclidean distances. It is noteworthy that the number of nearest neighbors is set to 1. SVM learns nonlinear support vector machine by kernel method to map nonlinear data into a linearly separable space, but the standard SVM method ignores the spatial information, focusing only on spectral information in HSI. The 3-D-CNN method enables effective extraction of deep spectral-spatial characteristics that contribute to the accurate classification of HSI.

Nevertheless, in the case of the FSL approaches (DFSL+SVM, DFSL+NN, and DCFSL), the samples in the source domain can be utilized to learn transferable knowledge since the classes may differ between the source and target domains. Concretely, learning metric space in DFSL+SVM and DFSL+NN methods to extract spectral-special features via a deep residual 3D CNN, and then, such metric space could be used in few-shot classification with a SVM or NN classifier. The DCFSL model is based on the DFSL+NN and DCFSL, which construct a unified structure to address FSL and domain adaptation problems. With the suggested CTFSL scheme, the aforementioned default arguments are applied for all experiments.

To confirm the effectiveness of suggested CTFSL, experiments on three datasets are compared to the foregoing comparison approaches. The first executed on the Indian Pines dataset. Comparing the different methods' performances, 5 labeled items per category were randomly sampled from the Indian Pines dataset. To objectively evaluate the performances of the different methods, 10 classification experiments were repeated for eliminating the influence from from stochastic sampling. The classification performance of all methods was assessed employing the mean and standard variance of the OA, AA, and Kappa coefficients. Specifically, the optimal values for each class are bolded to highlight, and the values in parentheses refer to the standard



(a) (b) (c)

Fig. 9. Influence of the input patch size on the performance of the proposed CTFSL method on different datasets. (a) Indian Pines dataset. (b) University of Pavia dataset. (c) Salinas dataset.

TABLE V
Cross-Scene Classification Performance [%] of Different Methods on Indian Pines With Five Labeled Samples Per Class

Class	KNN	SVM	3D-CNN	DFSL+NN	DFSL+SVM	DCFSL	CTFSL
1	60.98(2.44)	69.51(1.22)	40.24(4.40)	92.68(11.3)	92.68(12.0)	96.59(5.37)	**100.00(0.0)**
2	31.59(5.59)	29.87(2.81)	37.57(5.04)	**48.10(9.60)**	45.23(12.0)	37.69(7.03)	38.44(9.57)
3	37.82(3.15)	45.33(8.48)	31.26(3.55)	47.38(6.76)	44.67(10.8)	46.28(8.36)	**56.70(3.52)**
4	51.72(6.90)	58.62(0.86)	41.51(7.93)	77.76(19.1)	75.22(20.8)	**83.49(6.12)**	74.70(15.2)
5	60.36(16.2)	60.36(26.1)	20.48(2.30)	74.06(7.18)	73.81(9.84)	72.99(5.09)	**85.33(3.45)**
6	68.34(11.7)	73.38(14.1)	69.17(3.99)	85.53(7.81)	83.90(7.80)	83.16(12.8)	**91.27(3.56)**
7	91.30(0.00)	91.30(0.00)	**100.00(0.0)**	99.57(1.30)	98.70(1.99)	99.13(1.74)	**100.00(0.0)**
8	77.70(6.03)	76.85(11.1)	86.28(3.72)	83.97(12.8)	85.20(12.9)	86.98(8.78)	85.90(10.0)
9	73.33(6.67)	90.00(3.33)	**100.00(0.0)**	**100.00(0.0)**	99.33(2.00)	**100.00(0.0)**	**100.00(0.0)**
10	42.66(2.22)	48.55(1.19)	33.26(3.75)	61.84(8.36)	58.10(6.67)	65.67(2.25)	**68.63(3.22)**
11	35.82(1.69)	41.29(2.35)	50.89(2.37)	59.14(12.5)	61.49(9.52)	66.61(2.14)	**74.44(4.07)**
12	34.61(11.7)	33.67(16.7)	22.81(7.11)	39.63(8.38)	43.25(8.96)	43.54(6.78)	**43.91(7.18)**
13	88.25(1.25)	88.75(5.75)	95.40(1.11)	97.50(4.28)	97.40(3.42)	99.60(0.70)	**99.70(0.51)**
14	60.56(36.1)	65.00(32.1)	90.58(3.99)	80.77(10.8)	79.51(10.4)	**90.94(3.06)**	89.35(3.63)
15	21.13(5.12)	25.85(0.66)	61.71(12.7)	68.40(14.6)	69.71(10.4)	72.18(8.78)	**79.74(7.15)**
16	89.77(2.27)	89.77(2.27)	85.91(3.64)	98.75(2.41)	98.75(3.07)	**98.52(1.35)**	95.00(6.23)
OA(%)	46.05±3.28	49.51±1.56	52.08±0.68	64.34±3.23	63.90±3.16	66.69±1.11	**70.82±1.30**
AA(%)	57.87±0.81	61.76±0.09	60.44±1.23	75.94±2.13	75.43±2.38	77.71±1.46	**80.19±1.60**
Kappa	39.97±3.41	43.76±1.63	46.02±0.77	59.92±3.40	59.34±3.44	62.41±1.27	**66.85±1.47**

The numbers in parentheses represent the standard deviation of the accuracy obtained from repeated experiments.

deviation of the precisions achieved from ten experimentations. Table V shows the classification accuracy of every category for Indian Pines under different methods. The cross-domain FSL approaches (DFSL+SVM, DFSL+NN, DCFSL, and CTFSL) are clearly superior to those traditional classification approaches (KNN, SVM, and 3-D-CNN) in the case of limited methods. In particular, the proposed CTFSL's OA, AA, and Kappa value are at least 4.13, 2.48, and 4.44 percentage points higher than the comparison method, respectively, which indicates that the CTFSL method is generally feasible. To visually demonstrate the proposed CTFSL's effectiveness, Fig. 10 shows a corresponding classification map of all the aforementioned methods. As shown in the figure, the proposed CTFSL can see some noise, but in contrast, it shows a classification map still with the smoothest spatial distribution and it has the best precision with less mislabeling, which are the concordant outcomes with Table V.

The second among them conducted on the University of Pavia dataset. Table VI displays the OA, AA, and Kappa coefficient, and the detailed classification accuracies of each class on the University of Pavia with various classification approaches. As Table VI shows, the KNN, SVM, and 3-D-CNN classification methods only consider limited target domain samples to develop the training data, so OA values are only 60.48%, 65.08%, and 69.87%. By contrast, the OAs of the cross-domain FSL-based classification approaches (DFSL+SVM, DFSL+NN, DCFSL, and CTFSL) are usually greater than 78%, because they can make full use of the source domain information and the target few-shot labeled information. In addition, comparing the DCFSL approach, the OA with our suggested approach increased from 83.83% to 85.03%, which proves this method's effectiveness. As an instance, the classification precision has improved from 74.46% to 80.24% for Class 6 and from 56.62% to 90.75% for Class 8 by comparison to DCFSL. The AA and Kappa of CTFSL are also the highest among all the compared classification methods. Fig. 11 shows the classification result maps under different methods. In particular, the classification graph of CTFSL clearly demonstrates its classification advantages compared to other methods.

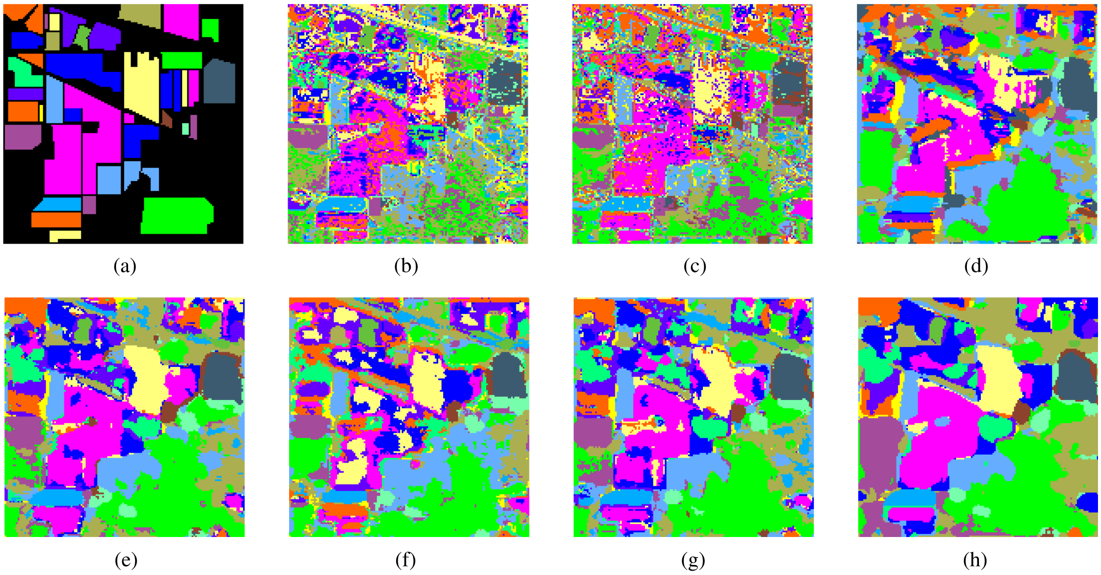


Fig. 10. Classification maps obtained by different classification methods on the Indian Pines image dataset with five labeled samples per class. (a) Reference map. (b) KNN. (c) SVM. (d) 3-D-CNN. (e) DFSL+NN. (f) DFSL+SVM. (g) DCFSL. (h) CTFSL.

TABLE VI
CROSS-SCENE CLASSIFICATION PERFORMANCE [%] OF DIFFERENT METHODS ON UNIVERSITY OF PAVIA WITH FIVE LABELED SAMPLES PER CLASS

Class	KNN	SVM	3D-CNN	DFSL+NN	DFSL+SVM	DCFSL	CTFSL
1	64.89(5.28)	62.74(9.03)	50.58(4.21)	81.57(6.08)	81.32(10.4)	91.22(2.02)	**92.74(2.05)**
2	52.03(9.61)	62.45(11.2)	90.72(1.77)	76.46(10.5)	76.77(9.68)	**90.39(1.66)**	83.04(6.15)
3	45.52(10.6)	54.64(11.5)	61.55(6.15)	**75.35(9.65)**	69.31(10.1)	50.30(3.07)	53.46(6.11)
4	89.06(8.15)	88.87(6.95)	87.11(3.70)	93.19(4.09)	94.25(3.44)	**95.39(1.24)**	94.48(1.03)
5	98.99(0.64)	98.58(1.75)	89.52(3.21)	99.43(0.60)	99.33(0.93)	**99.60(0.71)**	99.04(1.41)
6	50.24(10.2)	49.16(14.5)	43.68(3.53)	78.69(13.7)	80.04(13.1)	74.46(5.79)	**80.24(11.8)**
7	80.99(12.1)	**81.22(8.65)**	16.57(2.31)	78.85(6.02)	73.32(7.85)	64.72(6.00)	80.48(3.35)
8	62.62(14.6)	63.60(12.8)	31.22(5.11)	59.07(11.9)	58.69(17.6)	56.62(8.97)	**90.75(1.98)**
9	**99.83(0.14)**	99.75(0.11)	93.09(1.42)	99.24(1.08)	99.13(0.98)	99.79(0.22)	99.37(0.24)
OA(%)	60.48±3.81	65.08±4.10	69.87±0.89	78.46±4.77	78.28±5.70	83.83±0.97	**85.03±2.02**
AA(%)	71.57±1.72	73.44±1.86	62.67±1.34	82.43±2.30	81.35±2.72	80.28±1.66	**85.96±1.54**
Kappa	51.48±3.87	56.24±4.14	59.89±1.16	72.61±5.74	72.46±6.69	78.73±1.32	**80.64±2.39**

The numbers in parentheses represent the standard deviation of the accuracy obtained from repeated experiments.

The third carried out on the Salinas dataset had the analogous findings. Table VII shows the classification accuracy values yielded by the compared approaches and the suggested CTFSL. As an instance, compared to KNN, the classification precision has improved from 75.72% to 98.08% for Class 3, that of Class 8 has increased from 48.43% to 83.26%, and that of Class 15 has increased from 61.03% to 80.78%. Fig. 12 visually represents the proposed CTFSL's effectiveness by showing the corresponding classification maps yielded by all the aforementioned methods with the OAs. Apparently, the suggested CTFSL yields a classification map with the smoothest spatial distribution and it has the best precision with less mislabeling from Fig. 12, which are the concordant outcomes with Table VII.

To illustrate the proposed CTFSL's computational complexity effectively, Table VIII shows the computational efficiency (including training and testing times) of the above methods in different target domains. For three typical classification methods (KNN, SVM, and 3D-CNN) without a cross-domain, their training times are shorter than that of the other cross-domain FSL classification methods (DFSL+NN, DFSL+SVM, DCFSL, and CTFSL). The table shows that although our method takes a long time to train, it has the highest accuracy.

D. Parameter Analysis

To analyze the performance of the nearest neighbor size, the algorithm comparison experiments under different nearest neighbor size are carried out to analyze the sensitivity of the CTFSL algorithm on three target domain datasets. We set 1, 2, 3, 4, 5 as the size of the nearest neighbors to conduct 10 iterations of the experiment and obtain the average of the results to compare the performance, Table IX shows the classification accuracy for three datasets under different nearest neighbor size, where the best results are bolded to highlight. As can be seen from the results in the Table IX, the nearest neighbor size set to 1, 2, and 4 on the Indian Pines, University of Pavia, and Salinas datasets exhibit optimal classification performance, respectively.

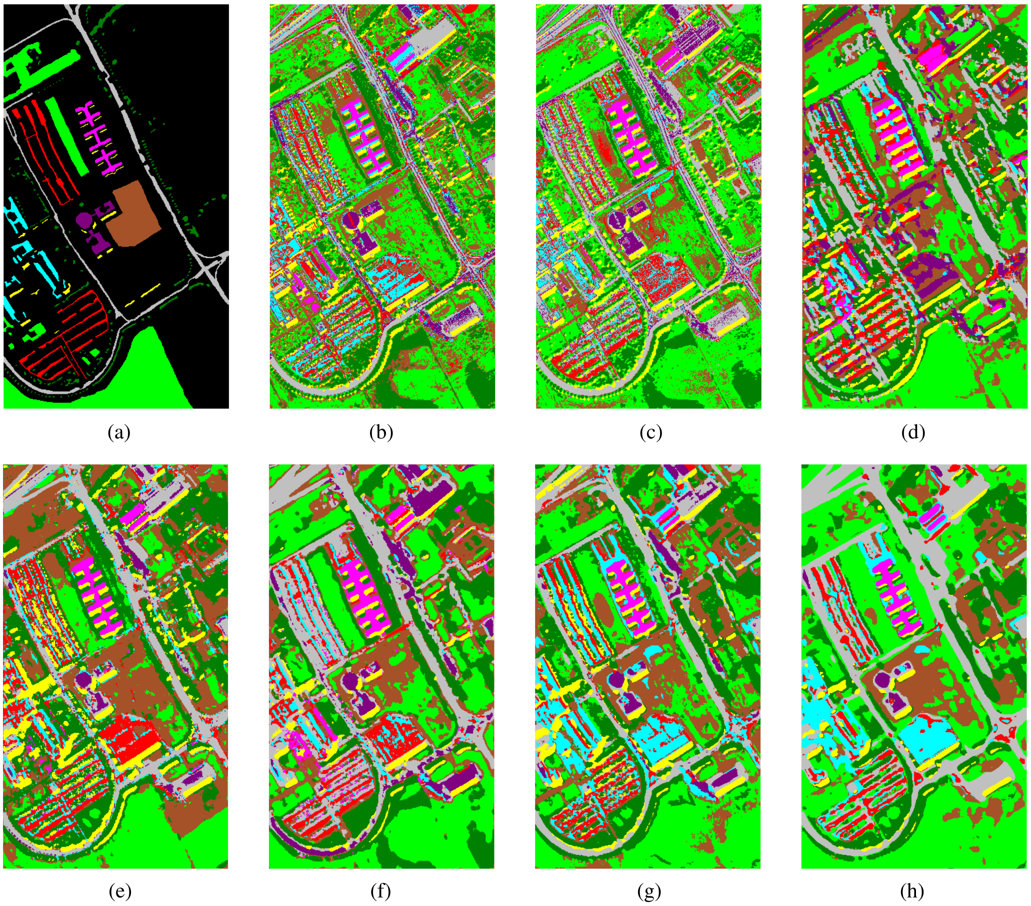


Fig. 11. Classification maps obtained by different classification methods on the University of Pavia image dataset with five labeled samples per class. (a) Reference map. (b) KNN. (c) SVM. (d) 3-D-CNN. (e) DFSL+NN. (f) DFSL+SVM. (g) DCFSL. (h) CTFSL.

TABLE VII
CROSS-SCENE CLASSIFICATION PERFORMANCE [%] OF DIFFERENT METHODS ON SALINAS WITH FIVE LABELED SAMPLES PER CLASS

Class	KNN	SVM	3D-CNN	DFSL+NN	DFSL+SVM	DCFSL	CTFSL
1	96.32(1.45)	97.36(2.17)	65.08(14.9)	99.48(0.66)	**99.54(0.79)**	99.53(0.57)	99.25(0.73)
2	93.27(3.87)	94.73(3.87)	79.54(10.1)	**99.00(1.51)**	98.64(2.13)	98.97(2.14)	98.69(1.86)
3	75.72(5.04)	82.98(6.76)	61.18(4.25)	89.24(12.1)	87.54(12.9)	96.24(2.06)	**98.08(2.52)**
4	99.37(0.25)	99.37(0.25)	52.78(3.10)	99.67(0.20)	99.65(0.12)	99.66(0.23)	**99.74(0.28)**
5	95.79(1.05)	**97.43(0.82)**	87.68(3.15)	90.67(3.32)	90.77(3.80)	92.66(2.53)	94.10(1.31)
6	98.31(1.14)	98.60(0.85)	93.52(4.56)	99.48(0.52)	99.48(0.50)	**99.87(0.26)**	99.56(0.76)
7	99.27(0.07)	99.43(0.08)	78.68(3.20)	98.92(1.63)	98.77(1.83)	**99.64(0.29)**	99.08(0.86)
8	48.43(8.27)	53.00(13.5)	68.54(8.33)	73.33(12.7)	73.47(9.56)	77.77(4.23)	**83.26(4.61)**
9	93.49(1.29)	95.30(0.54)	89.13(1.54)	99.22(1.04)	99.16(1.10)	**99.77(0.30)**	98.84(1.63)
10	78.59(5.67)	81.63(2.94)	38.21(6.53)	86.65(5.07)	86.89(5.47)	**89.46(3.34)**	89.04(3.02)
11	92.37(2.10)	93.52(1.52)	83.48(3.76)	98.38(1.40)	97.80(1.98)	**99.23(0.61)**	98.65(1.61)
12	95.30(4.05)	**99.56(0.32)**	75.30(9.12)	99.47(0.64)	99.44(0.65)	99.46(0.88)	99.38(0.68)
13	94.96(5.55)	94.86(5.49)	98.30(0.61)	99.17(1.14)	98.80(1.55)	99.17(0.79)	**99.54(0.53)**
14	91.15(2.69)	90.64(2.38)	92.57(5.12)	97.81(2.81)	97.86(2.57)	98.70(0.71)	**99.31(0.27)**
15	61.03(7.58)	59.86(10.4)	51.75(9.74)	76.63(7.81)	76.11(6.42)	77.89(5.13)	**80.78(8.49)**
16	64.00(5.24)	77.29(12.6)	50.62(7.35)	90.88(6.41)	90.21(7.76)	**94.27(4.92)**	91.66(8.06)
OA(%)	78.31±2.23	80.62±3.65	71.12±1.21	88.93±2.37	88.77(2.06)	90.84±0.65	**92.20±0.52**
AA(%)	86.09±0.43	88.47±0.66	72.90±1.14	93.63±1.22	93.38(1.27)	95.14±0.50	**95.56±0.58**
Kappa	76.02±2.40	78.56±3.98	67.79±1.30	87.72±2.59	87.53(2.27)	89.82±0.72	**91.33±0.58**

The numbers in parentheses represent the standard deviation of the accuracy obtained from repeated experiments.

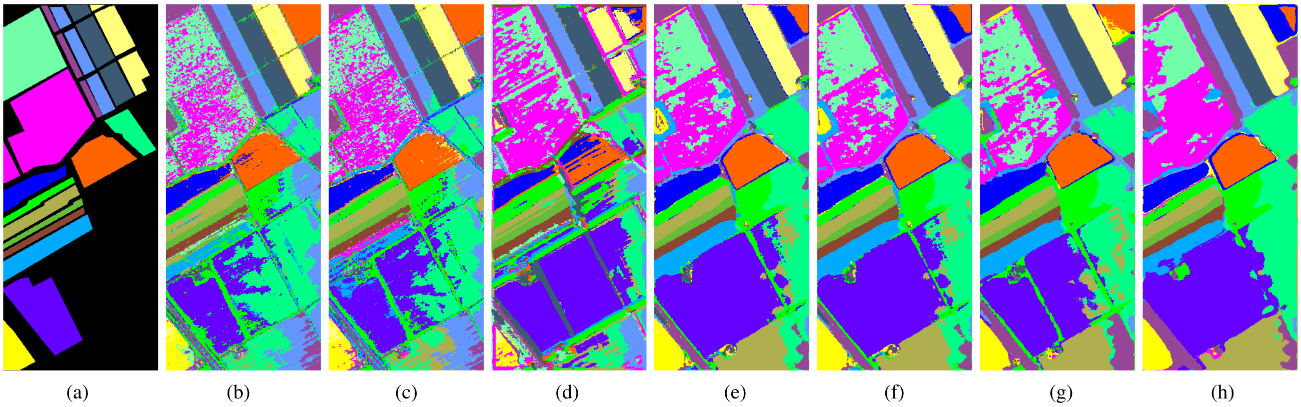


Fig. 12. Classification maps obtained by different classification methods on the Salinas image dataset with five labeled samples per class. (a) Reference map. (b) KNN. (c) SVM. (d) 3-D-CNN. (e) DFSL+NN. (f) DFSL+SVM. (g) DCFSL. (h) CTFSL.

TABLE VIII
COMPUTATIONAL EFFICIENCY (IN SECONDS) OF THE DIFFERENT METHODS IN THREE TARGET DOMAINS

Methods	Indian Pines		University of Pavia		Salinas	
	Training time(s)	Testing time(s)	Training time(s)	Testing time(s)	Training time(s)	Testing time(s)
KNN	0.27	5.70	1.60	14.69	1.44	29.68
SVM	0.27	0.50	1.61	1.71	1.43	4.66
3D-CNN	258.64	1.94	504.85	4.52	1444.29	10.94
DFSL+NN	1685.34	1.12	913.96	4.18	1495.94	5.72
DFSL+SVM	1506.03	1.05	726.88	2.08	1832.97	5.76
DCFSL	1758.95	0.70	1257.21	4.24	1818.64	5.72
CTFSL	4139.07	1.10	2723.48	3.98	4134.60	5.37

TABLE IX
CLASSIFICATION PERFORMANCE OF THE PROPOSED CTFSL METHOD WITH DIFFERENT NEAREST NEIGHBOR SIZE ON THE THREE HYPERSPECTRAL IMAGE DATASETS

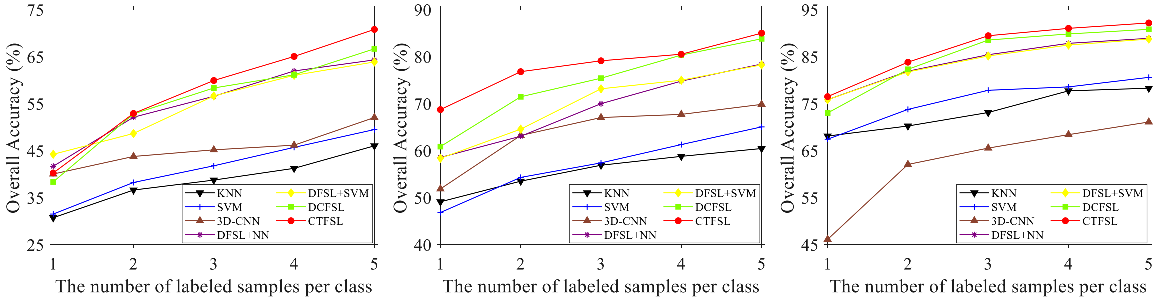
The Nearest Neighbor Size	Indian Pines			University of Pavia			Salinas		
	OA (%)	AA (%)	Kappa	OA (%)	AA (%)	Kappa	OA (%)	AA (%)	Kappa
1	**70.82±1.30**	**80.19±1.60**	**66.85±1.47**	85.03±2.02	**85.96±1.54**	80.64±2.39	92.20±0.52	**95.56±0.58**	91.33±0.58
2	68.38±2.02	77.89±2.08	64.09±2.21	**86.26±3.01**	83.81±1.34	**81.81±3.84**	92.25±0.51	95.32±0.55	91.37±0.57
3	68.18±2.16	77.92±2.61	63.75±2.62	83.57±2.91	84.35±1.45	78.74±3.52	91.83±0.69	95.19±0.71	90.91±0.77
4	68.70±2.81	78.06±2.71	64.31±3.26	84.27±2.57	83.11±1.49	79.30±3.19	**92.37±0.88**	95.51±0.87	**91.50±0.98**
5	68.63±2.44	78.17±2.08	64.35±2.85	83.08±2.43	83.12±1.52	77.96±2.99	92.00±0.64	95.13±0.75	91.10±0.71

In the comparison experiments, we set the number of nearest neighbors to 1, which is not optimal for University of Pavia and Salinas, but still shows better performance than the other methods. Although it is best for Indian pines when the number of nearest neighbors is set to 1, it can be seen from Tables V and IX that there are still better classification results than other algorithms when the nearest neighbor size takes other values. This further proves the superiority of our method.

To investigate the effect of the labeled sample size on the CTFSL method performance, 1, 2, 3, 4, and 5 labeled samples were also randomly selected from for each class of the target domains to build few-shot data respectively. Then the classification experiments with different numbers of labeled samples were performed ten repetitions, and the classification accuracies for each number of labeled samples with previously mentioned methods under the Indian Pines, University of Pavia, and Salinas datasets are shown in Tables X–XII, where the best results are bolded to highlight. To illustrate this visually, Fig. 13 shows the classification accuracy curves of different labeled sample numbers on three target domain datasets. As shown in Fig. 13, the OAs of the classification results obtained by all the methods are closely related to the change in labeled values, the increased number of labeled samples, the higher classification accuracy, and using five labeled samples per class exhibits the best performance. In particular, the CTFSL method is superior to the other methods mentioned with the same labeled samples, which shows the superior stability of the CTFSL method.

E. Analysis of Practical Applications

To verify the effectiveness and superiority of CTFSL method in practical application scenarios, we conduct an experimental



(a) (b) (c)

Fig. 13. Influence of the number of labeled samples per class on the performance of the proposed CTFSL method on different datasets. (a) Indian Pines dataset. (b) University of Pavia dataset. (c) Salinas dataset.

TABLE X
CLASSIFICATION PERFORMANCE [%] OF DIFFERENT NUMBER OF LABELED SAMPLES PER CLASS ON INDIAN PINES (N IS THE NUMBER OF LABELED SAMPLES PER CLASS)

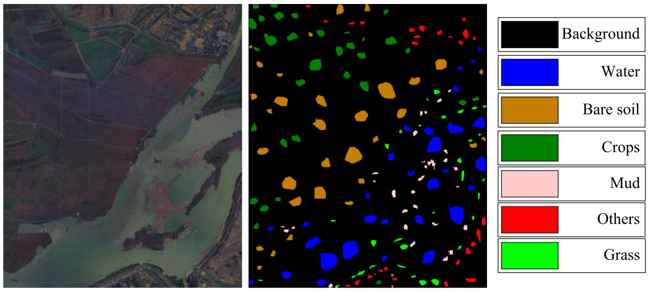
	KNN	SVM	3D-CNN	DFSL +NN	DFSL +SVM	DCFSL	CTFSL
N=1	30.75 ±2.60	31.53 ±2.04	40.05 ±0.83	41.70 ±5.57	**44.25** **±6.05**	38.38 ±4.26	40.27 ±2.91
N=2	36.64 ±3.77	38.25 ±3.73	43.79 ±1.44	52.14 ±3.54	48.68 ±5.96	52.84 ±4.54	**52.94** **±3.79**
N=3	38.76 ±3.53	41.80 ±4.15	45.20 ±0.92	56.59 ±2.81	56.66 ±4.62	58.36 ±3.86	**59.95** **±1.84**
N=4	41.26 ±3.70	45.72 ±2.52	46.10 ±0.32	61.98 ±3.45	61.05 ±2.92	61.16 ±4.10	**65.07** **±1.58**
N=5	46.05 ±3.28	49.51 ±1.56	52.08 ±0.68	64.34 ±3.23	63.90 ±3.16	66.69 ±1.11	**70.82** **±1.30**

TABLE XI
CLASSIFICATION PERFORMANCE [%] OF DIFFERENT NUMBER OF LABELED SAMPLES PER CLASS ON UNIVERSITY OF PAVIA (N IS THE NUMBER OF LABELED SAMPLES PER CLASS)

	KNN	SVM	3D-CNN	DFSL +NN	DFSL +SVM	DCFSL	CTFSL
N=1	49.14 ±9.03	46.86 ±10.44	51.88 ±6.73	58.58 ±7.22	58.40 ±3.46	60.89 ±5.55	**68.76** **±1.75**
N=2	53.53 ±8.43	54.33 ±7.02	63.26 ±4.31	63.09 ±6.21	64.60 ±4.59	71.49 ±2.21	**76.83** **±2.94**
N=3	56.92 ±3.67	57.42 ±5.65	67.07 ±1.81	70.02 ±6.96	73.19 ±5.06	75.44 ±4.35	**79.15** **±2.65**
N=4	58.81 ±5.41	61.32 ±4.37	67.74 ±3.27	74.82 ±4.57	75.01 ±4.76	80.36 ±3.25	**81.11** **±2.96**
N=5	60.48 ±3.81	65.08 ±4.10	69.87 ±0.89	78.46 ±4.77	78.28 ±5.70	83.83 ±0.97	**85.03** **±2.02**

TABLE XII
CLASSIFICATION PERFORMANCE [%] OF DIFFERENT NUMBER OF LABELED SAMPLES PER CLASS ON SALINAS (N IS THE NUMBER OF LABELED SAMPLES PER CLASS)

	KNN	SVM	3D-CNN	DFSL +NN	DFSL +SVM	DCFSL	CTFSL
N=1	68.13 ±4.11	67.46 ±3.82	46.14 ±0.94	75.87 ±2.86	75.87 ±2.86	73.04 ±2.03	**76.52** **±2.27**
N=2	70.27 ±3.13	73.80 ±2.94	62.09 ±2.45	81.93 ±2.38	81.78 ±2.46	82.33 ±2.47	**83.85** **±2.49**
N=3	73.14 ±3.67	77.87 ±1.92	65.57 ±1.51	85.44 ±2.60	85.17 ±2.63	88.56 ±1.57	**89.48** **±1.38**
N=4	77.75 ±2.13	78.59 ±2.27	68.43 ±2.52	87.86 ±1.94	87.50 ±1.72	89.86 ±1.29	**91.05** **±0.96**
N=5	78.31 ±2.23	80.62 ±3.65	71.12 ±1.21	88.93 ±2.37	88.77 ±2.06	90.84 ±0.65	**92.20** **±0.52**



(a) (b) (c)

Fig. 14. Application scenario dataset. (a) False-color image. (b) Groundtruth map. (c) Color coding.

analysis of HSI data for a scenario in the Dongting Lake Basin. The Dongting Lake Basin dataset was gathered by Hyper-Spectral Observation Satellite GaoFen (GF)-5 Advanced HyperSpectral Imager (AHSI) on December 8, 2019, it consists of 2008 × 2083 pixels with a spatial resolution of 30 m and 330 spectral bands in the wavelength range 400–2500 nm. GF-5 is the world's first hyperspectral satellite covering the full spectral range and enables comprehensive observation of the land and atmosphere. By processing GF-5 data from the Dongting Lake Basin, a scene with 452 × 380 pixels and 305 effective spectral bands was selected as the experimental dataset. The scene contains six different land-cover classes and 16 584 ground-truth labels, Fig. 14(a)–(c) shows the false-color image of the scene and the corresponding ground-truth map and the corresponding color code.

In the validation experiment, five labeled samples of each class are randomly selected for training in CTFSL and various comparison methods, and the rest are regarded as the testing data. The number of nearest neighbors is set to 1. The results show that the CTFSL method has more performance advantages with the highest classification accuracy that the OA value is 90.43%, compared to other methods. Fig. 15 shows the classification result maps and the corresponding average of OA values among ten repetitive experiments under different methods are

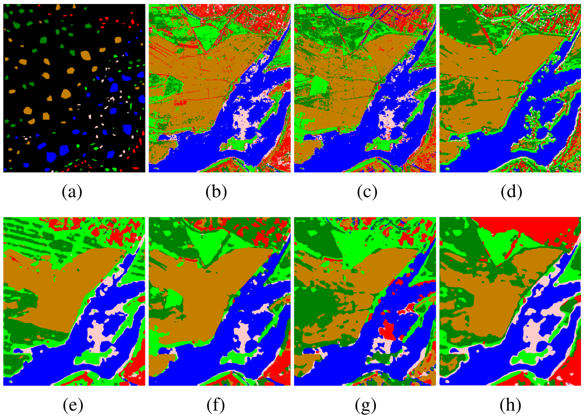


Fig. 15. Classification maps obtained by different classification methods on the practical HSI dataset with five labeled samples per class. (a) Reference map. (b) KNN (OA = 78.84%). (c) SVM (OA = 79.73%). (d) 3D-CNN (OA = 82.61%). (e) DFSL+NN (OA = 89.30%). (f) DFSL+SVM (OA = 89.09%). (g) DCFSL (OA = 88.54%). (h) CTFSL (OA = 90.43%).

in parentheses. As shown in the figure, the visualization map of CTFSL can see some noise, but in contrast, it still shows the most accurate and spatially smoothest classification map of classes with fewer mislabeled pixels, compared to other methods.

V. Conclusion

This article proposes convolutional transformer-based few-shot learning method for cross-domain hyperspectral image classification. The method includes three main parts: 1) distribution aligner based on few-shot learning to achieve the dimensionality reduction; 2) feature extractor based on convolutional transformer network to obtain the local-global features; 3) domain discriminator based on fully convolutional network to reduce the domain shift. Experiments have been performed on three different real hyperspectral images, and the results show that the proposed CTFSL outperformers the existing state-of-the-art FSL methods in cross-domain HSI classification, thus verifying its effectiveness. However, the good performance of the proposed CTFSL method relies on a relatively large computational cost. Further developments of this work should further improve its performance while reducing the computation time.

References

[1] N. He et al., “Feature extraction with multiscale covariance maps for hyperspectral image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 2, pp. 755–769, Feb. 2019.

[2] Y. Li, H. Zhang, and Q. Shen, “Spectral–spatial classification of hyperspectral imagery with 3D convolutional neural network,” *Remote Sens.*, vol. 9, no. 1, 2017, Art. no. 67.

[3] S. Jia et al., “Gradient feature-oriented 3-D domain adaptation for hyperspectral image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Apr. 2022, Art. no. 5505517.

[4] S. Jia et al., “3-D gabor convolutional neural network for hyperspectral image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Jun. 2022, Art. no. 5509216.

[5] S. Jia et al., “A semisupervised siamese network for hyperspectral image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Oct. 2022, Art. no. 5516417.

[6] B. Tu, Q. Ren, C. Zhou, S. Chen, and W. He, “Feature extraction using multidimensional spectral regression whitening for hyperspectral image classification,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 8326–8340, Aug. 2021.

[7] M. X. Sen Jia and Z. Zhan, “Shearlet-based structure-aware filtering for hyperspectral and lidar data classification,” *J. Remote Sens.*, vol. 2021, Art. no. 9825415.

[8] S. Jia, X. Deng, J. Zhu, M. Xu, J. Zhou, and X. Jia, “Collaborative representation-based multiscale superpixel fusion for hyperspectral image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 10, pp. 7770–7784, Oct. 2019.

[9] 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.

[10] Y. Zhang, W. Li, and R. Tao, “Domain adaptation based on graph and statistical features for cross-scene hyperspectral image classification,” in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2021, pp. 5374–5377.

[11] Y. Zhou, P. Chen, N. Liu, Q. Yin, and F. Zhang, “Graph-embedding balanced transfer subspace learning for hyperspectral cross-scene classification,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 2944–2955, Mar. 2022.

[12] 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. 11, pp. 9646–9660, Nov. 2021.

[13] H. Lee, S. Eum, and H. Kwon, “Cross-domain CNN for hyperspectral image classification,” in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2018, pp. 3627–3630.

[14] A. A. Bayanuddin et al., “Nadir vs. off-nadir: Initial look at LAPAN-A3 off-nadir acquisition mode on its spectral quality,” in *Proc. IEEE Int. Conf. Aerosp. Electron. Remote Sens. Technol.*, 2019, pp. 1–6.

[15] J. G. Masek et al., “A landsat surface reflectance dataset for north America, 1990-2000,” *IEEE Geosci. Remote Sens. Lett.*, vol. 3, no. 1, pp. 68–72, Jan. 2006.

[16] E. F. Vermote, D. Tanré, J. L. Deuze, M. Herman, and J.-J. Morcette, “Second simulation of the satellite signal in the solar spectrum, 6S: An overview,” *IEEE Trans. Geosci. Remote Sens.*, vol. 35, no. 3, pp. 675–686, May 1997.

[17] L. Bruzzone and D. F. Prieto, “Unsupervised retraining of a maximum likelihood classifier for the analysis of multitemporal remote sensing images,” *IEEE Trans. Geosci. Remote Sens.*, vol. 39, no. 2, pp. 456–460, Feb. 2001.

[18] 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.

[19] 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.*, early access, Sep. 2021, doi: 10.1109/TNNLS.2021.3109872.

[20] 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, Nov. 2021.

[21] H. Liu, W. Li, X.-G. Xia, M. Zhang, C.-Z. Gao, and R. Tao, “Spectral shift mitigation for cross-scene hyperspectral imagery classification,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 6624–6638, Jun. 2021.

[22] S. Jia, S. Jiang, Z. Lin, N. Li, M. Xu, and S. Yu, “A survey: Deep learning for hyperspectral image classification with few labeled samples,” *Neurocomputing*, vol. 448, pp. 179–204, Dec. 2021.

[23] D. Alajaji, H. S. Alhichri, N. Ammour, and N. Alajlan, “Few-shot learning for remote sensing scene classification,” in *Proc. Mediterranean Middle-East Geosci. Remote Sens. Symp.*, 2020, pp. 81–84.

[24] P.-C. Tu and H.-K. Pao, “A dropout style model augmentation for cross domain few-shot learning,” in *Proc. IEEE Int. Conf. Big Data*, 2021, pp. 1138–1147.

[25] Z. Li, M. Liu, Y. Chen, Y. Xu, W. Li, and Q. Du, “Deep cross-domain few-shot learning for hyperspectral image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Feb. 2021, Art. no. 5501618.

[26] Y. Zhang, W. Li, M. Zhang, and R. Tao, “Dual graph cross-domain few-shot learning for hyperspectral image classification,” in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2022, pp. 3573–3577.

[27] D. Hong et al., “SpectralFormer: Rethinking hyperspectral image classification with transformers,” *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Nov. 2021, Art. no. 5518615.

[28] X. He, Y. Chen, and Z. Lin, “Spatial-spectral transformer for hyperspectral image classification,” *Remote Sens.*, vol. 13, no. 3, 2021, Art. no. 498.

[29] A. Dosovitskiy et al., “An image is worth 16x16 words: Transformers for image recognition at scale,” in *Proc. Int. Conf. Learn. Representations*, Apr. 2020.

[30] X. Chen, S.-I. Kamata, and W. Zhou, "Hyperspectral image classification based on multi-stage vision transformer with stacked samples," in *Proc. IEEE Region 10 Conf.*, 2021, pp. 441–446.

[31] Y. Wu, J. Feng, G. Bai, Q. Gao, and X. Zhang, "Hyperspectral image classification based on spectrally-enhanced and densely connected transformer model," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2022, pp. 2746–2749.

[32] J. Feng, X. Luo, S. Li, Q. Wang, and J. Yin, "Spectral transformer with dynamic spatial sampling and gaussian positional embedding for hyperspectral image classification," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2022, pp. 3556–3559.

[33] Y. Peng, Y. Zhang, B. Tu, Q. Li, and W. Li, "Spatial–spectral transformer with cross-attention for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Sep. 2022, Art. no. 5537415.

[34] R. Gopalan et al., "Domain adaptation for visual recognition," *Foundations Trends Comput. Graph. Vis.*, vol. 8, no. 4, pp. 285–378, 2015.

[35] J. Zheng et al., "A two-stage adaptation network (TSAN) for remote sensing scene classification in single-source-mixed-multiple-target domain adaptation (s^{2}m^{2}t da) scenarios," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Aug. 2021, Art. no. 5609213.

[36] Y. Chen, C. Yang, Y. Zhang, and Y. Li, "Conditional adaptation deep networks for unsupervised cross domain image classifcation," in *Proc. 14th IEEE Conf. Ind. Electron. Appl.*, 2019, pp. 517–521.

[37] R. Gopalan, R. Li, and R. Chellappa, "Unsupervised adaptation across domain shifts by generating intermediate data representations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 11, pp. 2288–2302, Nov. 2014.

[38] Y. Fu, Y. Fu, and Y.-G. Jiang, "Meta-fdmixup: Cross-domain few-shot learning guided by labeled target data," in *Proc. 29th ACM Int. Conf. Multimedia*, 2021, pp. 5326–5334.

[39] J. Geng, X. Ma, W. Jiang, X. Hu, D. Wang, and H. Wang, "Cross-scene hyperspectral image classification based on deep conditional distribution adaptation networks," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2019, pp. 716–719.

[40] A. Tavera, F. Cermelli, C. Masone, and B. Caputo, "Pixel-by-pixel cross-domain alignment for few-shot semantic segmentation," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2022, pp. 1626–1635.

[41] N. Lai, M. Kan, C. Han, X. Song, and S. Shan, "Learning to learn adaptive classifier–predictor for few-shot learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 8, pp. 3458–3470, Aug. 2021.

[42] D. Chen, Y. Chen, Y. Li, F. Mao, Y. He, and H. Xue, "Self-supervised learning for few-shot image classification," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2021, pp. 1745–1749.

[43] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 4080–4090.

[44] Z. Li, F. Zhou, F. Chen, and H. Li, "Meta-SGD: Learning to learn quickly for few-shot learning," 2017, *arXiv:1707.09835*.

[45] M. Yan, "Adaptive learning knowledge networks for few-shot learning," *IEEE Access*, vol. 7, pp. 119041–119051, 2019.

[46] X. Sun, B. Wang, Z. Wang, H. Li, H. Li, and K. Fu, "Research progress on few-shot learning for remote sensing image interpretation," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 2387–2402, Jan. 2021.

[47] Y. Ding and P. Wang, "Reasearch on cross domain few-shot learning method based on local feature association," in *Proc. 6th Int. Symp. Comput. Inf. Process. Technol.*, 2021, pp. 754–759.

[48] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM Comput. Surv.*, vol. 53, no. 3, pp. 1–34, 2020.

[49] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1199–1208.

[50] K. Gao, B. Liu, X. Yu, J. Qin, P. Zhang, and X. Tan, "Deep relation network for hyperspectral image few-shot classification," *Remote Sens.*, vol. 12, no. 6, 2020, Art. no. 923.

[51] N. Parmar et al., "Image transformer," in *Proc. 35th Int. Conf. Mach. Learn.*, 2018, pp. 4055–4064.

[52] Z. Zhao, D. Hu, H. Wang, and X. Yu, "Convolutional transformer network for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, Apr. 2022, Art. no. 6009005.

[53] K. Han et al., "A survey on visual transformer," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 1, pp. 87–110, Jan. 2023.

[54] A. Vaswani et al., "Attention is all you need," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6000–6010.

[55] K. Han, A. Xiao, E. Wu, J. Guo, C. Xu, and Y. Wang, "Transformer in transformer," in *Proc. 35th Int. Conf. Neural Inf. Process. Syst.*, 2021, vol. 34, pp. 15908–15919.

[56] M. Popel and O. Bojar, "Training tips for the transformer model," 2018, *arXiv:1804.00247*.

[57] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations," 2018, *arXiv:1803.02155*.

[58] N. Yokoya and A. Iwasaki, "Airborne hyperspectral data over Chikusei," Space Appl. Lab., Univ. Tokyo, Tokyo, Japan, Tech. Rep. SAL-2016-05-27, May 2016.

[59] K. Huang, S. Li, X. Kang, and L. Fang, "Spectral–spatial hyperspectral image classification based on KNN," *Sens. Imag.*, vol. 17, no. 1, pp. 1–13, 2016.

[60] B. Liu, X. Yu, A. Yu, P. Zhang, G. Wan, and R. Wang, "Deep few-shot learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 4, pp. 2290–2304, Apr. 2018.

Yishu Peng (Member, IEEE) received the B.E., M.S., and Ph.D. degrees from Northeastern University, Shenyang, China, in 2009, 2011, and 2017, respectively, all in mechanical design and theory.

From 2017 to 2019, he was with the School of Mechanical and Engineering, Hunan Institute of Science and Technology, Yueyang, China, and since 2019, he has been with the School of Information Science and Technology. His research interests include the image processing, object detection, and target tracing.

Yaru Liu (Student Member, IEEE) received the B.S. degree in communication engineering from the Lanzhou University of Technology, Lanzhou, China, in 2016. She is currently working toward the M.S. degree in information and communication engineering with the Hunan Institute of Science and Technology, Yueyang, China.

Her research interests include hyperspectral image processing, computer vision, and deep learning.

Bing Tu (Member, IEEE) received the M.S. degree in control science and engineering from the Guilin University of Technology, Guilin, China, in 2009, and the Ph.D. degree in mechatronic engineering from the Beijing University of Technology, Beijing, China, in 2013.

From 2015 to 2016, he was a Visiting Researcher with the Department of Computer Science and Engineering, University of Nevada, Reno, NV, USA, which is supported by the China Scholarship Council. Since 2018, he had been an Associate Professor with the School of Information Science and Engineering, Hunan Institute of Science and Technology, Yueyang, China, where he is currently a Full Professor. His research interests include sparse representation, pattern recognition, and analysis in remote sensing.

Dr. Tu is an Associate Editor of the *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*.

Yuwen Zhang (Student Member, IEEE) received the B.S. degree in electrical engineering and automation from the Hunan Institute of Science and Technology, Yueyang, China, in 2020, where he is currently working toward the M.S. degree in information and communication engineering.

His research interests include image processing, classification of multisource remote sensing data, and object detection.