Deep Induction Network for Small Samples Classification of Hyperspectral Images

Kuiliang Gao, Wenyue Guo, Xuchu Yu, Bing Liu, Anzhu Yu, and Xiangpo Wei

***Abstract*—Recently, the deep learning models have achieved great success in hyperspectral images (HSI) classification. However, most of the deep learning models fail to obtain satisfactory results under the condition of small samples due to the contradiction between the large parameter space of the deep learning models and the insufficient labeled samples in HSI. To address the problem, a deep model based on the induction network is designed in this article to improve the classification performance of HSI under the condition of small samples. Specifically, the typical meta-training strategy is adopted, enabling the model to acquire stronger generalization ability, so as to accurately distinguish the new classes with only a few labeled samples (e.g., five samples per class). Moreover, in order to deal with the disturbance caused by the various characteristics of the samples in the same class in HSI, the class-wise induction module is introduced utilizing the dynamic routing algorithm, which can induce the sample-wise representations to the class-wise level representations. The obtained class-wise level representations possess better separability, allowing the designed model to generate more accurate and robust classification results. Extensive experiments are carried out on three public HSI to verify the effectiveness of the proposed method. The results demonstrate that our method outperforms existing deep learning methods under the condition of small samples.**

***Index Terms*—Deep learning, hyperspectral images (HSI) classification, induction network, meta-learning, small samples classification (SSC).**

I. Introduction

HYPERSPECTRAL images (HSI) contain rich spatial and spectral information, which makes it possible to accurately identify and classify ground objects. With the continuous development of imaging spectrometer, the spectral resolution and spatial resolution of HSI is increasing, which makes the description of gound object characteristics more elaborate. HSI classification, one of the most important steps in HSI analysis and application, has become an active research subject in the field of remote sensing [1].

Initially, some methods, such as principal components analysis [2] and linear discriminant analysis [3], are widely utilized to extract discriminant features from the abundant spectral information in HSI, so as to alleviate the Hughes phenomenon in the classification. Subsequently, many spatial information utilization methods, such as local binary patterns [4], extended morphological profile [5], and superpixel segmentation [6], are introduced to further improve the classification accuracy. In the traditional classification mode, the above feature extraction methods are usually combined with support vector machine (SVM) [7], random forest [8], extreme learning machine [9] and other classifiers to complete the classification. Admittedly, the traditional methods are somewhat effective in HSI classification, but its shallow learning model cannot extract the deep abstract features in HSI [10]. In addition, their classification accuracy largely depends on the handcrafted features and parameter settings, while lacking robustness in practical application.

Deep learning models, which can implement end-to-end learning by building a hierarchical framework, are very powerful for features extraction. In recent years, many deep models have been applied to HSI processing and analysis. Stacked AutoEncoder [11], ladder network [12], recurrent neural network [13], and deep belief networks [14] are first introduced to HSI classification. Given enough labeled samples, the above models can yield encouraging results. However, these models cannot fully take advantage of the spatial information in HSI because they cannot directly process the image data with 2-D structure. 2-D convolutional neural network (2D-CNN), which can directly perform 2-D convolution operation on images, has been widely used in HSI classification [15]. With the deepening of the network and the increase in the number of convolution kernels, 2D-CNN can extract rich features conducive for classification, which allows 2D-CNN to obtain better classification results. In [16], Xu *et al.* designed a 2D-CNN-based Random Patches Network utilizing the strategy of multiscale patch extraction. In [17], Zhu *et al.* proposed a novel deformable HSI classification method by introducing the deformable convolutional sampling locations. In [18], Zhao *et al.* trained a deep 2D-CNN model utilizing the strategy of multisource deep transfer learning. HSI, consisting of hundreds of bands, is a typical 3-D data cube. Therefore, compared with 2D-CNN, 3D-CNN can make better use of spatial-spectral information in HSI, to further improve the classification accuracy [19]. In [20], Fang *et al.* proposed a novel end-to-end 3-D dense convolutional network with spectral-wise attention mechanism to classify HSI. In [21], Wang *et al.* designed an alternately updated spatial-spectral convolutional network with a recurrent feedback structure to extract the refined spectral and spatial features in HSI. In [22], Li *et al.* proposed a multiscale deep middle-level feature fusion network, which can fully fuse the strong complementary and related information

Manuscript received April 1, 2020; revised May 7, 2020 and June 3, 2020; accepted June 10, 2020. Date of publication June 16, 2020; date of current version July 2, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 41801388. *(Corresponding author: Wenyue Guo.)*

The authors are with the PLA Strategic Support Force Information Engineering University, Zhengzhou 450001, China (e-mail: gokling1219@163.com; guowyer@163.com; xuchu_yu@sina.com; liubing220524@126.com; anzhu_yu@126.com; wxp0326@126.com).

Digital Object Identifier 10.1109/JSTARS.2020.3002787

among different scale features for HSI classification. In addition, CNN is also combined with capsule network (CN) [23], siamese network [24], and other novel network structures, obtaining satisfactory results with sufficient labeled samples.

Deep learning, a data-driven machine learning method, depends heavily on the quantity and quality of the labeled samples [25]. However, obtaining high-quality HSI labeled samples is both time- and labor-consuming in practice. For this reason, many researchers resort to data augmentation and semisupervised learning to improve the classification accuracy with limit labeled samples. For example, Haut *et al.* [26] utilize the random occlusion data augmentation for training model, and Li *et al.* [27] increases the number of training samples by building the pixel-pairs model. Semisupervised learning, which can combine the information of labeled and unlabeled samples, has become a hot research topic. Selftraining, an effective semisupervised learning method, has been introduced into the training phase of deep learning model [28], [29]. Generative adversarial network, containing a generative module and a discriminative module, is also widely combined with semisupervised learning to improve the HSI classification effect with limit labeled samples [30]–[32]. Another advanced semisupervised model, graph convolutional network (GCN), is also used for HSI classification and refreshes the classification accuracy [33]–[35]. Admittedly, the above methods can improve the classification accuracy to some extent, but a certain number of labeled samples must be required as guarantees for training vast network parameters. In other words, most of existing deep models cannot obtain satisfactory results when classifying HSI with only a few labeled samples (e.g., five labeled samples per class), which is called the small samples classification (SSC) of HSI [36].

When dealing with the problem of SSC, the deep learning models are very easy to overfit due to the large number of trainable parameters [37]. Humans, by contrast, particularly excel in learning with small samples because they can quickly and accurately identify objects with only a few examples. The core reason is that humans can make full use of the existing learning experience and have the ability to learn how to learn. Meta-learning, which can imitate this learning mechanism, has become a new research direction to solve the problem of SSC [38], [39]. In the field of remote sensing, Liu *et al.* first introduced the training mechanism of meta-learning into HSI classification and obtained encouraging results with a few labeled samples available [40]. Subsequently, Tang *et al.* utilized prototype network, a meta-learning implementation method, to further explore the small samples classification of HSI [41]. Gao *et al.* and Ma et al., respectively, designed different meta-learning frameworks based on relation networks (RN) for HSI classification [42], [43]. These studies have yielded promising results and opened new visions for SSC of HSI.

The above meta-learning-based methods all adopt task-based learning strategies, in which a query set are compared to a very small support set at the sample-wise level, to simulate the SSC [44], [45]. However, it is a common phenomenon that the same class has different characteristics, and different classes have similar characteristics in HSI. Therefore, the sample-wise level learning may be severely disturbed by this phenomenon, which will lead to unstable classification results. In order to mitigate the problems as much as possible, a generalized class-wise abstract representation rather than sample-wise representation needs to be learned [46]. In other words, there is a need for an architecture which can model the abstract representations of samples and dynamically induce sample-wise level representations to class-wise level representations.

The purpose of this article is to further explore the problem of HSI SSC utilizing the idea of meta-learning, so as to obtain more robust and better results. Different from previous meta-learning-based classification methods, this article attempts to learn a generalized class-wise level representation in deep metric space. Specifically, a deep model based on induction network is designed for HSI SSC (DIN-SSC). The designed model includes three modules: the feature embedding module, the class-wise induction module, and the relation-learning module. The feature embedding module, equivalent to a nonlinear transformation function, is responsible for generating abstract representations of the support samples and query samples. The class-wise induction module, the core module of the designed model, can dynamically induce sample-wise level representations to class-wise level representations leveraging the matrix transformation and the dynamic routing algorithm which are the information transmission mechanism in CN [47]. In practice, sample-wise representations and class-wise representations are treated as the input and output capsules, respectively. The relation learning module generates a relation score by comparing the similarity between the class-wise representations from the support sets and the sample-wise representations from the query sets, so as to complete the SSC of HSI. The proposed method is actually a metric-based meta-learning method. The core idea is to transform the input data into a deep metric space where samples belonging to the same class are clustered together, and samples belonging to different classes are separated. The standard meta-training process, i.e., the task-based learning strategy, is used for training the designed model. The designed model first performs meta-training on the precollected HSI, and then classifies the target HSI with small labeled samples (five labeled samples per class). It should be noted that there is no intersection between the HSI for meta-training and the HSI for SSC.

The main contributions of this article can be summarized as follows

1) We design an end-to-end deep model based induction network for SSC of HSI and train it with the standard meta-training strategy. Specifically, we design the feature embedding module utilizing 3-D convolution and deep residual structure, to make full use of the spatial-spectral information in HSI. Furthermore, we design the relation learning module by combining the convolutional layers with the fully connected layers, to carry on relation learning in the deep metric space.
2) To deal with the disturbance caused by the various characteristics in the same class in HSI, the class-wise induction module is introduced, which can enable the model to induce the class-wise level representations from the sample-wise representations, so as to enhance the separability between samples from different classes and

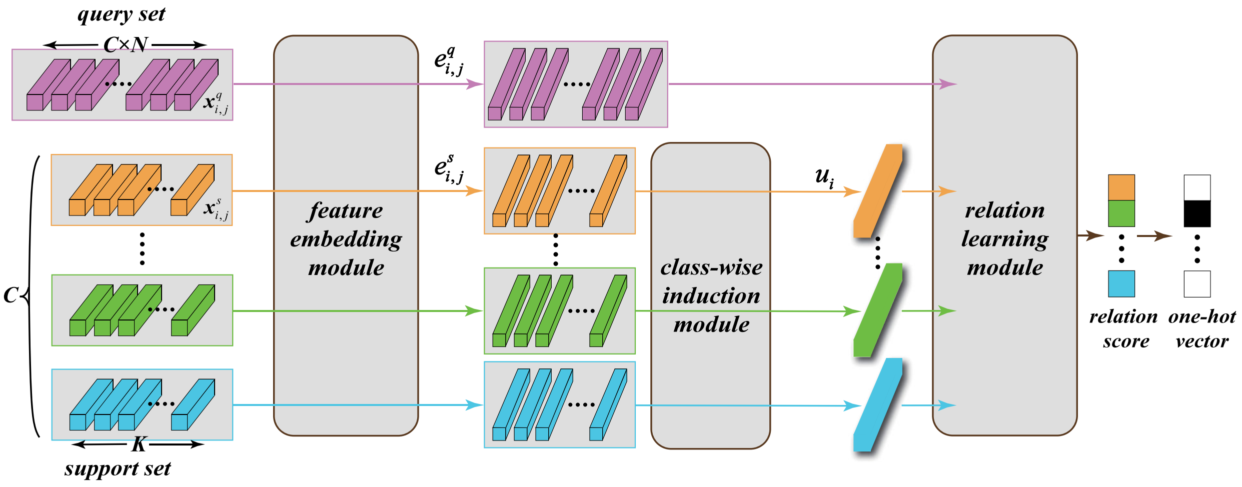


Fig. 1. Designed deep induction network model for SSC of HSI. $x^s_{i,j}$ and $x^q_{i,j}$ denotes the support samples and the query samples, respectively, $e^s_{i,j}$ and $e^q_{i,j}$ denotes the sample-wise representations, u_i denotes the class-wise level representations.

the aggregation between samples from the same classes. To the best of our knowledge, it is the first attempt to apply the class-wise induction module in HSI classification. In addition, we make a visual analysis on the effectiveness of this module.

The remainder of this article is structured as follows. In Section II, the designed deep model (DIN-SSC) and meta-training strategy are described in detail. In Section III, experimental results and detailed analysis are presented. Finally, Section IV concludes this article.

II. Proposed Method

The designed model, based on the typical meta-learning framework where tasks containing support sets and query sets are treated as the basic unit for training, mainly contains three modules: the feature embedding module, the class-wise induction module, and the relation learning module (see Fig. 1). Given a task consisting of a small support set and a query set, the feature embedding module first models the abstract representations of the samples. Then, the class-wise induction module transforms the sample-wise representations from the support sets to the class-level representations. Finally, the relation learning module calculates the relation score by comparing the similarity between the sample-wise representations from the query sets and the class-level representations from the support sets to determine the classes of the query samples. In the following sections, the designed model and the meta-training strategy are described in detail.

A. Feature Embedding Module

For clarity, the support set and the query set are denoted as $\mathcal{S}$ and $\mathcal{Q}$, respectively. Following the general principles for building tasks in meta-learning [44], [45], the support set consisting of C classes with K samples per class can be denoted as $\mathcal{S} = \{(x^s_{i,j}, y^s_{i,j})\}_{i=1,\ldots,C,j=1,\ldots,K}$, and the query set consisting of C classes with N samples per class can be denoted as $\mathcal{Q} = \{(x^q_{i,j}, y^q_{i,j})\}_{i=1,\ldots,C,j=1,\ldots,N}$. The feature embedding module, which can be regarded as a nonlinear function $f()$, is responsible for mapping the samples $x_{i,j}$ to the abstract representations $f(x_{i,j})$. This nonlinear mapping is at the sample-wise level due to it performs the same operations for each sample from the support sets and query sets.

Considering the fact that HSI contain both abundant spatial and spectral information, we design the feature embedding module based on the 3-D convolution and the deep residual structure (see Fig. 2). The feature embedding module takes the data cubes around the pixels in HSI as input and the feature vectors, i.e., the sample-wise representations, as output. 3-D convolution, extracting more informative and robust features in spatial and spectral domains at the same time, has been widely used in HSI processing and analysis. The feature embedding module is a deep residual network including many basic operational units which contains 3-D convolutional layers, batch normalization layers and Relu activation function. The residual structure is conducive for training the deep network which can allow the convolutional layers gradually extract more abstract features from the input data cube. Batch normalization layers and Relu function both can alleviate the problem of vanishing gradient and speed up the convergence. In convolutional layers, the kernel size is uniformly set to $3 \times 3 \times 3$, which has been proved to be effective for local features extraction [48]. Between the residual blocks, the max-pooling layers are added to reduce the dimension of the feature maps.

In summary, a deep residual network consisting of three residual blocks and ten convolutional layers is constructed, to take full advantage of the spatial-spectral information in HSI and generate the sample-wise level representations.

B. Class-Wise Induction Module

The phenomenon that the samples from the same class possess different characteristics and the samples from different classes possess similar characteristics, is very common in HSI. In this

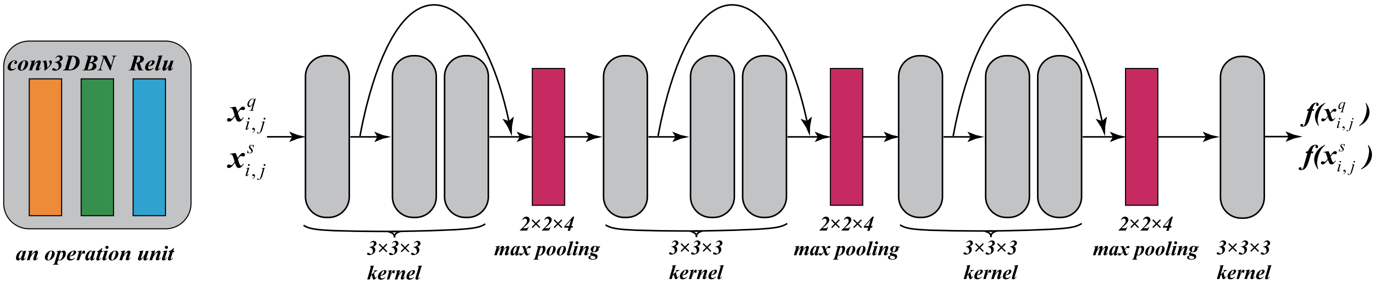


Fig. 2. Feature embedding module based on the 3-D convolution and the deep residual structure. BN denotes a batch normalization layer, conv3D denotes a 3-D convolution layer.

case, it is susceptible that only utilizing the sample-wise level representations for comparison in the metric space. Therefore, the designed model should be able to ignore the information irrelevant to classification, encapsulate the information closely related to classification into a class vector, which is just the goal of the class-wise induction module. For the samples from the support sets, the class-wise induction module induce the sample-wise representations belonging to the same class $e^s_{i,j}$ to the class-wise level representations u_i

$$\{e^s_{i,j}\}_{i=1,\ldots,C,j=1,\ldots,K} \mapsto \{u_i\}_{i=1,\ldots,C}. \tag{1}$$

The dynamic routing algorithm, which is the core information transmission mechanism in CN, is introduced to implement the class-wise induction module. In CN, the dynamic routing algorithm is used to model the abstract spatial relationship between the parts in low-level capsules and the whole in high-level capsules. In the problem of HSI SSC, we can regard the samples as the parts and the classes as the whole. In other words, the sample-wise level representations are treated as the input capsules and the class-wise level representations are treated as the output capsules in the dynamic routing algorithm.

Specifically, the weight-sharable transformation matrix W_s and bias b_s are employed across all the support samples, to accommodate all possible inputs, i.e., a C-way K-shot task at any scale. In this case, the prediction vector $\hat{e}^s_{i,j}$ can be computed by

$$\hat{e}^s_{i,j} = \text{squash}(W_s e^s_{i,j} + b_s) \tag{2}$$

$$\text{squash}(x) = \frac{\|x\|^2}{1+\|x\|^2}\frac{x}{\|x\|}. \tag{3}$$

In (3), squash is a nonlinear squashing function designed for vector calculation [47], which can keep the direction of the vector unchanged but scales its magnitude to $[0, 1]$. Through the affine transformation, (2) implements the map from sample-wise space to class-wise space. Next, we need to apply the dynamic routing algorithm to induce the abstract representations belonging to the same class to a class-wise level representation

$$d_i = \text{softmax}(b_i) \tag{4}$$

$$\hat{u}_i = \sum_j d_{ij} \cdot \hat{e}^s_{i,j} \tag{5}$$

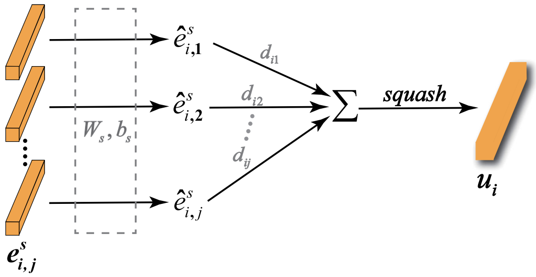


Fig. 3. Visual representation of the class-wise induction module.

$$u_i = \text{squash}(\hat{u}_i) \tag{6}$$

$$b_{ij} = b_{ij} + \hat{e}^s_{i,j} \cdot u_i. \tag{7}$$

The dynamic routing algorithm can be mainly divided into four steps: "routing softmax," calculating weighted sum, applying squash function and updating coupling coefficient. In (4), b_i denoting the logits of coupling coefficients is initialized by 0 in the first iteration. The coupling coefficients d_i, which can be automatically modified in each iteration, sum to 1 between class i and all support samples in this class. Each class candidate vector $\hat{u}_i$ is obtained by calculating the weighted sum of all sample prediction vectors $\hat{e}^s_{i,j}$ in this class (5). Then, the squash function is applied to the vector $\hat{u}_i$, to ensure its magnitude will not exceed 1 (6). Finally, the logits of coupling coefficients b_{ij} is updated according to the scalar output between the out class vector u_i and the sample prediction vectors $\hat{e}^s_{i,j}$. This update is very effective for the SSC due to it does not need to save any parameters. In order to clearly illustrate the whole workflow, the class-wise induction module is summarized as Algorithm 1 in the form of pseudocode. In addition, Fig. 3 shows the visual representation of the class-wise induction module.

Utilizing transformation matrix and dynamic routing algorithm in the class-wise induction module, multiple sample-wise representations from the same class are finally encapsulated into one representation. In the process, the various information irrelevant to classification in the sample-wise representations is abandoned, and the discriminant information closely related to classification is reserved. Consequently, the obtained class-wise representations possess better separability in the metric space, which will be verified in Section III-F.

Algorithm 1: The Class-Wise Induction Module.

```
Require: The sample-wise representations $e^s_{i,j}$ in the support sets and the logits of coupling coefficients $b_{ij}$ is initialized as 0.
Ensure: The class-wise level representations $u_i$
for all sample-wise representations $j = 1, \ldots, K$ in class $i$: $\hat{e}^s_{i,j} = squash(W_s e^s_{i,j} + b_s)$
for $r$ in iterations do
    $d_i = softmax(b_i)$
    $\hat{u}_i = \sum_j d_{ij} \cdot \hat{e}^s_{i,j}$
    $u_i = squash(\hat{u}_i)$
    for all sample-wise representations $j = 1, \ldots, K$ in class $i$: $b_{ij} = b_{ij} + \hat{e}^s_{i,j} \cdot u_i$
end for
return $u_i$
```

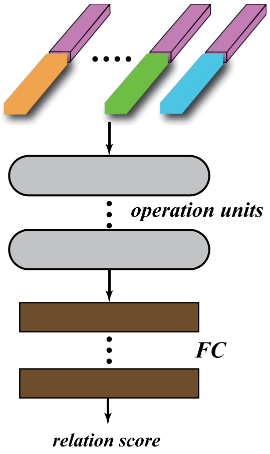


Fig. 4. Relation learning module. FC denotes a fully connected layer.

C. Relation Learning Module

After the action of the first two modules, the abstract representations of the query samples and the class-wise level representations of the support samples have been obtained. The purpose of the relation learning module is to determine the class of the query samples by comparing the similarities between different representations in the deep metric space. As shown in Fig. 4, the abstract representation of each query sample is concatenated to the class-wise representations from the support set. Then, the relation score of the resulting concatenation is calculated by a designed network. Specifically, if the two representations in a concatenation belong to the same class, the relation score should be equal to 1, otherwise 0.

Theoretically, any network structure can be used to calculate the relation scores. Referring to [45], 2-D convolutional layers and the fully connected layers are utilized to construct the network. The convolutional layers, which are packed in an operation unit as Section II-A, can further extract the feature information in different representations, while the fully connected layers can deepen the network, to enhance the nonlinearity. In addition, Dropout is added between the fully connected layers to enhance the generalization capability of the network. At the end of the network, the sigmoid function is used to map the output to [0, 1].

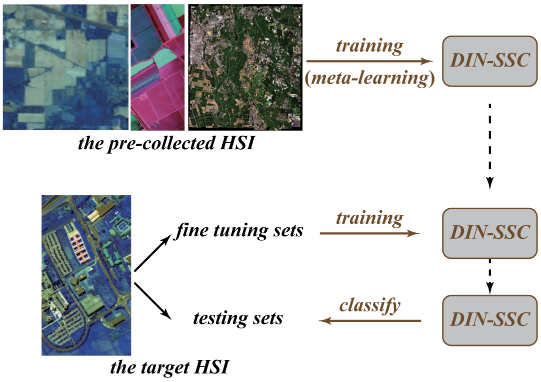


Fig. 5. Meta-training strategy for HSI SSC.

D. Meta-Training Strategy

The core of meta-learning is to enable the model to acquire rich transferable knowledge and master the ability of how to learn through vast task-based learning, which can simulate SSC in the early stage. However, there is no large available image database like ImageNet in the field of remote sensing. Therefore, we artificially divide the existing HSI into the precollected HSI and the target HSI, so that the model to fully perform meta-learning. In addition, we randomly select five samples per class from the target HSI to build the fine tuning sets, and utilize the remaining samples as the testing sets. As shown in Fig. 5, the designed model is first trained on the precollected HSI (meta-learning), then trained on the fine-tuning sets, and finally perform the classification on the testing sets. During the training phases, the mean square error (MSE) is adopted as the loss function of the designed model

$$L_{\text{MSE}} = \sum_{i=1}^{C} \sum_{k=1}^{C\times N} (r_{i,k} - 1 \cdot (y^q_k == i))^2 \tag{8}$$

where $r_{i,k}$ denotes the relation scores, y^q_k denotes the true labels of the query samples.

In the whole process, only five samples per class are randomly selected from the target HSI to fine tune the designed model. In other words, only five samples per class, which are extremely small, are utilized as the supervised samples. Therefore, what the designed model performs is the SSC of HSI.

III. Experiments and Discussion

The designed model and other algorithms are developed and implemented based on Python and Pytorch library. All experimental results are generated on a laptop with an Intel Core i7-9750H, 16 GB memory, and an NVIDIA GeForce RTX 2070.

TABLE I
DETAILS OF THE HSI FOR META-LEARNING

	IP	**KSC**	**CH**
Spatial size	145 × 145	512 × 614	2517 × 2335
Spectral range	400-2500	400-2500	363-1018
No. of bands	200	176	128
GSD	20	18	2.5
Sensor type	AVIRIS	AVIRIS	Hyperspec-VNIR-C
Areas	Indiana	Florida	Chikusei
Number of class	16	13	19
Labeled samples	10249	5211	77592

Ground sample distance (GSD) (m), spatial size (Pixel), spectral range (nm), airborne visible infrared imaging spectrometer (AVIRIS), and reflective optics system imaging spectrometer (ROSIS).

TABLE II
DETAILS OF THE TARGET HSI

	UP	**SA**	**HS**
Spatial size	610 × 340	512 × 217	349 × 1905
Spectral range	430-860	400-2500	380-1050
No. of bands	103	204	144
GSD	1.3	3.7	2.5
Sensor type	ROSIS	AVIRIS	ITRES-CASI 1500
Areas	Pavia	California	Houston
Number of class	9	16	30
Labeled samples	42776	54129	15029

Ground sample distance (GSD) (m), spatial size (Pixel), spectral range (nm), airborne visible infrared imaging spectrometer (AVIRIS), and reflective optics system imaging spectrometer (ROSIS).

A. Building the Experimental Datasets

In order to verify the effect of DIN-SSC on HSI SSC, the experiments are carried out using the six HSI, including Indian Pines (IP), Kennedy Space Center (KSC), Chikusei (CH), University of Pavia (UP), Salinas (SA) and Houston (HS). The six HSI are captured by different imaging spectrometers with different ground objects, ground sample distance and spectral range, which is helpful for building rich and various tasks for meta-learning (see Tables I and II). In the experiments, the meta-training strategy in Section II-D was employed. Specifically, the first three HSI are used for meta-learning and the last three for classification. This is a viable strategy, and in fact any other strategy leveraging knowledge in the precollected HSI can be adopted. In practice, an advanced band selection method (ISSC) [49] was utilized to select 100 bands instead of all bands for each HSI, to standardize the spectral dimensions of the different HSI, while reducing spectral redundancy. Referring to [50], the $17 \times 17 \times 100$ cubes around the pixel are selected as the input data. The above approaches can ensure that the samples from different HSI have a uniform data dimension, which facilitates subsequent calculations. For IP, KSC, and CH, 200 samples per class are randomly selected to build to the datasets for meta-learning. Consequently, the resulting dataset consists of 41 class with 8200 samples For UP, SA, and HS, five samples per class are randomly selected to build to the datasets for fine-tuning, and the remaining samples are used for testing.

TABLE III
OVERALL ACCURACY (OA) (%) UNDER DIFFERENT TASK SETTINGS

	C = 10			*C* = 20			*C* = 30		
K/N	1/19	5/15	10/10	1/19	5/15	10/10	1/19	5/15	10/10
UP	83.04	85.36	82.94	84.79	**86.22**	82.93	84.15	85.26	82.02
SA	89.19	90.77	89.84	90.93	**91.69**	89.31	91.81	90.94	90.76
HS	51.30	52.75	51.26	53.26	56.49	56.65	58.09	**59.39**	55.12

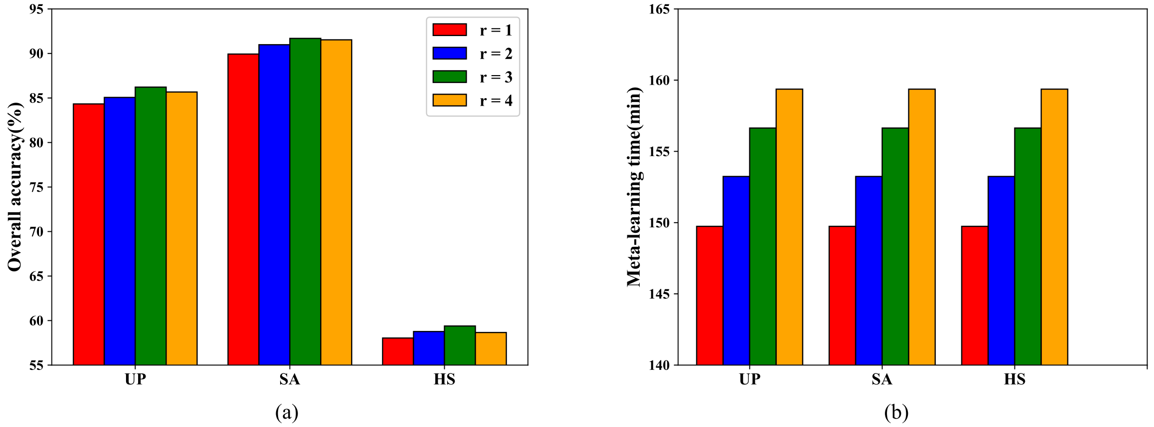
B. Experimental Setting

The model performs meta-learning on the precollected HSI, to acquire a wealth of knowledge and develop the ability to learn how to learn. In the meta-learning phase, tasks are the basic units of model training. In other words, a task is just an iteration. Specifically, a task, generated randomly from the precollected HSI, is determined by the number of classes C, the number of the support samples per class K, and the number of the query samples per class N. Therefore, the influence of C, K, and N on the classification accuracy is first investigated experimentally.

It can be seen from the results in Table III that the values of C, N, and K have a significant impact on the classification results, which indicates that tasks setting in the meta-learning phase is very important. Specifically, the following two observations can be summarized from Table III.

1) The number of the classes C can directly affect the complexity of the tasks, i.e., the larger the value of C, the more complex the tasks. The appropriate value of C can enable the model to obtain the best classification results, too large or too small will lead to the decline in classification accuracy. For UP and SA, the optimal value of C is 20, while for HS, the optimal value of C is 30. This indicates that the optimal value of C is correlated with the number of classes in the target HSI (UP:9, SA:16, HS:30).
2) K and N determine the ratio between the support samples and the query samples in a task. Some studies have shown that setting K far less than N in the tasks (e.g., $K = 1$ and $N = 19$) can improve the classification accuracy by simulating the condition of SSC in the training phase [42], [44]. However, in the designed model, the class-wise induction module needs a certain number of support samples for the induction of class-level representations, which is beneficial to accurately determine the classes of query samples. Considering the above two points, it is easy to understand why the optimal values of K and N are 5 and 15, respectively.

The dynamic routing algorithm is the key to implement the class-wise induction module. In each training iteration, the number of routing r has a great influence on the effect of the



(a)

(b)

Fig. 6. Influence of r. (a) OA under different r. (b) Meta-learning time under different r.

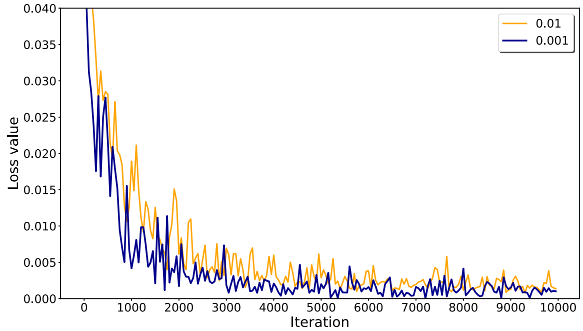


Fig. 7. Loss value of the designed model with different learning rate.

TABLE IV
OA (%) ON UP UNDER DIFFERENT NETWORK STRUCTURE

No.	1	2	3	4
FEM	$COV^2(16)$ $RB^2(16)$ $MP^2(2\times 2)$ $COV^2(32)$ $RB^2(32)$ $MP^2(2\times 2)$ $COV^2(64)$ $RB^2(64)$ $MP^2(2\times 2)$ $COV^2(128)$	$COV^3(16)$ $MP^3(2\times 2\times 4)$ $COV^3(32)$ $MP^3(2\times 2\times 4)$ $COV^3(64)$ $MP^3(2\times 2\times 4)$ $COV^3(128)$	$COV^3(16)$ $RB^3(16)$ $MP^3(2\times 2\times 4)$ $COV^3(32)$ $RB^3(32)$ $MP^3(2\times 2\times 4)$ $COV^3(64)$ $RB^3(64)$ $MP^3(2\times 2\times 4)$ $COV^3(128)$	$COV^3(16)$ $RB^3(16)$ $MP^3(2\times 2\times 4)$ $COV^3(32)$ $RB^3(32)$ $MP^3(2\times 2\times 4)$ $COV^3(64)$ $RB^3(64)$ $MP^3(2\times 2\times 4)$ $COV^3(128)$
RLM	$COV^2(512)$ $FC(128)$ $FC(1)$	$COV^2(512)$ $FC(128)$ $FC(1)$	$COV^2(512)$ $FC(128)$ $FC(1)$	$COV^2(512)$ $COV^2(1024)$ $FC(256)$ $FC(128)$ $FC(1)$
OA	83.94	84.90	86.22	86.04

The Feature Embedding Module (FEM), The Relation Learning Module (RLM), Residual Block (RB), Max Pooling (MP), Fully Connected Layer (FC). CONV Denotes a Convolutional Layer. The Superscript 2 or 3 Denotes Two or 3-D Operations.

dynamic routing algorithm. Fig. 6 shows the influence of r on the classification results and the meta-learning time. As we can see, with the increase of r, the classification accuracy increases gradually at first and then tends to be stable, while the meta-learning time increases linearly. Considering the classification accuracy and execution efficiency comprehensively, r is uniformly set to 3.

The learning rate determines the convergence speed of the model, which can indirectly affects the meta-learning. Referring to relevant experiments, we explored the influence of learning rate at 0.01 and 0.001 on the loss value. Fig. 7 shows the experimental results. It can be found that a smaller and more stable loss value can be obtained when the learning rate is 0.001, which means that the model can learn more fully.

Finally, we explored the influence of different network structures on the classification results of UP. Specifically, the residual structure and the dimension of the convolution operation in the feature embedding module, and the number of network layers in the relation learning module are investigated in detail. Table IV lists the specific structure settings and the corresponding classification accuracy. For example, $\text{COV}^3(16)$ denotes a 3-D convolutional layer with 16 convolution kernels, $\text{RB}(16)$ indicates that the number of all convolution kernels in the residual block is 16, and FC (128) denotes a fully connected layer containing 128 neurons. Apart from the information listed in this table, the network details are exactly identical to Sections II-A and II-C. According to the results in Table IV, the following three observations can be obtained.

1) Comparing No .1 and No. 3 network settings, we can find that 3-D convolution operation can produce higher classification accuracy compared 2-D convolution operation. HSI possesses rich spectral and spatial information at the same time, and 3-D convolution can make full use of the spatial-spectral information, to further improve the classification accuracy.
2) By comparing No.2 and No.3 network settings, it is easy to see that the model with residual structure can obtain the higher classification accuracy. By introducing residual structure, the model can extract more abstract and robust features, thus further improving the classification performance.
3) By comparing No.3 and No.4 network settings, we find that using deep network in the relation learning module

TABLE V
CLASSIFICATION RESULTS OF DIFFERENT METHODS ON UP (FIVE SAMPLES PER CLASS ARE USED AS THE SUPERVISED SAMPLES; BOLD VALUES REPRESENT THE BEST RESULTS AMONG THESE METHODS)

Class No.	SVM	3D-CNN	iCapsNet	TL+CRNN	HSGAN	CNN+GCN	RN-FSC	DFSL+SVM	DIN-SSC
1	80.59	69.79	65.84	71.05	96.94	97.73	87.28	78.05	96.36
2	81.43	82.38	84.55	86.18	92.19	92.38	84.33	88.54	94.00
3	28.58	21.86	29.91	35.49	52.30	53.84	90.42	57.40	83.46
4	54.41	98.81	82.70	75.81	99.22	89.81	78.09	62.73	95.64
5	97.69	74.94	100.00	91.50	100.00	99.34	99.56	98.82	99.93
6	73.53	55.36	40.22	44.34	38.34	38.36	63.25	43.26	53.13
7	34.99	21.70	21.40	25.56	57.20	61.04	52.09	100.00	84.96
8	14.33	34.11	57.19	66.98	90.14	85.30	84.81	71.66	90.53
9	100.00	96.81	0.00	70.14	48.10	99.27	95.94	97.83	98.32
OA	65.73	64.35	66.63	70.29	73.71	75.39	81.94	73.51	**86.22**
AA	62.84	61.75	53.54	63.00	74.94	79.67	81.75	77.57	**88.48**
Kappa	55.25	52.55	56.07	61.18	67.16	69.25	75.84	66.21	**82.25**

will not improve the classification accuracy further. The network consisting of a convolutional layer and two fully connected layers is sufficient to implement the relation learning between the different representations.

In summary, through the above experiments, the hyperparameters including the optimal task settings, the number of routing, and learning rate in the meta-learning phase, as well as the network settings of the model are explored in detail, so that the design model can obtain the best classification effect. Referring to [40] and [42], the number of iterations in the meta-learning phase and the fine-tuning phase are set to 10 000 and 1000, respectively. The process of meta-learning and fine-tuning is task-based, and a task is equivalent to an iteration. It means that 10 000 and 1000 tasks need to be generated for meta-learning and fine-tuning, respectively. In addition, Adam is adopted as the optimization algorithm, so that the design model can perform the sufficient meta-learning.

C. Comparison and Analysis

In order to verify the effectiveness of the proposed model in the SSC of HSI, we compare the experimental results with SVM, two classical deep learning methods 3-D-CNN [51] and iCapsNet [23], two advanced semisupervised deep learning methods HSGAN [31] and CNN+GCN [33], and two meta-learning-based methods RN-FSC [42] and DFSL+SVM [40]. SVM, more suitable for processing the high-dimensional data than other traditional classifiers, has been widely used in HSI classification. 3-D-CNN, a classical deep learning model including 3-D convolutional layers and fully connected layers, can make full use of the spatial-spectral information in HSI. iCapsNet, a variant of the CN used for HSI classification, can encode the abstract features at a higher level utilizing the capsule neurons. Both HSGAN and CNN+GCN can further improve the classification accuracy by using the information of unlabeled samples. RN-FSC and DFSL+SVM, also designed based on the meta-learning framework, have been demonstrated to achieve better results in the problem of HSI SSC. In addition, a deep transfer learning method TL+CRNN [52], which utilizes the UBC dataset for transfer learning, is also used for comparison. It should be emphasized again that for all methods, there are only five labeled samples per class of the target HSI as the supervised samples. In addition, the OA, average accuracy (AA) and Kappa coefficient are selected as the evaluation indicators to quantitatively compare the classification results of different methods.

Tables V–VII list the classification results of different methods, which can be summarized as the following five points.

1) The classification results of SVM are generally better than the two classic deep learning models. With only five labeled samples per class used for training, the deep learning models are prone to overfit, which seriously affects the classification accuracy. For example, the OA of 3-D-CNN on SA is only 74.24%. More seriously, the classification accuracy of iCapsNet on No. 9 class of UP is 0.00%.
2) TL+CRNN, a typical transfer learning method, effectively improves the classification accuracy by utilizing the transferable knowledge from the UBC dataset. In the three HSI, the classification performance of TL+CRNN is better than that of other two deep learning models. However, there is still a significant gap between the classification results of TL+CRNN and that of the three meta-learning-based methods. Transfer learning aims at transferring knowledge from a specific task space, while the goal of meta-learning is to develop the ability of learning how to learn through vast different tasks.
3) Semisupervised learning can effectively utilize the information in unlabeled samples and alleviate the problem of overfitting caused by insufficient labeled samples to some extent, so as to improve the classification accuracy. Therefore, it can be found that HSGAN and CNN+GCN, two semisupervised deep learning models, possess better classification results than 3-D-CNN and iCapsNet on the three HSI.
4) Two meta-learning-based methods, RN-FSC and DFSL+SVM, can further improve the classification

TABLE VI
CLASSIFICATION RESULTS OF DIFFERENT METHODS ON SA (FIVE SAMPLES PER CLASS ARE USED AS THE SUPERVISED SAMPLES; BOLD VALUES REPRESENT THE BEST RESULTS AMONG THESE METHODS)

Class No.	SVM	3D-CNN	iCapsNet	TL+CRNN	HSGAN	CNN+GCN	RN-FSC	DFSL+SVM	DIN-SSC
1	80.42	89.02	93.41	39.73	59.35	99.95	99.26	87.33	81.20
2	98.93	97.38	67.16	99.40	98.88	95.28	100.00	99.68	83.06
3	76.93	72.43	86.49	87.68	77.24	84.24	97.87	79.97	98.97
4	96.63	80.79	90.36	99.19	93.30	96.99	99.50	97.00	99.85
5	98.96	95.54	90.39	99.59	86.95	98.09	97.81	96.77	98.55
6	100.00	96.87	99.02	99.29	99.90	99.95	99.35	99.92	99.85
7	95.63	92.33	95.07	90.23	84.94	92.50	100.00	94.87	97.13
8	72.64	62.27	82.13	89.55	68.06	62.58	66.24	70.52	96.98
9	90.49	93.84	98.63	96.91	91.43	99.10	97.34	97.95	99.47
10	74.32	77.13	93.12	78.23	93.86	85.19	93.66	83.97	95.33
11	56.18	81.57	60.09	77.76	49.04	53.25	73.96	69.36	93.44
12	83.65	99.57	94.78	92.77	88.50	97.95	99.84	96.21	100.00
13	87.70	81.03	49.51	92.33	45.48	80.04	100.00	96.41	93.56
14	90.21	73.25	76.76	94.23	90.34	97.22	96.39	95.69	98.26
15	53.88	42.02	49.76	62.24	54.62	72.64	68.85	66.95	72.27
16	74.31	100.00	96.89	83.76	60.71	88.56	99.89	95.26	97.71
OA	79.51	74.24	77.26	80.96	75.66	82.69	86.99	85.36	**91.69**
AA	83.18	83.44	82.72	86.43	77.66	87.72	93.12	89.24	**94.10**
Kappa	77.32	71.66	74.92	78.98	72.90	80.62	85.44	83.68	**90.78**

accuracy of HSI when only a few labeled samples are available. Before classifying the target HSI, both RN-FSC and DFSL+SVM have performed sufficient meta-learning on the precollected HSI, which can significantly improve the generalization ability of the models. Therefore, compared with the deep learning models which only utilize the information in the target HSI, RN-FSC, and DFSL+SVM can achieve higher classification accuracy.

5) The proposed method, DIN-SSC, obtains the best classification results on the three HSI, with the highest OA, AA, and Kappa. Compared with the other two meta-learning-based methods (RN-FSC and DFSL+SVM), the significant advantage of DIN-SSC is the class-wise induction module, which can dynamically induce the sample-wise representations to the class-wise level representations and enhance the separability between different classes, thus further improving the classification accuracy. On the three HSI, DIN-SSC improves the OA by $4\% - 7\%$ and Kappa by $4\% - 8\%$ than the other meta-learning-based methods, which is a significant improvement.

6) The Houston data set including 30 classes, is very challenging for accurate classification. As can be seen from Table VII, the OA of all classification methods is below 52% except the proposed method. The OA and Kappa of DIN-SSC are 8.20% and 8.15% higher than DSFL+SVM, which can show the advantages of DIN-SSC in processing the complex HSI.

The classification maps, which can more directly show the spatial information of ground objects, is very important for HSI processing and analysis. Therefore, we utilize the above methods to classify all the samples in the three HSI, so as to generate the global classification maps (see Figs. 8–10). It can be seen that, with the continuous improvement of classification accuracy, the misclassification phenomenon in the global classification maps gradually decreases. From the perspective of spatial consistency, DIN-SSC can obtain the best global classification map.

In order to enhance the persuasion of the experimental results, we repeated the experiment ten times utilizing different methods and plot the boxplots of Kappa. In Fig. 11, different colors represent different methods, and the circles (○) represent outliers in the experimental results. As we can see, the Kappa of DIN-SSC on the three HSI are more clustered than other methods, which means it has better robustness. In addition, the information contained in the boxplots can indicate that the observed increase in classification accuracy is statistically significant.

D. *Influence of the Spatial Size of the Input Cubes*

As described in Section III-A, the $17 \times 17 \times 100$ cubes around the pixel are selected as the input data. In this section, the influence of the spatial size of the input cubes on the classification results is explored in detail. The spatial size of the input cubes affects how much spatial information it contains. Theoretically, the larger the spatial size, the more neighborhood pixels the cube contains and the more spatial information the model can utilize. Table VIII lists the results of DIN-SSC using cubes with different spatial size. In general, the classification accuracy has a significant improvement as the spatial size becomes larger, and then decreases slightly after reaching the peak. It indicates that the appropriate spatial size can enable the model to achieve the best classification performance, too large or too small will lead to the decline in classification accuracy.

TABLE VII
CLASSIFICATION RESULTS OF DIFFERENT METHODS ON HS (FIVE SAMPLES PER CLASS ARE USED AS THE SUPERVISED SAMPLES; BOLD VALUES REPRESENT THE BEST RESULTS AMONG THESE METHODS)

Class No.	SVM	3D-CNN	iCapsNet	TL+CRNN	HSGAN	CNN+GCN	RN-FSC	DFSL+SVM	DIN-SSC
1	25.97	25.40	71.43	14.58	12.56	18.32	15.35	19.00	31.83
2	19.68	15.55	16.67	27.52	28.75	25.12	21.11	22.39	23.62
3	43.56	22.94	26.99	51.39	63.37	8.58	34.70	39.43	54.02
4	20.42	16.90	19.95	20.53	19.49	20.30	14.95	21.77	22.04
5	17.30	17.19	12.88	15.56	22.30	10.39	16.18	18.49	22.24
6	64.91	40.12	86.67	49.58	74.36	55.33	70.51	78.67	77.84
7	15.35	12.23	9.02	19.43	23.11	15.58	12.72	29.58	22.52
8	10.68	15.36	27.42	34.56	18.80	15.23	7.69	21.60	29.89
9	9.76	13.25	13.24	33.54	22.85	5.77	10.65	17.68	19.67
10	7.49	26.03	13.81	16.31	26.29	16.45	8.03	24.55	25.16
11	10.65	27.56	21.48	46.90	35.03	12.28	11.80	32.85	20.89
12	10.21	11.50	15.55	42.86	0.00	6.80	15.13	20.61	26.63
13	7.21	36.75	27.50	61.39	29.38	41.59	38.21	71.15	29.21
14	30.46	43.16	37.50	33.48	49.38	20.67	43.87	79.90	47.18
15	34.53	80.73	50.32	80.20	63.82	34.15	27.72	91.96	59.16
16	86.94	77.91	79.70	42.50	64.00	36.55	89.43	51.43	90.38
17	83.67	53.84	40.25	49.17	33.33	83.90	87.62	96.24	89.81
18	84.48	55.87	57.72	87.66	57.71	48.85	77.45	95.61	88.63
19	83.76	83.78	85.54	57.69	74.14	74.39	82.33	58.51	87.00
20	84.46	84.87	71.80	73.10	98.67	76.88	80.77	92.52	91.98
21	45.45	24.80	30.28	18.18	57.06	26.32	44.17	49.36	70.00
22	63.37	80.97	61.11	59.07	58.71	73.99	61.24	92.77	90.40
23	33.10	36.42	58.32	35.24	70.57	87.87	85.33	95.62	85.21
24	54.82	69.56	54.50	54.78	84.12	75.45	83.03	88.36	77.51
25	75.74	53.21	63.17	83.56	61.52	82.47	72.17	74.19	84.41
26	35.93	70.43	51.57	72.01	65.14	40.33	62.40	78.82	78.53
27	53.64	72.76	67.81	66.74	52.29	37.04	75.37	76.76	81.71
28	10.37	53.07	16.59	58.27	62.60	42.05	32.00	89.35	38.26
29	44.29	52.20	41.10	32.66	76.88	35.04	60.56	88.11	61.62
30	72.54	90.33	74.18	75.89	62.11	71.31	31.11	94.80	91.40
OA	40.81	41.15	39.20	42.10	45.29	40.20	47.06	51.19	**59.39**
AA	41.36	45.49	43.47	47.15	48.95	39.97	45.79	60.10	**57.29**
$Kappa$	38.28	38.95	36.92	39.90	43.21	38.00	44.92	49.54	**57.69**

TABLE VIII
CLASSIFICATION RESULTS(OA, %) OF DIN-SSC USING CUBES WITH DIFFERENT SPATIAL SIZE

taget HSI	5 × 5	9 × 9	13 × 13	17 × 17	21 × 21
UP	82.86	83.41	85.67	86.22	86.13
SA	87.37	89.06	91.73	91.69	91.26
HS	55.72	56.08	58.63	59.39	59.07

E. Influence of the Number of Labeled Samples

In order to further verify the effectiveness of the proposed method in HSI SSC, the classification results of different methods with different number of labeled samples are investigated. Specifically, 10, 15, 20, and 25 labeled samples per class are randomly selected as the supervised samples in the experiments. It should be noted that all setups in the experiments are identical to the previous sections, except for the number of labeled samples. Fig. 12 shows the experimental results. As we can see, the OA of all methods is increasing with the increase of labeled samples, and the OA of DIN-SSC is always higher than that of other methods, which indicates that DIN-SSC possesses the best adaptability to the variation of the number of labeled samples.

The above experiments have demonstrated that the proposed method can effectively improve the classification accuracy under the condition of small samples. However, an ideal model should be adaptable enough to the classification condition, so as to meet the requirements in practical application. To this end, we explore the classification effect of DIN-SSC when the number of labeled samples is further increased. Take UP as an example, 50, 100, and 200 labeled samples per class are randomly selected as the fine-tuning datasets to fully train the designed model. Table IX lists the experimental results of the proposed methods and several advanced classification methods including MDSFV [53], IEPF-G-c [54], SVM+SCNN [24], and

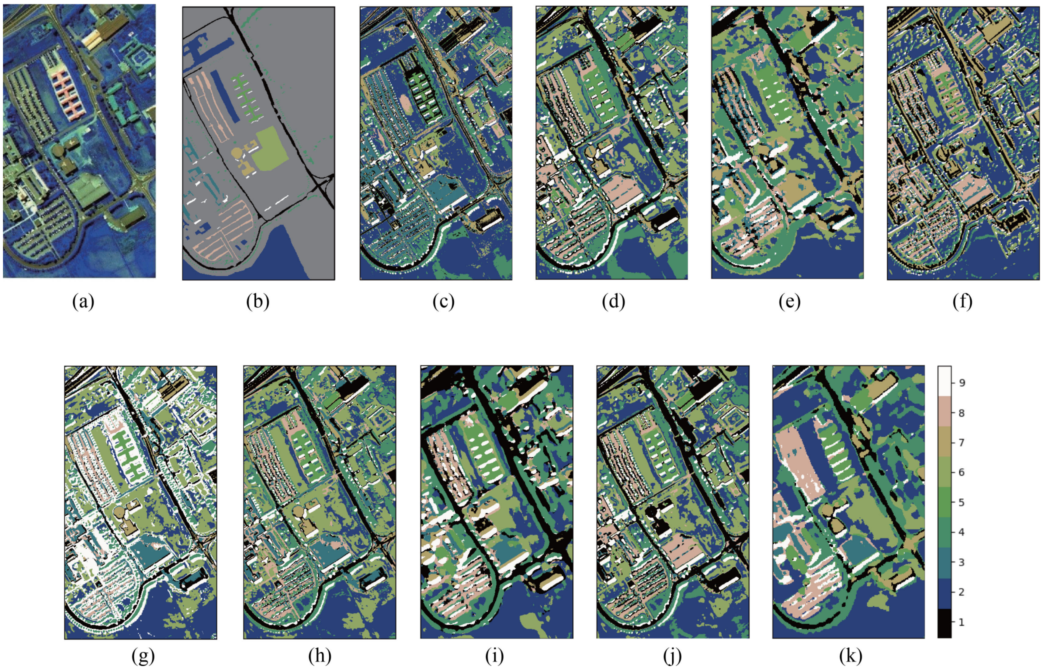


Fig. 8. Global classification maps resulting from different methods on UP. (a) Pseudo-color image. (b) Ground truth. (c) SVM. (d) 3D-CNN. (e) TL+CRNN. (f) iCapsNet. (g) HSGAN. (h) CNN+GCN. (i) RN-FSC. (j) DFSL+SVM. (k) DIN-SSC.

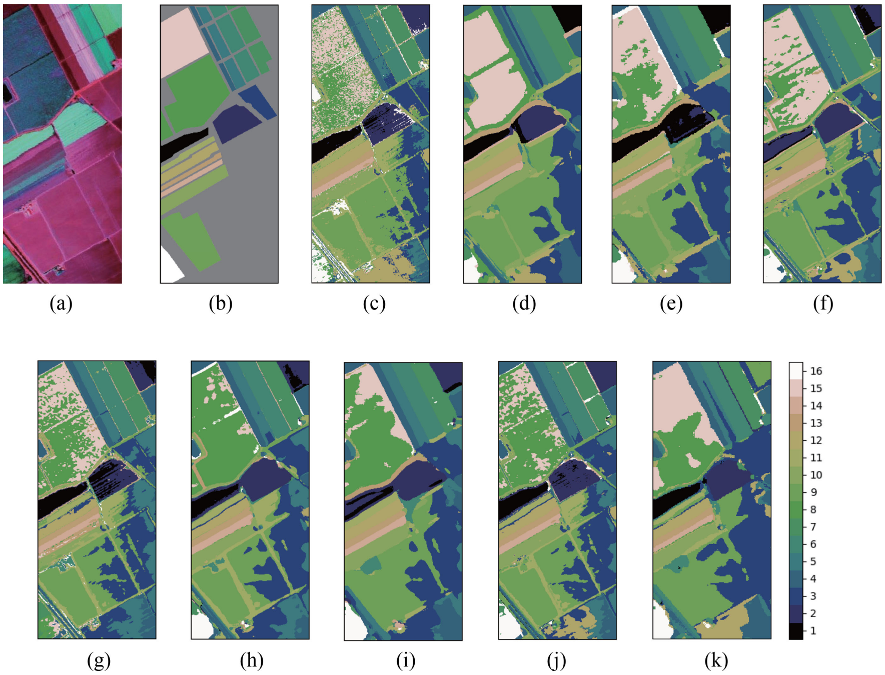


Fig. 9. Global classification maps resulting from different methods on SA. (a) Pseudo-color image. (b) Ground truth. (c) SVM. (d) 3D-CNN. (e) TL+CRNN. (f) iCapsNet. (g) HSGAN. (h) CNN+GCN. (i) RN-FSC. (j) DFSL+SVM. (k) DIN-SSC.

CSA-MSO3DCNN [55]. Compared with MDSFV, the proposed method can obtain comparative classification results while utilizing fewer labeled samples. When the number of labeled samples is fixed at 100, the proposed method can achieve better classification results than IEPF-G-c. When the number of labeled samples is fixed at 200, the difference between the proposed method and CSA-MSO3DCNN is only 0.05% according to the OA.

In addition, we explore the performance of DIN-SSC with unbalanced sampling. DIN-SSC utilizing fewer labeled samples (4%) achieves the comparative results to that of CSVM-MSS (5%) [56], which again proves its effectiveness in HSI classification. Generally speaking, DIN-SSC can obtain competitive results compared with the state-of-the-art methods, which indicates that the classification performance of the proposed method is not limited to the condition of small samples.

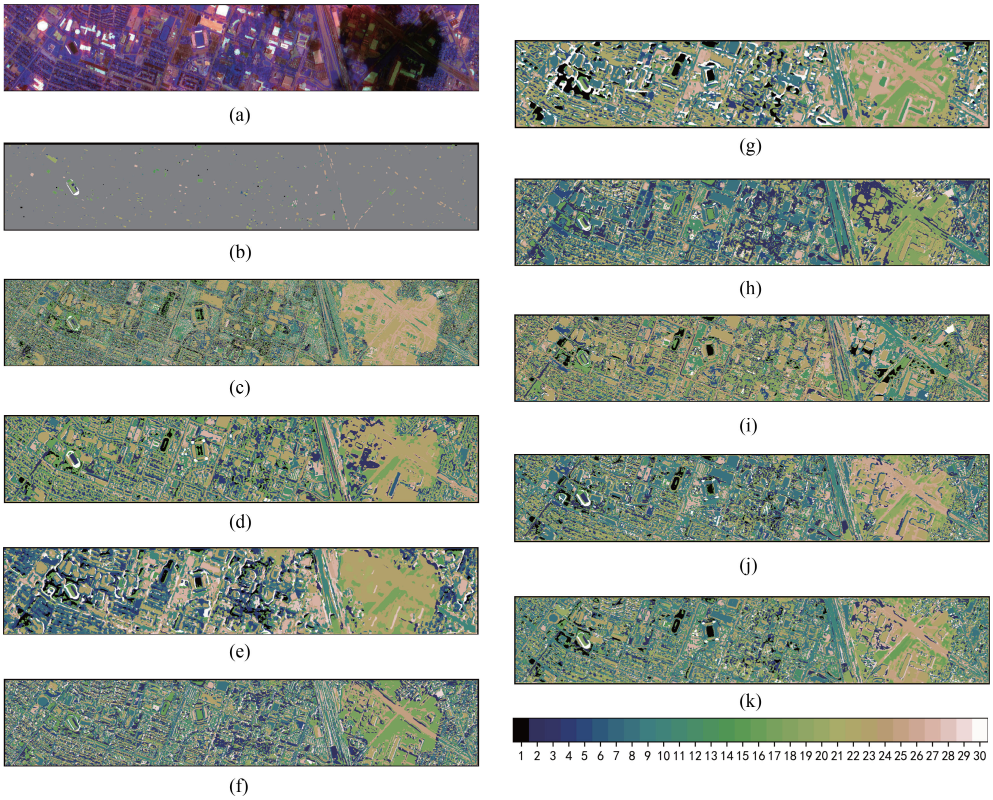
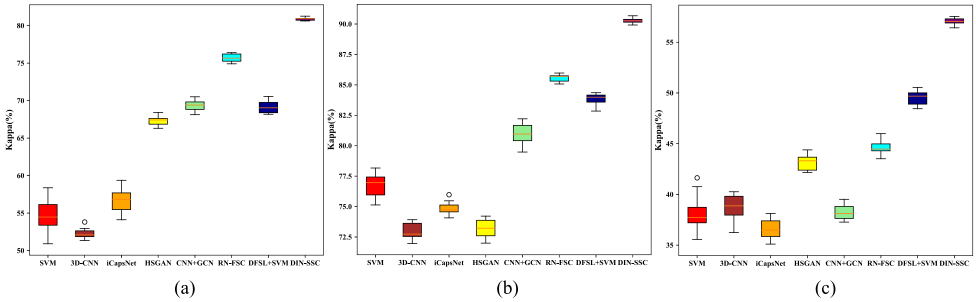
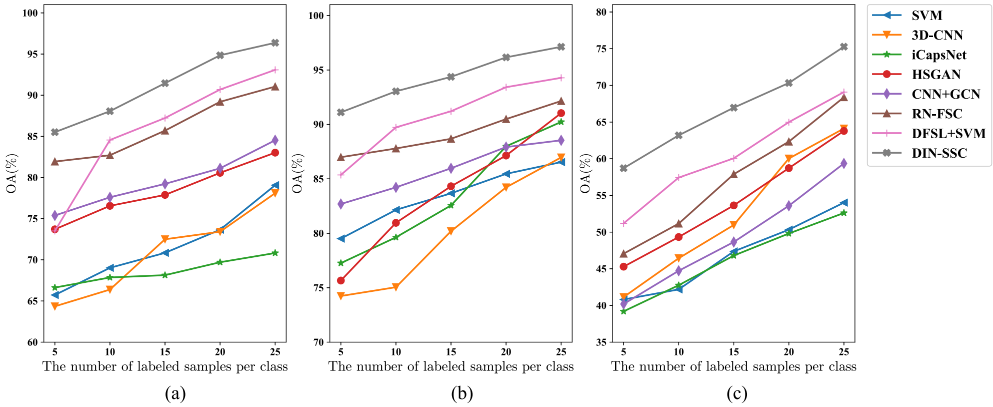

(a)
(b)
(c)
(d)
(e)
(f)
(g)
(h)
(i)
(j)
(k)

Fig. 10. Global classification maps resulting from different methods on HS. (a) Pseudo-color image. (b) Ground truth. (c) SVM. (d) 3D-CNN. (e) TL+CRNN. (f) iCapsNet. (g) HSGAN. (h) CNN+GCN. (i) RN-FSC. (j) DFSL+SVM. (k) DIN-SSC.



(a)
(b)
(c)

Fig. 11. Boxplots of Kappa of different methods. (a) UP. (b) SA. (c) HS.

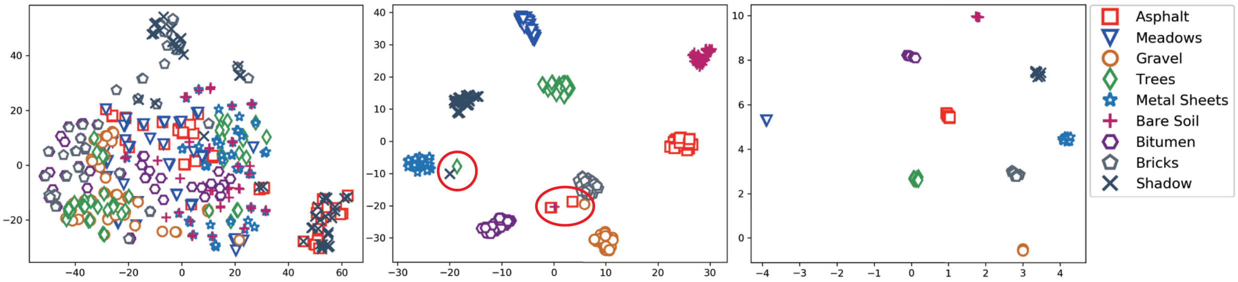


(a)
(b)
(c)

Fig. 12. OA of the different methods with a different number of labeled samples. (a) UP. (b) SA. (c) HS.

TABLE IX
CLASSIFICATION RESULTS OF THE DIFFERENT METHODS ON UP

	MDSFV	IEPF-G-c	SVM+SCNN	CSA-MSO3DCNN	DIN-SSC (50)	DIN-SSC (100)	DIN-SSC (200)	CSVM-MSS	DIN-SSC (4%)
Number of labeled samples	100 samples per class	100 samples per class	200 samples per class	200 samples per class	50 samples per class	100 samples per class	200 samples per class	5% samples	4% samples
OA	98.50	99.05	99.08	99.76	98.23	99.23	99.71	99.38	99.43
AA	98.94	99.38	99.08	99.66	97.69	99.22	99.69	99.33	99.26
Kappa	98.01	98.85	98.79	99.67	97.66	98.98	99.61	99.17	99.25



(a) (b) (c)

Fig. 13. Illustration of the effectiveness of the class-wise induction module. (a) The input data cubes. (b) The sample-wise representations generated by the feature embedding module. (c) The class-wise level representations generated by the class-wise induction module.

TABLE X
EXECUTION TIMES OF DIFFERENT METHODS ON THE THREE HSI (FIVE LABELED SAMPLES PER CLASS ARE USED AS THE SUPERVISED SAMPLES)

Target HSI	execution time	RN-FSC	DFSL+SVM	DIN-SSC
UP	meta-learning	12.83 *h*	106.2 *min*	156.64 *min*
	fine tuning	217.27 *s*	46.49 *s*	161.43 *s*
	classification	72.57 *s*	9.22 *s*	51.25 *s*
SA	meta-learning	12.83 *h*	106.2 *min*	156.64 *min*
	fine tuning	632.98 *s*	57.96 *s*	274.33 *s*
	classification	81.23 *s*	16.03 *s*	51.13 *s*
HS	meta-learning	12.83 *h*	106.2 *min*	156.64 *min*
	fine tuning	1305.68 *s*	90.68 *s*	503.19 *s*
	classification	25.36 *s*	2.33 *s*	13.33 *s*

F. *Effectiveness of the Class-Wise Induction Module*

The meta-learning-based classification methods represented by RN-FSC and DFSL+SVM perform the comparison and calculation at the sample-wise level, which is very susceptible to the phenomenon that the samples belonging to the same class possess the different characteristics, and the samples belonging to different class possess similar characteristics in HSI. The processing for the support samples in RN-FSC and DFSL+SVM is limited to the sum operations or the average operations. In fact, such simple algebraic operations inevitably lose important discriminant information and are easily interfered with the useless information. Compared with RN-FSC and DFSL+SVM, the biggest advantage of DIN-SSC is actually the class-wise induction module, which can induce the class-wise level representations from the sample-wise representations utilizing the dynamic routing algorithm, so as to improve the accuracy and robustness of the classification results. Specifically, the class-wise representations generated by the class-wise induction module are characterized by the aggregation of the samples from the same class, and the separation of the samples from the different classes. As a result, the classes of the query samples can be determined more accurately by comparison with the class-wise level representations in the deep metric space.

In order to illustrate the effectiveness of the class-wise induction module more intuitively, we carried out the visualization experiments on UP. As shown in Fig. 13, the t-SNE algorithm [57] is utilized to visualize the input data cubes, the sample-wise representations generated by the feature embedding module and the class-wise level representations generated by the class-wise induction module. It is easy to see that there is still classes confusion (the red circles) in Fig. 13(b), which undoubtedly leads to the misclassification. Compared with Fig. 13(b), the class-wise level representations in Fig. 13(c) are more clustered, which is conducive to determine the classes of the query samples more accurately. In summary, the visualization experiments directly demonstrate the effectiveness of the class-wise induction module in improving the classification accuracy.

G. *Execution Time Analysis*

In this section, we compare the execution efficiency of RN-FSC, DFSL+SVM, and DIN-SSC. Table X shows the execution time of the three methods on the three HSI. As we can see, all the three methods can be divided into three phases: meta-learning, fine-tuning (training), and classification. Compared to the latter two phases, the meta-learning phase takes a longer time because a large number of tasks are generated to enable the model to be fully trained. RN-FSC possesses the longest execution time due to the 3-D convolution operations and the complex concatenations in its feature extraction phase. DFSL+SVM utilizes SVM

TABLE XI
STATISTICS OF PARAMETERS AND FLOPS OF THE PROPOSED METHOD DURING META-LEARNING

Target HSI	parameters	FLOPs
UP	493729	42206899200
SA	493729	42206899200
HS	493729	63310348800

as classifier which can avoid the complex network computation to some extent. Therefore, the fine tuning time and classification time of DFSL+SVM are the shortest. In general, the proposed method is superior to RN-FSC but inferior to DFSL+SVM in execution efficiency. However, considering the significant improvement of DIN-SSC in classification accuracy, such a time cost is acceptable.

In addition, we count the number of parameters and FLOPs of DIN-SSC during meta-learning. Table XI shows the results. We employ the model with the same structure for the three different HSI, so the number of parameters in Table XI is all 493729. The optimal task setting is (30-way 5-shot 15-query) for HS, while the optimal task settings for the other three HSI are (20-way 5-shot 15-query). It means that there is a significant increase in the amount of data going through the network. Therefore, when classifying HS, the meta-learning phase requires more computation and time.

H. Discussion

When performing the SSC of HSI, deep learning models are often difficult to obtain satisfactory results due to the severe overfitting. To address the problem, some researchers have attempted to improve the classification accuracy using the idea of meta-learning. The designed model (DIN-SSC), also based on the meta-learning framework, can achieve better classification results with a few labeled samples available. The main reasons can be analyzed from the following two aspects.

First, the importance of meta-learning. The designed model first performs meta-learning on the precollected HSI, to acquire sufficient transferable knowledge and cultivate the ability to learn how to learn. Specifically, the typical task-based training strategy is employed to simulate the condition of HSI SSC. As a result, the model can acquire stronger generalization ability and identify new classes with only a few labeled samples. Furthermore, in order to adapt to the characteristics of HSI, a deep 3-D residual network is designed for meta-learning, making full use of the spatial-spectral information while effectively improving the computational efficiency.

Second, the further improvement owing to the induction of the class-level representations. The sample-wise representations obtained in the feature extraction phase are susceptible to the common phenomenon in HSI that the samples from the same class possess the different characteristics, and the samples from different class possess similar characteristics. The class-wise induction module can induce the sample-wise representations to the class-wise level representations possessing better separability in the metric space, to further improve the accuracy and robustness of the classification results.

The experimental results on the three HSI demonstrate that the designed model has higher classification accuracy and better robustness compared with the existing deep learning models. Moreover, the boxplots show that the observed improvement is statistically significant. The classification accuracy is gradually increasing as the number of labeled samples increases, indicating that the designed model has a good adaptability to the number of labeled samples and is not limited to the small samples condition. Finally, the execution time of different methods are compared and analyzed, showing that the proposed method is more efficient.

IV. CONCLUSION

In order to further improve the classification accuracy of HSI under the condition of small samples, this article design a deep model based on the induction network and train it with the typical meta-training strategy. The meta-learning phase on the precollected HSI can enable the designed model to acquire rich transferable knowledge and stronger generalization ability, to identify the new classes more accurately with only a few labeled samples. The obtained class-wise level representations are characterized by the aggregation of the samples from the same class, and the separation of the samples from the different classes, allowing the model to generate the more accurate and robust classification results. The experiments on three public HSI demonstrate that our model possesses better classification performance than existing deep learning models under the condition of small samples.

ACKNOWLEDGMENT

The authors would like to thank Prof. P. Gamba and Dr. N. Yokoya for providing the HSI used in this article. The authors would also like to thank all the researchers for kindly providing the codes associated with the experiments.

REFERENCES

[1] L. He, J. Li, C. Liu, and S. Li, “Recent advances on spectral–spatial hyperspectral image classification: An overview and new guidelines,” *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 3, pp. 1579–1597, Mar. 2018.

[2] A. Agarwal, T. El-Ghazawi, H. El-Askary, and J. Le-Moigne, “Efficient hierarchical-PCA dimension reduction for hyperspectral imagery,” in *Proc. IEEE Int. Symp. Signal Process. Inf. Technol.*, Dec. 2007, pp. 353–356.

[3] C. Li, H. Chu, B. Kuo, and C. Lin, “Hyperspectral image classification using spectral and spatial information based linear discriminant analysis,” in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2011, pp. 1716–1719.

[4] S. Jia, J. Hu, J. Zhu, X. Jia, and Q. Li, “Three-dimensional local binary patterns for hyperspectral imagery classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 4, pp. 2399–2413, Apr. 2017.

[5] P. Quesada-Barriuso, F. ArgÃello, and D. B. Heras, “Spectral–spatial classification of hyperspectral images using wavelets and extended morphological profiles,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 4, pp. 1177–1185, Apr. 2014.

[6] J. Fan, H. L. Tan, M. Toomik, and S. Lu, “Spectral-spatial hyperspectral image classification using super-pixel-based spatial pyramid representation,” *Proc. SPIE*, vol. 10004, pp. 315–321, 2016.

[7] S. Sun, P. Zhong, H. Xiao, F. Liu, and R. Wang, “An active learning method based on SVM classifier for hyperspectral images classification,” in *Proc. 7th Workshop Hyperspectral Image Signal Process.: Evol. Remote Sens.*, Jun. 2015, pp. 1–4.

[8] Y. Ren, Y. Zhang, W. wei, and L. Li, “A spectral-spatial hyperspectral data classification approach using random forest with label constraints,” in *Proc. IEEE Workshop Electron., Comput. Appl.*, May 2014, pp. 344–347.

[9] A. Samat, P. Du, S. Liu, J. Li, and L. Cheng, "$\mathrm{E^2LMs}$: Ensemble extreme learning machines for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 4, pp. 1060–1069, Apr. 2014.

[10] 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.

[11] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2094–2107, Jun. 2014.

[12] B. Liu, X. Yu, P. Zhang, X. Tan, A. Yu, and Z. Xue, "A semi-supervised convolutional neural network for hyperspectral image classification," *Remote Sens. Lett.*, vol. 8, no. 9, pp. 839–848, 2017.

[13] L. Mou, P. Ghamisi, and X. X. Zhu, "Deep recurrent neural networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3639–3655, Jul. 2017.

[14] A. Mughees and L. Tao, "Multiple deep-belief-network-based spectral-spatial classification of hyperspectral images," *Tsinghua Sci. Technol.*, vol. 24, pp. 183–194, Apr. 2019.

[15] H. Petersson, D. Gustafsson, and D. Bergstrom, "Hyperspectral image analysis using deep learning – A review," in *Proc. 6th Int. Conf. Image Process. Theory, Tools Appl.*, 2016, pp. 1–6.

[16] Y. Xu, B. Du, F. Zhang, and L. Zhang, "Hyperspectral image classification via a random patches network," *ISPRS J. Photogrammetry Remote Sens.*, vol. 142, pp. 344–357, 2018.

[17] J. Zhu, L. Fang, and P. Ghamisi, "Deformable convolutional neural networks for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 8, pp. 1254–1258, Aug. 2018.

[18] X. Zhao, Y. Liang, A. J. Guo, and F. Zhu, "Classification of small-scale hyperspectral images with multi-source deep transfer learning," *Remote Sens. Lett.*, vol. 11, no. 4, pp. 303–312, 2020.

[19] M. Paoletti, J. Haut, J. Plaza, and A. Plaza, "Deep learning classifiers for hyperspectral imaging: A review," *ISPRS J. Photogram. Remote Sens.*, vol. 158, pp. 279–317, 2019.

[20] B. Fang, Y. Li, H. Zhang, and J. Chan, "Hyperspectral images classification based on dense convolutional networks with spectral-wise attention mechanism," *Remote Sens.*, vol. 11, Jan. 2019, Art. no. 159.

[21] W. Wang, S. Dou, and S. Wang, "Alternately updated spectral–spatial convolution network for the classification of hyperspectral images," *Remote Sens.*, vol. 11, Jul. 2019, Art. no. 1794.

[22] Z. Li, L. Huang, and J. He, "A multiscale deep middle-level feature fusion network for hyperspectral classification," *Remote Sens.*, vol. 11, Mar. 2019, Art. no. 695.

[23] J. Yin, S. Li, H. Zhu, and X. Luo, "Hyperspectral image classification using capsnet with well-initialized shallow layers," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 7, pp. 1095–1099, Jul. 2019.

[24] B. Liu, X. Yu, P. Zhang, A. Yu, Q. Fu, and X. Wei, "Supervised deep feature extraction for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 4, pp. 1909–1921, Apr. 2018.

[25] L. Zhang, L. Zhang, and B. Du, "Deep learning for remote sensing data: A technical tutorial on the state-of-the-art," *IEEE Geosci. Remote Sens. Mag.*, vol. 4, no. 2, pp. 22–40, Jun. 2016.

[26] J. M. Haut, M. E. Paoletti, J. Plaza, A. Plaza, and J. Li, "Hyperspectral image classification using random occlusion data augmentation," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 11, pp. 1751–1755, Nov. 2019.

[27] W. Li, G. Wu, F. Zhang, and Q. Du, "Hyperspectral image classification using deep pixel-pair features," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 2, pp. 844–853, Feb. 2017.

[28] Y. Wu, G. Mu, C. Qin, Q. Miao, W. Ma, and X. Zhang, "Semi-supervised hyperspectral image classification via spatial-regulated self-training," *Remote Sens.*, vol. 12, Jan. 2020, Art. no. 159.

[29] F. Li, D. A. Clausi, L. Xu, and A. Wong, "ST-IRGS: A region-based self-training algorithm applied to hyperspectral image classification and segmentation," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 1, pp. 3–16, Jan. 2018.

[30] H. Wang, C. Tao, J. Qi, H. Li, and Y. Tang, "Semi-supervised variational generative adversarial networks for hyperspectral image classification," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2019, pp. 9792–9794.

[31] Y. Zhan, D. Hu, Y. Wang, and X. Yu, "Semisupervised hyperspectral image classification based on generative adversarial networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 2, pp. 212–216, Feb. 2018.

[32] J. Qin *et al.*, "Semi-supervised classification of hyperspectral data for geologic body based on generative adversarial networks at Tianshan area," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2018, pp. 4776–4779.

[33] F. F. Shahraki and S. Prasad, "Graph convolutional neural networks for hyperspectral data classification," in *Proc. IEEE Global Conf. Signal Inf. Process.*, 2018, pp. 968–972.

[34] A. Qin, Z. Shang, J. Tian, Y. Wang, T. Zhang, and Y. Y. Tang, "Spectral–spatial graph convolutional networks for semisupervised hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 2, pp. 241–245, Feb. 2019.

[35] S. Wan, C. Gong, P. Zhong, B. Du, L. Zhang, and J. Yang, "Multiscale dynamic graph convolutional network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 5, pp. 3162–3177, May 2020.

[36] M. Imani and H. Ghassemian, "Feature extraction using attraction points for classification of hyperspectral images in a small sample size situation," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 11, pp. 1986–1990, Nov. 2014.

[37] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. 34th Int. Conf. Mach. Learning*, 2017, pp. 1126–1135.

[38] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," in *Proc. 30th Conf. Neural Inform. Process. Syst.*, 2016, pp. 3630–3638.

[39] L. Bertinetto, J. F. Henriques, P. Torr, and A. Vedaldi, "Meta-learning with differentiable closed-form solvers," in *Proc. Int. Conf. Learn. Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=HyxnZh0ct7

[40] 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. 2019.

[41] H. Tang, Y. Li, X. Han, Q. Huang, and W. Xie, "A spatial–spectral prototypical network for hyperspectral remote sensing image," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 1, pp. 167–171, Jan. 2020.

[42] 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, Mar. 2020, Art. no. 923.

[43] X. Ma, S. Ji, J. Wang, J. Geng, and H. Wang, "Hyperspectral image classification based on two-phase relation learning network," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 12, pp. 10398–10409, Dec. 2019.

[44] J. Snell, K. Swersky, and R. S. Zemel, "Prototypical networks for few-shot learning," in *Proc. Conf. Neural Inform. Process. Syst.*, 2017, pp. 4080–4090. [Online]. Available: http://papers.nips.cc/paper/6996-prototypical-networks-for-few-shot-learning

[45] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2018, pp. 1199–1208. [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2018/html/Sung_Learning_to_Compare_CVPR_2018_paper.html

[46] R. Geng, B. Li, Y. Li, X. Zhu, P. Jian, and J. Sun, "Induction networks for few-shot text classification," 2019, *arXiv:1902.10482*.

[47] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Proc. Conf. Neural Inform. Process. Syst.*, 2017, pp. 3859–3869. [Online]. Available: http://papers.nips.cc/paper/6975-dynamic-routing-between-capsules

[48] J. Gu **et al.**, "Recent advances in convolutional neural networks," *Pattern Recognit.*, vol. 77, pp. 354–377, 2018, doi: 10.1016/j.patcog.2017.10.013.

[49] W. Sun, L. Zhang, B. Du, W. Li, and Y. Lai, "Band selection using improved sparse subspace clustering for hyperspectral imagery classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 6, pp. 2784–2797, Jun. 2015.

[50] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, "Deep feature extraction and classification of hyperspectral images based on convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 10, pp. 6232–6251, Oct. 2016.

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

[52] B. Liu, X. Yu, A. Yu, and G. Wan, "Deep convolutional recurrent neural network with transfer learning for hyperspectral image classification," *J. Appl. Remote Sens.*, vol. 12, 2018, Art. no. 1.

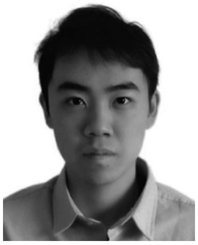
[53] L. Liu, Z. Shi, B. Pan, N. Zhang, H. Luo, and X. Lan, "Multiscale deep spatial feature extraction using virtual RGB image for hyperspectral imagery classification," *Remote Sens.*, vol. 12, Jan. 2020, Art. no. 280.

[54] S. Zhong, C. Chang, and Y. Zhang, "Iterative edge preserving filtering approach to hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 1, pp. 90–94, Jan. 2019.

[55] Q. Xu, Y. Xiao, D. Wang, and B. Luo, “CSA-MSO3DCNN: Multiscale octave 3D CNN with channel and spatial attention for hyperspectral image classification,” *Remote Sens.*, vol. 12, Jan. 2020, Art. no. 188.

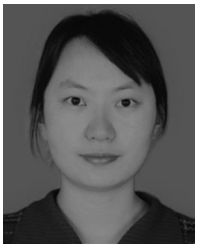
[56] X. Cao, D. Wang, X. Wang, J. Zhao, and L. Jiao, “Hyperspectral imagery classification with cascaded support vector machines and multiscale superpixel segmentation,” *Int. J. Remote Sens.*, vol. 41, no. 12, pp. 4530–4550, 2020.

[57] L. van der Maaten and G. Hinton, “Viualizing data using t-SNE,” *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, 2008.



Kuiliang Gao received the B.S. degree in remote sensing science and technology from the PLA Strategic Support Force Information Engineering University, Zhengzhou, China, in 2019, where he is currently working toward the M.S. degree.

His research interests include hyperspectral image processing, pattern recognition, and deep learning.



Wenyue Guo received the bachelor’s and master’s degree in cartography and geographic information engineering from the PLA Strategic Support Force Information Engineering University, Zhengzhou, China in 2012 and 2015, respectively, and the Ph.D. degree in surveying and mapping from the PLA Strategic Support Force Information Engineering University, Zhengzhou, China, in 2018.

She is currently working with PLA Strategic Support Force Information Engineering University as a Lecturer. Her research interests include geographic information science and graph representation.

Xuchu Yu received the Ph.D. degree from the Institute of Surveying and Mapping, Zhengzhou, China, in 1997.

He is currently working with the PLA Strategic Support Force Information Engineering University as a Professor and a Doctoral Supervisor. His research interests include photogrammetry, remote sensing, and pattern recognition.

Bing Liu received the B.S. degree in measurement and control engineering from the PLA Strategic Support Force Information Engineering University, Zhengzhou, China, in 2013, the M.S. degree in pattern recognition and intelligent system from the PLA Strategic Support Force Information Engineering University, Zhengzhou, China, in 2016, and the Ph.D. degree in surveying and mapping from the PLA Strategic Support Force Information Engineering University, Zhengzhou, China, in 2019.

He is currently working with the PLA Strategic Support Force Information Engineering University as a Lecturer. His research interests include machine learning, pattern recognition, and signal processing in earth observation.

Dr. Liu is an active Reviewer for the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING, IEEE TRANSACTIONS ON CYBERNETICS, IEEE ACCESS, *International Journal of Remote Sensing*, *Remote Sensing Letter, and Journal of Applied Remote Sensing*.

Anzhu Yu received the B.S. degree in remote sensing science and technology the M.S. degree in photogrammetry and remote sensing from the PLA Strategic Support Force Information Engineering University, Zhengzhou, China, in 2011 and 2014, respectively, and the Ph.D. degree from the Institute of Surveying and Mapping, PLA Strategic Support Force Information Engineering University, Zhengzhou, China, in 2018.

He is currently working at the Institute of Surveying and Mapping, PLA Strategic Support Force Information Engineering University as an associate professor. His research interests include signal processing in earth observation.

Xiangpo Wei received the B.S. degree in remote sensing science and technology from the PLA Strategic Support Force Information Engineering University, Zhengzhou, China, in 2012, the M.S. degree in photogrammetric and remote sensing from the PLA Strategic Support Force Information Engineering University, Zhengzhou, China, in 2015, and the Ph.D. degree in surveying and mapping from the PLA Strategic Support Force Information Engineering University, Zhengzhou, China, in 2019.

He is currently working with the PLA Strategic Support Force Information Engineering University as a Lecturer. His research interests include hyperspectral image processing, pattern recognition, and deep learning.