Hyperspectral Image Classification Based on Two-Phase Relation Learning Network

Xiaorui Ma, *Member, IEEE*, Sheng Ji, Jie Wang, *Senior Member, IEEE*, Jie Geng, and Hongyu Wang

***Abstract*— Deep learning-based classification methods are competent to achieve an excellent performance under one necessary condition, i.e., there are sufficient labeled samples in each class, which is extremely impractical in most of the remote sensing tasks. To improve the performance with small training sets, we resort to other hyperspectral images and design a two-phase relation learning network that can be transferred between different images for general information sharing and fine-trained on a specific hyperspectral image for individual information learning. Specifically, we use a relation learning method to compare samples and deal with the task inconsistency between different data sets, and we adopt an episode-based training strategy to mimic the testing setup and learn the transferable comparison ability. Benefited from these two strategies, the proposed network takes the advantage of extra knowledge for information supplement and learns to compare rather than to classify for information exploration, which guarantees a reasonable performance even with small training sets. Extensive experiments and analysis on three benchmarks demonstrate that the proposed method can provide an effective solution for hyperspectral image classification with small training sets, which makes it possible to work on large-scale applications of earth observation with less effort on field investigation.**

***Index Terms*— Classification, deep network, hyperspectral image.**

I. Introduction

DUE to the abundant spatial and spectral information, hyperspectral images show superiority in lots of remote sensing tasks, such as land-use investigation, environment monitoring, and mineral exploration [1]. Hyperspectral image classification, which labels each sample a category tag according to its spectral and spatial information, is a crucial problem in lots of remote sensing applications, e.g., land-use investigation needs to classify the land-cover type in advance, environment monitoring should recognize the vegetation species before assessment, and mineral exploration has to rely on the precise mineral distribution map [2], [3]. As a fundamental and critical problem, hyperspectral image classification plays an important role in every regard of earth observation mission. Therefore, plenty of methods have been proposed during the past decade [4]–[8] and refreshed the classification accuracy constantly. However, the superior performance of these methods heavily relies on one condition, i.e., there are sufficient labeled samples in every class, which is extremely expensive to satisfy and even infeasible in practice. Therefore, realizing classification with small training sets becomes an urgent demand for remote sensing applications.

To improve the classification performance with small training sets, researchers have conducted valuable explorations. Zhang *et al.* [9] combined the nearest neighbor graph to take the advantage of space features to solve the paucity problem of the labeled samples, and Liu *et al.* [10] use limited labeled data and abundant unlabeled data to train efficient classifiers with small training sets. These methods either attempt to better explore the information involved in the insufficient labeled samples with prior knowledge or resort to semisupervised learning to supplement the information with unlabeled samples, which all focus on how to take full advantage of the data set being processed, i.e., the target data set. Interestingly, a human can recognize an object accurately after seeing only a few examples in the same category by transferring the ability of recognizing other objects [11], which indicates that the knowledge learned from other data sets should be enlightening for further classification. Moreover, for large-scale remote sensing applications, training a new classifier for each data set seems to be inefficient and infeasible. Inspired by the above-mentioned observations, we think it is possible and necessary to develop a new classification method to further improve the classification performance with small training sets by borrowing the knowledge of other hyperspectral images. This proposal can boost the training efficiency and narrow down the gap between the classification theory and the large-scale applications by sharing the information of different data sets.

In order to use the knowledge of other hyperspectral images for information compensation, the proposed method should be able to transfer between different data sets, which may

Manuscript received November 5, 2018; revised February 25, 2019 and May 14, 2019; accepted August 6, 2019. This work was supported in part by the National Natural Science Foundation of China under Grant 61801078, Grant 61671103, and Grant 61671102, in part by the China Postdoctoral Science Foundation under Grant 2018M630288, in part by the Fundamental Research Funds for the Central Universities under Grant DUT19RC(4)018 and Grant 3132019216, in part by the Liaoning Province Natural Science Foundation under Grant 20180520026, in part by the Dalian High-level Talent Innovation Support Program Project under Grant 2017RQ096, and in part by the Dalian Science and Technology Innovation Foundation under Grant 2018J12GX044. *(Corresponding author: Xiaorui Ma.)*

X. Ma, S. Ji, and H. Wang are with the School of Information and Communication Engineering, Dalian University of Technology, Dalian 116024, China (e-mail: maxr@dlut.edu.cn; jisheng@mail.dlut.edu.cn; whyu@dlut.edu.cn).

J. Wang is with the Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian 116024, China, and also with the School of Information Science and Technology, Dalian Maritime University, Dalian 116026, China (e-mail: wangjie@dlut.edu.cn).

J. Geng is with the School of Electronics and Information, Northwestern Polytechnical University, Xi'an 710068, China (e-mail: gengjie@nwpu.edu.cn).

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

Digital Object Identifier 10.1109/TGRS.2019.2934218

be covered by different quantities and types of land covers. Thus, the proposed method should focus on finding out the transferable information, learning to explore the transferable information, and transferring the information to the target data set, i.e., what to learn, how to learn, and how to transfer. To this end, we propose a two-phase relation learning network (RL-Net) with episode-based training strategy, which learns the comparison ability on other hyperspectral images, i.e., the source data set, so that it reduces the difficulty of transferring knowledge between different hyperspectral images. Especially, we design a two-phase RL-Net with both feature embedding module and relation learning module to learn the transferable comparison ability and use an episode-based training strategy to mimic the test setting and transfer the learned knowledge to the target data set. With the aforementioned strategies, the proposed method is able to take advantage of the knowledge learned from other hyperspectral images to improve the classification performance with small training sets.

The main contributions can be summarized as follows.

1) We propose the idea of taking advantage of other hyperspectral images to improve the classification performance with small training sets. It uses meta-learning on the source data set to learn transferable information, uses fine-training on the target data set to extract individual information, and gives solutions for a transfer-based classification method on what to learn, how to learn, and how to transfer.
2) We propose a two-phase RL-Net to learn the comparison ability, which extracts spatial–spectral information for each hyperspectral samples in a feature embedding module and learns the relation between different samples in a relation learning module. It learns the transferable ability of comparing samples, which makes it feasible to transfer between different data sets even covered by different cover types.
3) We propose to use the episode-based training strategy to transfer the learned knowledge and relieve the overfitting problem, which subsamples the training sets into small batches to mimic the testing environment. This strategy makes the trained model more faithful to the testing environment, thereby improves generalization ability.

The remainder of this article is organized as follows. Section II reviews the related work on hyperspectral image classification. Section III gives the detailed implementation of the two-phase RL-Net. Section IV presents the framework and training strategy of the proposed classification method. Section V validates the proposed method and presents the corresponding analysis. Conclusions are finally summarized in Section VI.

II. Related Work

The idea of hyperspectral image classification starts from spectral classification method, which takes the original spectral data (i.e., digital number) or the spectral features as input and performs classification with spectral information only [12]. Support vector machine and the variations are the most representative spectral classification methods [13], [14]. With the development of hyperspectral imagers, the spatial resolution of hyperspectral images evolves from hundreds of meters to dozens of centimeters, which makes spatial information more valuable [3]. Multiple kernel-based classifier [4], [15], sparse representation-based classifier [16], [17], and the Markov random field-based classifier [18], [19] have proven that spatial information is as much important as spectral information. Therefore, nowadays, spectral–spatial classifiers are dominant in hyperspectral image classification.

As all classification tasks, how to represent the features of hyperspectral images is also an important research topic. Extensive works have been studied on feature representation. Principal component analysis [5], minimum noise fraction [6], and attribute profiles [20] are widely used feature representation methods. However, they are hand-crafted or mapping-based features. Some of them have unmanageable parameters that depend heavily on the knowledge about the data set and the experience of the expert [21]–[24]. Deep learning, which learns to represent data by a hierarchical deep network, is able to generate nonlinear and abstract representation without human interference [25]. Inspired by the excellent feature extraction ability of deep learning, some works try to introduce deep learning for hyperspectral feature representation [26]–[28]. They try to design a new deep architecture according to the characteristic of hyperspectral image, Gong *et al.* [29] improve the CNN with multiscale convolution and diversified metric, Hao *et al.* [30] propose a two-stream deep network, Jiao *et al.* [31] design a fully convolutional network, and Zhong *et al.* [32] develop a 3-D network. Some others rearrangement the training strategy to adjust deep networks so as to make it more suitable for hyperspectral image classification task, Li *et al.* [33] propose pixel-pair strategy to learn the large size of parameters in deep neural networks, Lee and Kwon [34] train a deeper network by a multi-scale convolutional filter bank, and Zhou *et al.* [35] take the cotraining strategy to train deep networks in semisupervised way. These works learn features or perform feature optimization and show the feasibility and effectiveness of using deep learning in hyperspectral image classification. However, the performance of the aforementioned methods heavily relies on the sufficient labeled samples, which might be extremely expensive or infeasible in practice. Thus, the overfitting problem troubles deep network and limits its learning ability. In summary, classification with small training sets is still a challenging research topic, which is the objective of our work.

There are very few works on classification with small training sets in hyperspectral image classification area, but some related works on computer vision area may be enlightening. The fundamental problem of classification with small training sets is the scarcity of labeled samples, and a direct solution to supply labeled samples is data augmentation, such as generating new samples using generative adversarial networks [36]. However, directly augmenting samples may not increase feature discrimination. Some works try to solve classification with small training sets by improving supervised classifiers by instance-based learning or deep generative models [37], but training a supervised classifier with only a few labeled samples is difficult. The aforementioned methods only

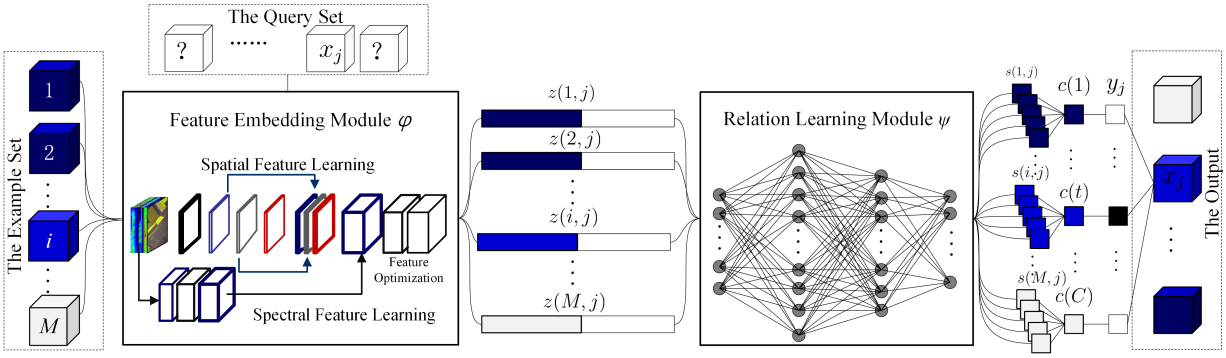


Fig. 1. Two-phase RL-Net that learns to represent and compare the samples from the example set and the query set. It takes all functions into one deep architecture that is made up of a feature embedding module and a relation learning module.

consider exploiting the information of the target data sets, which is restricted by the limited labeled samples. Recently, some works borrow information from other data sets to develop transfer-learning-based approaches, which has shown great potentials [38]. In particular, some of these transfer-learning-based methods use metric learning to transfer knowledge from other hyperspectral images to the target data set for information compensation, which can improve the classification performance remarkably with few samples [39], [40]. Inspired by the aforementioned works, we want to develop a transfer-based method with metric learning to improve the classification with small training sets.

Conventional metric learning approaches focus on learning metric from the embedding space, which assumes that features are linearly separable and can be compared elementwise. However, the hyperspectral samples from different sensors may not share the same spectral resolution, which violates the elementwise comparison assumption. To this end, we design a new metric learning method for transfer-based classification of hyperspectral images. It is based on the deep RL-Net that uses the cascaded features as input, learns to compare these features, and then outputs the relation score. Since the proposed method learns to compare instead of classifying, it has the following advantages: 1) it is not restricted by the number of classes in the source data set and 2) it does not require the features that must be elementwise comparable.

III. Two-Phase Relation Learning Network

This article proposes a two-phase RL-Net with episode-based training strategy for hyperspectral image classification with small training sets, which is a common situation in lots of remote sensing applications. It is based on the architecture of the deep network to learn the ability to represent and compare and takes the advantage of metalearning on a source data set to learn better. This section presents the two-phase RL-Net.

A. Network Architecture

As shown in Fig. 1, the proposed network is a two-phase deep architecture, which consists of two modules: a feature embedding module and a relation learning module. Suppose $\boldsymbol{x}$ is the input sample, which is a spectral vector or a small patch of the hyperspectral image, and $\boldsymbol{y}$ is the corresponding vectorized label, which uses the location of each element to indicate its label information. An example set is denoted as $E = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{M}$, and a query set is denoted as $Q = \{(\boldsymbol{x}_j, \boldsymbol{y}_j)\}_{j=1}^{N}$. In particular, during testing, there is no labeled information in the query set. During training, both of them are built by the randomly selected samples from the training set. The proposed network takes the samples from both the example set and the query set as input and then tries to learn the relationship between each query sample and each example. First, it takes all samples $\{\boldsymbol{x}_i\}_{i=1}^{M}$ of E and one sample $\boldsymbol{x}_j$ in Q and feed them into the feature embedding module with embedding function φ to produce spectral–spatial features $\boldsymbol{r}_i = \varphi(\boldsymbol{x}_i)$ and $\boldsymbol{r}_j = \varphi(\boldsymbol{x}_j)$, for $i = 1, \ldots, M$. Second, the embedding feature of each example is concatenated with the feature of the query sample, $\boldsymbol{z}(i, j) = (\boldsymbol{r}_i; \boldsymbol{r}_j)$, for $i = 1, \ldots, M$. Third, it learns to compare features by a relation learning module with function ψ to learn the relation score $s(i, j) = \psi(\boldsymbol{z}(i, j))$, for $i = 1, \ldots, M$. Finally, the label of the query sample $\boldsymbol{x}_j$ is determined by the example with the highest relation score with the query sample. We will give detailed information about how to learn the spectral–spatial features for each sample and the relation metric between every two samples in Sections III-B and III-C.

B. Feature Embedding Module

Feature embedding module is designed for feature learning, which projects the original input into a new space, where the representation should contain more discriminative information. Considering the characteristic of the hyperspectral image, we design a deep architecture to learn spectral and spatial features. As mentioned earlier, we denote the feature embedding as one function φ, which takes sample $\boldsymbol{x}$ as input and generates embedding features $\varphi(\boldsymbol{x})$. A hyperspectral image is a data cube with both spatial and spectral information; every operation on the spatial domain will affect the spectral information and causes distortions, vice versa. Therefore, we use a deep spectral–spatial feature learning architecture that learns spatial information by a contextual layer with only one kernel, and we learn spectral information with multiple fully connected layers.

All feature learning networks that are able to produce pixelwise features can be adopted in our framework, such as the two-branch network with skip architectures. Without loss of generality, we take a simple contextual deep network to present the working procedure of the proposed method. For spatial feature learning, we design a convolutional layer that takes small patches as inputs and performs spatial filtering. More specially, in order to keep the fidelity of spectral information, all bands are processed by the same kernel. For spectral features learning, we use multiple fully connected layers that use autoencoders to initialize all parameters instead of random initialization. This operation will give a better initialization, and the whole network will converge after a few iterations. The input of this branch is the spectral vector, which is the pixel of the hyperspectral image. After learning spectral and spacial features, spectral–spatial features for all samples are ready for the next relation learning module. Henceforth, the embedding function φ is related to the activation function, the convolutional kernel, all weights of the fully connected layer, and the bias of all layers.

C. Relation Learning Module

Through the feature embedding module, we can get the spectral–spatial features for all samples of the example set and the query set. Suppose samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are from the example set and the query set, respectively, $\boldsymbol{r}_i = \varphi(\boldsymbol{x}_i)$ and $\boldsymbol{r}_j = \varphi(\boldsymbol{x}_j)$ are the corresponding features, and the concatenated feature of the query sample and the example is denoted as $\boldsymbol{z}(i, j) = (\boldsymbol{r}_i; \boldsymbol{r}_j)$, which is taken as the input of the relation learning module. For each layer of this module, we take the output of the previous layer as input and project it into a new space by an active function with a weight matrix and bias. These fully connected layers optimize the concatenated features according to the loss function and finally project the contented features into a relation score between the query sample and the example $s(i, j)$. As mentioned earlier, we denote the embedding function of the relation learning module as ψ; then, the relation score is

$$s(i, j) = \psi(\boldsymbol{z}(i, j)). \tag{1}$$

If there is more than one example for each class in the example set, the relation score from the same class will be averaged to get a class relation score $c(t, j)$, which indicates the relation measurement between class t and query sample j. The query sample should be a member of the most related class, which has the maximum of class relation scores.

After building all layers, we get a complete two-phase RL-Net. During the training process, the network learns all parameters according to a loss function; during the testing process, it outputs the label of testing samples by maximizing the relation score. A fixed classifier requires consistence of cover types between different data sets, which is impossible for the hyperspectral images captured over different scenes. However, the proposed two-phase RL-Net learns to compare other than to classify. It can be transferred between different data sets and is ready for taking source data set for information compensation. The detail information of training the whole network will be presented in Section IV.

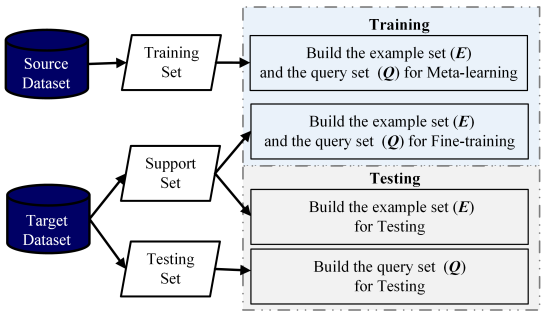


Fig. 2. Set information of the proposed method, which is made up of a training set from the source data set, a support set from the target data set, and a testing set also from the target data set.

IV. Transfer-Based Classification With Relation Learning Network

This article proposes a two-phase RL-Net with an episode-based training strategy, which is essentially a transfer-based classification method that uses the knowledge of the source data set by metalearning. Section III has given the two-phase RL-Net to learn transferable knowledge; this section will present the strategy on how to transfer the network for information compensation.

A. Problem Definition

Considering a transfer-based classification problem, which works on two data sets: a source data set and a target data set, as shown in Fig. 2. Unlike traditional classification, in the proposed transfer-based classification, the training set with label information is from the source data set, i.e., other hyperspectral images whose label space may be disjoint with the target data set. The support set, which also has the label information, is constructed from the target data set. The testing set contains all unknown samples of the target data set, and it shares the same label space with the support set. Like two hyperspectral images, i.e., the Indian Pines data set and the Salinas data set, they are captured by the same sensor but slightly different in the available spectral bands. We can use the Indian Pines data set as the source data set to build the training set and use the Salinas data set as the target data set to build the support set and testing set. We capture a few labeled samples from the Salinas data set to build the support set and classify the remaining unlabeled samples using the support set with the help of the knowledge learned from the Indian Pines data set. Theoretically, we can train a supervised classifier with the support set only and classify the unlabeled samples in the testing set. However, the performance of such a classifier will be far from satisfaction due to the lack of sufficient labeled samples. To this end, we take the advantage of the knowledge learned from other hyperspectral images and perform metalearning on the training set to extract transferable knowledge, which allows us to perform better learning on the support set, and thus classify the testing set more successfully.

B. Classification With Extra Knowledge

A naive way to achieve transfer-based classification is using the training set to pretrain a model and then further training

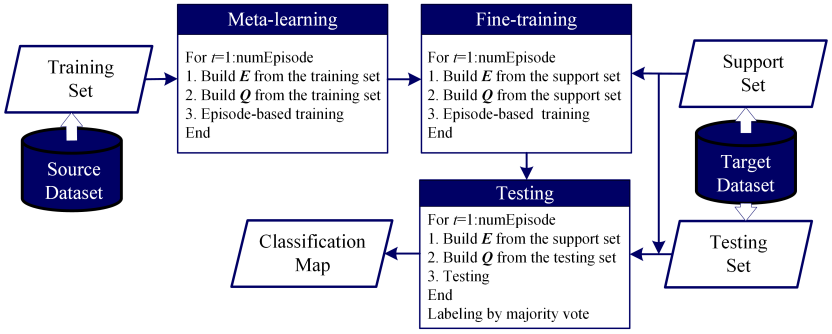


Fig. 3. Framework of the proposed classification method, which is made up of three steps, i.e., metalearning on the training set, fine-training on the support set, and testing on both the support set and the testing set.

it on the support set. However, due to the scarcity of labeled samples in the support set, this kind of methods can be easily troubled by overfitting. To address the aforementioned issues and achieve classification with few labeled samples, we use an episode-based training strategy to transfer the learned knowledge from the source data set to the target data set. It mimics the testing environment by sampling the training set into several episodes and then trains the network on the episodes. The two-phase RL-Net is transferable between the source data set and the target data set, and the episode-based training makes the trained model more faithful to the testing environment, thereby improves generalization.

As shown in Fig. 3, there are three steps to perform our transfer-based classification method. First, metalearning is performed on the training set to learn the general information from the sufficient labeled samples of the source data sets. Then, fine-training is executed on the support set to learn the individual information from the limited labeled samples of the target data set. Finally, testing is carried out on the testing set to estimate the labels by maximizing the relation score. During training, an episode-based training strategy is used to simulate the testing situation. Each episode is made up of an example set and a query set. In each iteration, we train the network by finding the correct relation between an example and a query sample. The source of the episodes in the metalearning and fine training phases is different. The episodes in the metalearning are from the training set that is selected from the source data set, but the episodes in the fine-training are from the support sets that are selected from the target data set. During testing, the example set is from the support set, and the query set is from the testing set. The output label of each testing sample is computed by maximizing the relation score, and the final label is determined by the majority vote of multiple testings.

C. Episode-Based Training Strategy

The episode-based training is used in both the metalearning and the fine-training, and the idea of the episode is used through the whole training procedure. It builds episodes by sampling the training sets into an example set and a query set to mimic the testing situation. Taking the episode-based training in the metalearning as an example, we summarize the whole procedure of the episode-based training in Algorithm 1. Since metalearning is carried on the training set, all the samples in the following steps are from the training set. In each iteration of the episode-based training, we first build an episode with an example set and a query set, and it selects L_1 labeled samples for each of the C classes to build an example set $E = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{M}$, where $M = L_1 \times C$, and L_2 labeled samples for each of the C classes to build the query set $Q = \{(\boldsymbol{x}_j, \boldsymbol{y}_j)\}_{j=1}^{N}$, where $N = L_2 \times C$. Second, for each query sample $\boldsymbol{x}_j$ and each example $\boldsymbol{x}_i$, we feed them through the two-phase RL-Net to learn the relation score $s(i, j) = \psi(z(i, j))$. Third, since the examples with high relation scores should share the same label with the query sample, we compute the related class relation score and decide the final label by maximizing the class relation score. Finally, the whole network is fine-tuned by the following loss function:

$$J = \sum_{j=1}^{N} \|\boldsymbol{y}_j^* - \boldsymbol{y}_j\|_2^2 \tag{2}$$

where $\boldsymbol{y}_j^*$ is the output label. The loss function often comes along with some regulation terms, such as elementwised two-norm to relief overfitting by restricting big parameters and the sparsity constraint to reduce the number of working activations. So far, we gave the information for training the RL-Net with both metalearning on the training set and fine-training on the target data set, which is able to transfer the knowledge from the training set to the target data set to better learn and compare.

There is still one more problem need to be addressed, i.e., how to generate the episode. A naive approach is that to randomly select L_1 samples for each of the C classes to build the example set E and take the rest to construct the query set Q. However, this strategy would increase the computing time of the training process, which is also not necessary. We randomly select a certain number L_2 of labeled samples for each of C classes to build the query set, where $3L_1 \geq L_2 \geq L_1$. This strategy learns to compare multiple query samples at the same time and considers all possible classes within one comparison, which can boost the training efficiency. If there are only several samples in each class, it is not possible to perform fine-training and the network will execute testing directly. Otherwise, we build episodes for fine-training as in the metalearning. Moreover, the number of

Algorithm 1 Episode-Based Training in Pseudocode

Build episode:
Build the example set: $E = \{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^{M}$
Build the query set: $Q = \{\boldsymbol{x}_j, \boldsymbol{y}_j\}_{j=1}^{N}$
For $j = 1, \ldots, N$
Feature embedding of each query sample: $\boldsymbol{r}_j = \varphi(\boldsymbol{x}_j)$
For $i = 1, \ldots, M$
Feature embedding of each example: $\boldsymbol{r}_i = \varphi(\boldsymbol{x}_i)$
Feature catenation: $z(i, j) = (\boldsymbol{r}_i; \boldsymbol{r}_j)$
Relation learning: $s(i, j) = \psi(z(i, j))$
End For
Compute class relation score $c(t, j)$
Labeling by finding the maximum of $c(t, j)$
End For
Fine-tuning the network with the following loss function:
$$J = \sum_{j=1}^{N} \|\boldsymbol{y}_j^* - \boldsymbol{y}_j\|_2^2$$

classes C maybe not equal to the number of classes in the training set C_1, even maybe not equal to the number of classes in the target data set C_2.

V. Experiments and Analysis

In this section, we evaluate the performance of the proposed transfer-based classification method, two-phase RL-Net, using several real hyperspectral data sets, give a detailed analysis of relevant parameters, and present comprehensive comparisons with related methods. All the experiments are implemented with PyTorch toolbox on the platform of a desk computer with Intel Core i7 4.0-GHz CPU, GeForce GTX 1080Ti GPU, and 32-GB memory. Prior to these experiments, we present the data set description and experimental setup.

A. Experimental Data Set and Setup

We use three hyperspectral data sets to evaluate the performance of the proposed method, which are all widely used and can be acquired from the website of Computational Intelligence Group from the Basque University.[1] The detailed information about these data sets is shown in the following.

1) The data set of the Indian Pines scene was gathered by the Airborne Visible/Infrared Imaging Spectrometer over north-western Indiana, USA, in 1992, which contains 145 × 145 pixels with 20-m spatial resolution and 200 bands covering from 400 to 2500 nm after removing 20 water absorption bands and four low-SNR bands. Except for unknown samples (indicated with white color in the ground-truth map), 16 land-cover types are labeled in the ground-truth map, and most of them are corps in different growth phases, which results in similar spectral characters, thus makes the classification of this data set more difficult. The pseudocolor image (composition of band 50, 27, and 17) and the available ground-truth map are shown in Fig. 4.
2) The data set of the Pavia Center scene was acquired by the Airborne Visible/Infrared Imaging Spectrometer over Pavia city, northern Italy, in 2002, which includes 1096 × 1096 pixels with 1.3-m spatial resolution after removing a black region of 381 × 1096 pixels and 102 bands covering from 400 to 1000 nm after removing 13 bands. We only use a subset of 1096 × 492 to test our method. Except for unknown samples, nine cover types are labeled in the reference map, which includes buildings and city green belt. The pseudocolor (composition of band 60, 30, and 2) and the ground-truth map are shown in Fig. 4.
3) The data set of the Salinas scene was captured by the Reflective Optics System Imaging Spectrometer over Salinas Valley, California, USA, which contains 512 × 217 pixels with 3.7-m spatial resolution and 204 spectral bands covering from 400 to 2500 nm after discarding water absorption bands. Except for unknown samples, there are 16 classes, including vegetables, bare soils, and vineyard fields. The pseudocolor image (composition of band 50, 27, and 17) and the available ground-truth map are shown in Fig. 4.

Since the proposed method takes advantage of extra knowledge from the source hyperspectral image to improve the classification performance, we use two strategies to set up our experiments. The first strategy uses two different hyperspectral images with not exactly the same cover types and takes one data set as the source data set and another as the target data set. We use the data set of the Indian Pines scene as the source data set, and that of the Salinas scene as the target data set. The second strategy is a simulated situation, which divides one hyperspectral data set into two subsets with no overlap in cover types and uses one subset as the source data set and another as the target data set.

Moreover, in order to give an objective and quantitative evaluation, some measurements are used, which are widely used in the works of hyperspectral image classification. Overall accuracy (OA) represents the accuracy of all testing samples; it is the percentage of samples that are classified correctly. Class accuracy (CA) is the percentage of samples that are classified correctly in a certain class. Average accuracy (AA) is the averaged CA over all classes. Kappa coefficient (κ) is a robustness measurement with the degree of agreement.

B. Parameters Analysis

In this section, a series of experiments are conducted to analyze several important parameters of the proposed RL-Net. All involved parameters can be classified into two catalogs, the parameters related to the structure of the two-phase RL-Net, i.e., network parameters, and the parameters related to the arrangement of training samples, i.e., episode parameters. Some other routine parameters of the deep network, which are widely used in the optimization algorithm will be given directly in the related experiments, such as learning rate and iteration number.

Except for the target parameter being analyzed, all parameters are set as the following descriptions. C_1 is the number

[1]All data sets used in this article are publicly available:http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes.

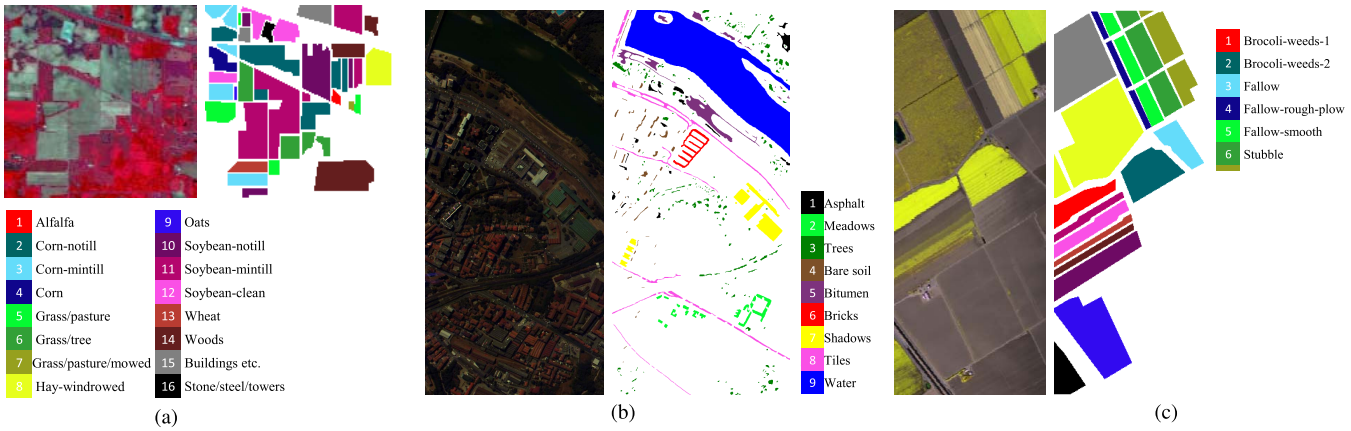


Fig. 4. Three data sets used in this article, including the pseudocolor images and the corresponding ground truths with color index. (a) Indian Pines scene. (b) Pavia center scene. (c) Salinas scene.

of classes in the source data set, C_2 is the number of class in the target data set, C is the number of classes in each episode, and L_1 and L_2 are the number of samples in the example set and the query set, respectively. For the Indian Pines scene, the parameter are $C = C_1 = C_2 = 8, L_1 = 5$, and $L_2 = 15$, which means we use eight classes of the labeled samples as the source data set (classes 11, 12, and 3–8) and the remaining eight classes as the target data set, i.e., eight classes in the training set and eight classes in the support set. All labeled samples of the source data set are used to build training sets; each episode is made up of eight classes with five samples per class in the example set and 15 samples per class in the query set. For the Pavia Center, $C = 4, C_1 = 5, C_2 = 4, L_1 = 5$, and $L_2 = 15$. For the data set of the Salinas scene, $C = C_1 = C_2 = 16, L_1 = 1$, and $L_2 = 15$.

1) Episode Parameters: Episode parameters are about how to build each episode and how many episodes used. Each episode is made up of an example set and a query set, and both sets are built by the samples from different classes; therefore, the episode parameters include the number of episodes, the number of classes C in each episode, the number of samples of each class in the example set L_1, and the number of samples of each class in the query set L_2. Moreover, the number of classes in the training set C_1 and the number of classes in the support set C_2, which are related to the difficulties of the classification task, also influence the episode setting.

We give the analysis of all the episode parameters using the data set of the Indian Pines scene in this section, including class number C of the episode, C_1 of the training set (the source data set), and C_2 of the support set (the target data set) and sample number L_1 of each class in the example set and L_2 of each class in the query set. As given in Table I, we divide the labeled samples in the Indian Pines scene into two disjoint subsets, one subset acts as source data set, which is used to build training sets, and another subset is the target data set, which is used to build the support sets and the testing sets. Theoretically, the class number of the episode C should not be bigger than the class number of the support set C_1.

TABLE I

CLASSIFICATION PERFORMANCE (EVALUATED BY OA, %) UNDER DIFFERENT EPISODE SETTINGS USING THE INDIAN PINES DATA SET

No.	C_1	C_2	C	$L_1 = 1$		$L_1 = 5$	
				$L_2 = 5$	$L_2 = 15$	$L_2 = 5$	$L_2 = 15$
1	8	8	8	74.71	71.53	78.99	83.97
			4	93.99	92.96	94.12	**94.29**
2	8	6	6	82.00	84.35	92.52	92.70
			5	84.78	84.62	92.78	**93.13**
3	10	6	6	90.45	87.31	95.20	94.56
			5	88.48	89.24	**95.54**	95.46
4	6	10	10	62.05	66.98	71.64	71.46
			5	77.73	79.11	84.24	**85.56**

Take Line 3, subline 2 as example, it means there are ten classes in the source data set and six classes in the target data set, we randomly selected five classes to build each episode, and the performance with one and five samples per class in the example set and 5 and 15 samples in the query set are presented.

As given in Table I, we can give the following conclusions.

1) More cover types of the target data set increase the difficulties of the classification task, which makes sense. Look at Lines 1 and 2, when the number of classes increases from 6 to 8, the OA drops more than 5%.
2) More cover types in the source data set give more information. Like Lines 2 and 3, when C_1 changes from 8 to 10, the OA increases from 82% to 90%. Take Lines 3 and 4, when C_1 is bigger than C_2, the accuracy drops drastically, which indicates that the source data set cannot provide enough information for classification of the target data set.
3) For the class number in the episode, C, it cannot be bigger than C_2, and smaller C give better performance than bigger ones, which indicates the more the cover types, the more the difficult.
4) For the number of samples in each episode, if there is only one sample per class in the example set, i.e.,

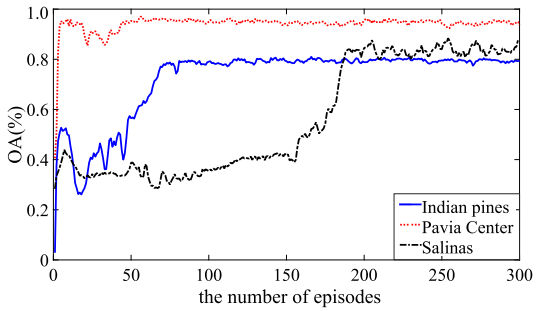


Fig. 5. Classification performance (evaluated by OA, %) of the proposed RL-Net with different numbers of episodes using all three data sets

$L_1 = 1$, and five samples per class in the query set, i.e., $L_2 = 5$ give better performance than $L_2 = 15$, but when $L_1 = 5$, $L_2 = 15$ produces better results than $L_2 = 5$, which indicates that L_2 should be bigger than L_1 but should not too big.

We also present the analysis of the number of episodes using all three data sets. The relationships between the episode number and the final classification results (represented by the averaged OA over multiple trials) are displayed in Fig. 5. From Fig. 5, we can see that a larger number of episodes guarantees better performance until the maximum of OA appears, and then, the performance becomes stable. We can conclude that more episodes can train better model until reaching the model limit, and then, the more episodes give no more information but only the computation cost. The best episode number for the Indian Pines scene is about 100, for the Pavia Center scene is 20, and for the Salina scene is 200. The following experiments will follow the aforementioned episode parameters that can produce the best classification results.

2) Network Parameters: Network parameters include the number of layers in the feature embedding module, the number of layers of the relation learning module, the number of units in each fully connected layer, and the quantity and spatial size of kernels in each convolutional layer. All these parameters work together to decide the architecture of the proposed network. Hereby, we analyze all network parameters together by a different arrangement of network structures. We use the data set of the Indian Pines scene to evaluate the performance under different network settings. As mentioned earlier, we use the labeled samples of eight classes as the source data set and the remaining eight classes as the target data set. Each episode is made up of eight classes with five samples per class in the example set and 15 samples per class in the query set.

The performance of classifications under different network settings, which is represented by averaged OA with the range of fluctuation over multiple trials, is given in Table II. Taking Line 2 for instance, the parameter setting in Line 2, the second column, feature embedding, is 200 (17 × 17)-150, which means there are two layers in this module: the first layer is a convolutional layer with kernel size 17 × 17 and kernel number (also the number of feature maps) 200 and the second layer is a fully connected layer with 150 units. The parameter setting in the third column, relation learning, is 300-150-50-1, which indicates four layers in the relation learning module,

TABLE II

NETWORK PARAMETERS ANALYSIS (EVALUATED BY OA, %) OF THE PROPOSED NETWORKS USING THE INDIAN PINES DATA SET

No.	Feature Embedding	Relation Learning	Measurement
1	200(21×21)-150	300-150-50-1	86.05±4.79
2	200(17×17)-150	300-150-50-1	**86.97±5.36**
3	200(13×13)-150	300-150-50-1	85.72±5.74
4	200(17×17)-200	400-200-100-1	85.40±5.85
5	200(17×17)-100	200-100-30-1	84.59±6.28
6	200(17×17)-150	300-150-80-30-1	85.26±6.53
7	200(17×17)-150	300-150-1	83.14±4.47

and all of them are fully connected layers with different unit numbers. Especially, the last layer computes the relation scores between a sample and five different examples.

From Table II, we can make the following conclusions.

1) For the number of layers, generally speaking, more layers can produce better results until the performance reaches the limitation of the proposed method. Meanwhile, more layers bring more parameters, which increase training difficult and add computational cost.
2) For the number of units in the fully connected layers, the layer with a large unit number indicates that it explores hidden information, and small unit number means it compresses redundant information. The setting of unit number is related to the number of layers, and the shallower network should work with more units to extract more information. Therefore, as in Lines 2 and 6, the deeper one works better with a smaller unit number, but the shallower one works better with larger unit number.
3) About the spatial size of kernels in the convolutional layers, it related to the spatial resolution of the hyperspectral image; the higher the spatial resolution, the larger the spatial size of the kernels. As shown in Lines 1–3, the performance of the Indian Pines scene gets better when the spatial size changes from 13 to 17, but when it bigger than 17, the computational cost will increase. Therefore, after the tradeoff between computation cost and classification performance, the appropriate parameters for the data set of the Indian Pines scene are two layers in the feature learning module: 200 kernel of size 17 × 17 in the first layer, 150 units in the second layer, and two layers in relation learning module.

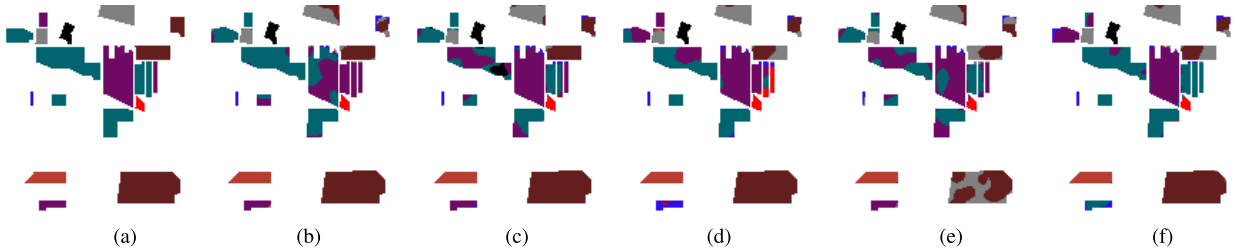
C. Performance Evaluation

To evaluate the performance of the proposed RL-Net, we compare it with four other methods: one directly supervised method based on deep network (Sup-Net), one semisupervised method based on deep network (Semi-Net), one fine-tuned deep learning-based method (FT-Net), and one more state-of-the-art deep learning-based classification method, i.e., band-specific spectral–spatial classification network (BASS-Net) [7]. For a fair comparison, we try to set all methods under the same settings; in other words, all the

TABLE III

EXPERIMENTAL DETAILS OF ALL THREE DATA SETS, INCLUDING THE SOURCE DATA SET, THE NETWORK PARAMETERS, AND THE OUTPUT SIZE

Dataset	Auxiliary Dataset	Feature Embedding				Relation Learning				
		Input	Conv.	Fc.1	Output	Input	Fc.2	Fc.3	Fc.4	Output
Indian Pines(9-16)	Indian Pines(1-8)	17×17×200	17×17,1	200×150,150	1×150	1×300	300×150,150	150×50,50	50×1,1	1
Pavia Center(6-9)	Pavia Center(1-5)	17×17×102	17×17,1	102×80,80	1×80	1×160	160×80,80	80×30,30	30×1,1	1
Salinas(1-16)	Indian Pines(1-16)	17×17×200	17×17,1	200×150,150	1×150	1×300	300×150,150	150×50,50	50×1,1	1



(a) (b) (c) (d) (e) (f)

Fig. 6. Classification maps of different methods for the Indian Pines data set with five training samples for each class. (a) Ground truth. (b) Sup-Net. (c) Semi-Net. (d) FT-Net. (e) BASS-Net. (f) RL-Net.

methods use the same spectral–spatial features except BASS-Net, which proposes a special designed deep feature extraction method for hyperspectral image classification. The directly supervised method tries to explore labeled samples of the target data set exhaustively, which uses the feature embedding module in the proposed network with softmax as a classifier. This method indicates the classification result with the features from the insufficient labeled training samples only, and it serves as a comparison baseline. The semisupervised method takes advantage of both labeled samples and unlabeled samples, and it also uses the feature embedding module in the proposed network with softmax as a classifier and takes the advantage of unlabeled samples with multi-decision. This method gives a clue of the classification result with the target data set only. FT-Net trains the deep network on the source data set and then fine-tunes it on the target data set; it also uses the feature embedding module in the proposed network with softmax as a classifier and fine-tunes the supervised method on the target data set. This method is a naive solution for using the source data. Moreover, one more method, extended morphological attribute profile (EMAP) with spectral features (EMAP-Spe), is also adopted to represent shallow feature extraction methods. Furthermore, in order to prevent the bias caused by different training samples, all results are averaged over ten runs with randomly selected samples, and the one trial close to the averaged accuracy is selected to demonstrate.

In the beginning, we present a detailed experimental setting and parameters, as given in Table III. Taking the Indian Pines scene as an example, we use the samples of the first eight classes as the source data set and take the remaining samples to classify. All the network parameters can be seen in Table III. The episode parameters are $\{C = C_1 = C_2 = 8, L_1 = 5, L_2 = 15$, numEpisode=300$\}$, and the optimize parameter learning rate is 0.001. The classification maps of all methods along with the ground-truth map are shown in Fig. 6. For the data set of the Pavia Center scene, the episode parameters are $\{C = 4, C_1 = 5, C_2 = 4, L_1 = 5, L_2 = 15$, numEpisode=300$\}$, and the optimize parameter learning rate is 0.001. The classification maps of all methods along with the ground-truth map are shown in Fig. 7. For the data set of Salinas scene, the episode parameters are $\{C = C_1 = C_2 = 16, L_1 = 1, L_2 = 15$, numEpisode $= 300\}$, and the optimize parameter learning rate is 0.001. The classification maps of the proposed method along with all the comparisons are shown in Fig. 8. Moreover, the averaged OA, AA, and Kappa coefficient along with corresponding derivations over ten trials of all methods for all three data sets are given in Table IV.[2]

From all Figs. 5–7, we can see that the proposed method can produce the best classification results with small training sets that are consisted of five samples per class, 40 samples in total. More information gives higher accuracy, Semi-Net uses the information of unlabeled samples for information compensation, and FT-Net takes the information from the source data set by further training the pertained deep networks; therefore, both of them give better performance than Sup-Net and BASS-Net. Moreover, FT-Net is not an effective way to take advantage of the source data set since it cannot give better performance than Semi-Net that takes unlabeled samples for information compensation. Furthermore, the proposed RL-Net gives better performance than FT-Net, which indicates that it is a better choice of using source data sets.

Moreover, Table IV gives the performance evaluated by all three measurements of different methods. From Table IV, we can see that the proposed RL-Net gives the highest measurements with the smallest fluctuations, such as for the data set of Pavia Center scene, all five methods are able to give the satisfied performance, three of them give OA higher than 96%, but the proposed RL-Net can produce results with the least fluctuations. Therefore, we can conclude that the

[2]More experiments can be found on the following website: https://www.researchgate.net/profile/Xiaorui_Ma3/research.

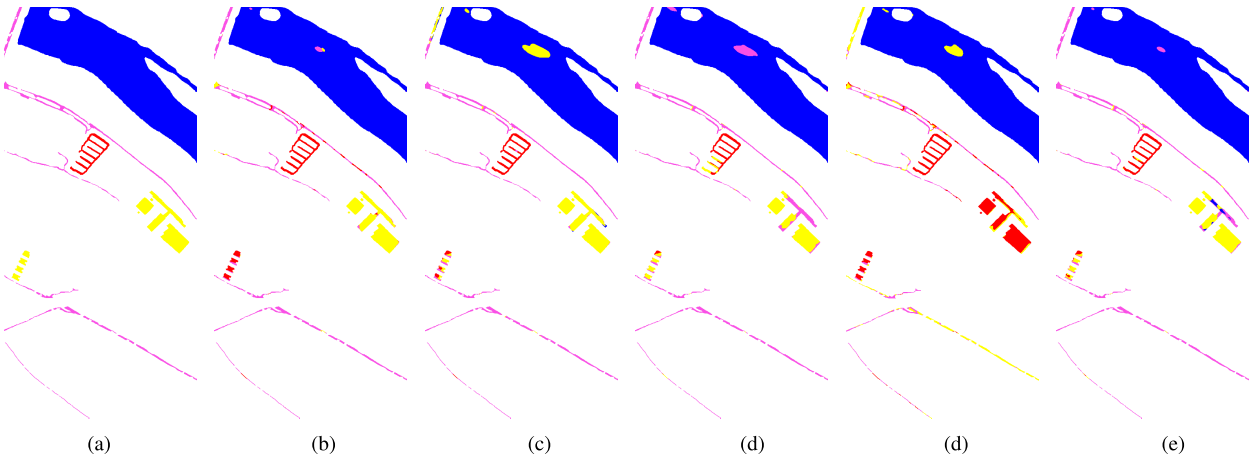


Fig. 7. Classification maps of different methods for the Pavia Center data set with five training samples for each class. (a) Ground truth. (b) Sup-Net. (c) Semi-Net. (d) FT-Net. (d) BASS-Net. (e) RL-Net.

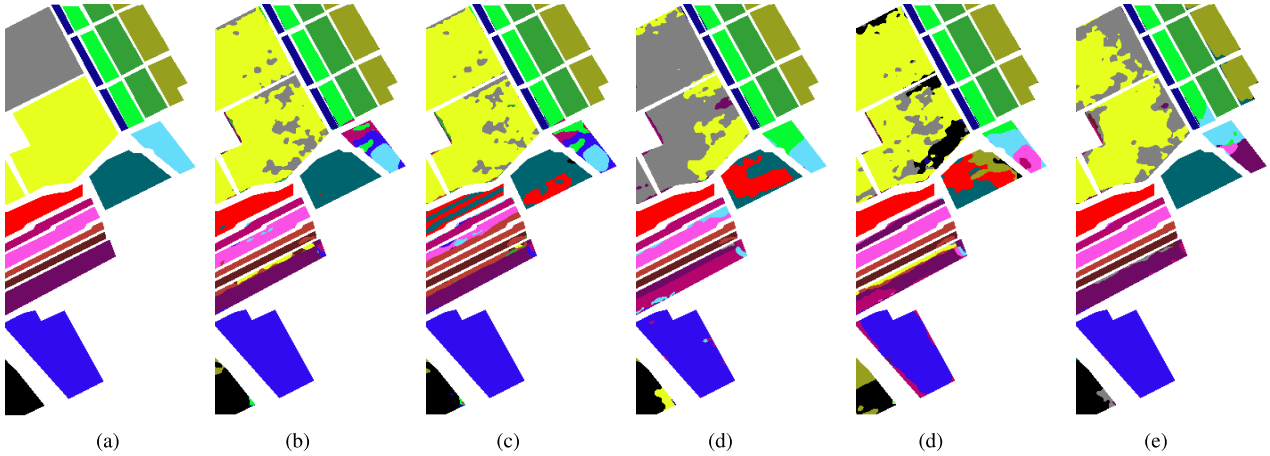


Fig. 8. Classification maps of different methods for the Salinas data set with one training samples for each class. (a) Ground truth. (b) Sup-Net. (c) Semi-Net. (d) FT-Net. (d) BASS-Net. (e) RL-Net.

TABLE IV

CLASSIFICATION PERFORMANCE REPRESENTED BY AVERAGED OA, AA, AND κ OF DIFFERENT METHODS FOR ALL THREE HYPERSPECTRAL IMAGES

Dataset	Measurement	EMAP-Spe	Sup-Net	Semi-Net	FT-Net	BASS-Net	RL-Net
Indian Pines	OA(%)	70.42±7.19	79.13±10.83	83.34±6.02	80.67±7.23	64.22±7.59	**83.97±5.36**
	AA(%)	60.56±6.87	86.42±9.25	90.28±5.69	75.15±6.58	62.13±6.67	**91.17±2.14**
	$\kappa(\times 100)$	62.42±9.93	73.01±13.35	78.37±7.67	75.04±7.06	63.89±8.25	**79.41±4.61**
Pavia Center	OA(%)	90.71±3.85	96.05±3.85	**96.82±4.09**	94.81±4.21	83.04±4.97	**96.99±3.82**
	AA(%)	90.40±5.15	92.68±5.15	93.54±3.45	85.70±4.69	82.19±5.16	**96.50±3.01**
	$\kappa(\times 100)$	88.46±7.54	89.43±9.45	91.30±10.09	87.01±8.33	81.58±8.32	**92.34±6.69**
Salinas	OA(%)	66.70±9.45	72.70±8.14	77.45±7.14	73.78±7.62	69.42±10.37	**79.27±5.28**
	AA(%)	69.56±7.78	78.96±4.23	84.81±3.45	80.51±3.89	70.74±3.28	**85.83±2.20**
	$\kappa(\times 100)$	65.56±9.80	69.83±8.04	75.00±7.95	70.96±7.10	66.38±7.63	**77.01±5.33**

proposed network can produce more robust results with small training sets than the other compared methods.

Furthermore, we analyze the effect of the training set. We use the Indian Pines scene to build three support sets with 5, 10, and 15 samples per class, respectively, and then train the RL-Net using the support set only, or using both the support set and the training set, and compare them with the Sup-Net. The classification accuracies evaluated by OA are given in Table V. The classification result using the support set only is denoted as RL-Net(SS) in the third column. From V, we can see that the classification accuracies of RL-Net(SS) are close to those of the Sup-Net but slightly better with smaller support sets, which shows the advantage with less training samples. Moreover, RL-Net that takes advantage of the source

TABLE V

CLASSIFICATION PERFORMANCE (EVALUATED BY OA, %) OF THE PROPOSED RL-NET WITHOUT AND WITH SOURCE DATA SET USING THE INDIAN PINES SCENE

# Sample (per class)	Sup-Net	RL-Net(SS)	RL-Net
5	79.42±10.83	81.55±5.91	83.97±5.36
10	85.67±8.50	86.36±5.52	88.40±4.87
15	90.02±7.99	89.61±4.50	91.18±4.27

TABLE VI

COMPUTATIONAL COMPLEXITY (THE NUMBER OF FLOATING-POINT OPERATIONS) OF DIFFERENT DEEP NETWORKS FOR THE SALINAS SCENE

Method	Complexity	Parameter details
BASS-Net	480,400	5 convolutional layers with multiple kernels, 2-3 fully-connected layers
PPF-Net	450,620	8 convolutional layers with multiple kernels, 2 fully-connected layers
RL-Net	82,839	1 convolutional layer with only one kernel, 4 fully-connected layers

data set gives better performance than both Sup-Net and RL-Net(SS), which indicates the advantage brought by the source data set.

Finally, for the Salinas scene, the proposed RL-Net takes about 328 s to train the network on the source data set and spends about 12 s to test all samples. Except for BASS-Net, all the contrastive methods use the same spectral–spatial features, they take exactly the same time for feature learning. For deep learning, the most essential factor of the time cost is network architecture. Time complexity, which is also known as the number of floating-point operations, is an effective measurement to evaluate the time cost of deep network architectures. Therefore, we give time complexities of several prevalent deep networks in hyperspectral image classification in Table VI, including BASS-Net, PPF-Net [33], and the proposed RL-Net. Since most layers of the RL-Net are fully connected layers, which contain fewer parameters than the convolutional layers and are easy to train, the proposed network is relatively faster than other deep networks.

VI. CONCLUSION

This article proposed a transfer-based classification method that is a two-phase RL-Net with episode-based training strategy. To take advantage of the source data sets, the proposed method focused on what information to transfer, how to learn the transferable information, and how to transfer the learned information. The experiments and related analysis prove that the proposed method can take advantage of the knowledge from other data sets that may be covered by different land-cover types and achieve competitive classification performance with small training sets, even with only a few labeled samples.

In the future, we will continue the research on the classification of a few labeled samples and improve the proposed method in the following aspects. First, most of the source data sets we used are from the same sensors, and we will develop a new method to deal with more challenging data sets, such as those with different band numbers. Second, in this article, there are labeled samples for each of the class in the target data set, and we want the achieved zero sample classification to recognize new cover type without seen it before. Finally, in order to make the proposed method more robust to the training sets, we are going to develop a sample elimination strategy that will remove the unreliable samples from the support set.

ACKNOWLEDGMENT

The authors would like to thank all professionals for kindly providing hyperspectral images, corresponding reference information, and the codes of comparing methods.

REFERENCES

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

[2] G. Camps-Valls, D. Tuia, L. Bruzzone, and J. A. Benediktsson, "Advances in hyperspectral image classification: Earth monitoring with statistical learning methods," *IEEE Signal Process. Mag.*, vol. 31, no. 1, pp. 45–54, Jan. 2014.

[3] 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.

[4] Y. Gu, J. Chanussot, X. Jia, and J. A. Benediktsson, "Multiple kernel learning for hyperspectral image classification: A review," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 11, pp. 6547–6565, Nov. 2017.

[5] X. Kang, X. Xiang, S. Li, and J. A. Benediktsson, "PCA-based edge-preserving features for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 12, pp. 7140–7151, Dec. 2017.

[6] L. Gao, B. Zhao, X. Jia, W. Liao, and B. Zhang, "Optimized kernel minimum noise fraction transformation for hyperspectral image classification," *Remote Sens.*, vol. 9, no. 6, pp. 548–567, Jun. 2017.

[7] A. Santara *et al.*, "BASS net: Band-adaptive spectral-spatial feature learning neural network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 9, pp. 5293–5301, Sep. 2017.

[8] S. Mei, J. Ji, J. Hou, X. Li, and Q. Du, "Learning sensor-specific spatial-spectral features of hyperspectral images via convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 8, pp. 4520–4533, Aug. 2017.

[9] C. Zhang, J. Wang, Y. Zhang, and Y. Liu, "Small-sample classification of hyperspectral data in a graph-based semi-supervision framwork," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2017, pp. 3194–3197.

[10] Q. Liu, Y. Sun, R. Hang, and H. Song, "Spatial–spectral locality-constrained low-rank representation with semi-supervised hypergraph learning for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 9, pp. 4171–4182, Sep. 2017.

[11] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 594–611, Apr. 2006.

[12] 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.

[13] 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.

[14] G. Taşkin, H. Kaya, and L. Bruzzone, "Feature selection based on high dimensional model representation for hyperspectral images," *IEEE Trans. Image Process.*, vol. 26, no. 6, pp. 2918–2928, Jun. 2017.

[15] W. Gao and Y. Peng, "Ideal kernel-based multiple kernel learning for spectral-spatial classification of hyperspectral image," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 7, pp. 1051–1055, Jul. 2017.

[16] Y. Chen, N. M. Nasrabadi, and T. D. Tran, "Hyperspectral image classification via kernel sparse representation," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 1, pp. 217–231, Jan. 2013.

[17] S. Jia, J. Hu, Y. Xie, L. Shen, X. Jia, and Q. Li, "Gabor cube selection based multitask joint sparse representation for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 6, pp. 3174–3187, Jun. 2016.

[18] P. Chen, J. D. B. Nelson, and J.-Y. Tourneret, "Toward a sparse Bayesian Markov random field approach to hyperspectral unmixing and classification," *IEEE Trans. Image Process.*, vol. 26, no. 1, pp. 426–438, Jan. 2017.

[19] X. Zhang, Z. Gao, L. Jiao, and H. Zhou, "Multifeature hyperspectral image classification with local and nonlocal spatial information via Markov random field in semantic space," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 3, pp. 1409–1424, Mar. 2018.

[20] P. Ghamisi, M. D. Mura, and J. A. Benediktsson, "A survey on spectral–spatial classification techniques based on attribute profiles," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 5, pp. 2335–2353, May 2015.

[21] J. Li, X. Zhao, Y. Li, Q. Du, B. Xi, and J. Hu, "Classification of hyperspectral imagery using a new fully convolutional neural network," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 2, pp. 292–296, Feb. 2018.

[22] L. Shu, K. McIsaac, and G. R. Osinski, "Hyperspectral image classification with stacking spectral patches and convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 10, pp. 5975–5984, Oct. 2018.

[23] 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.

[24] M. He, B. Li, and H. Chen, "Multi-scale 3D deep convolutional neural network for hyperspectral image classification," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 3904–3908.

[25] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[26] X. X. Zhu *et al.*, "Deep learning in remote sensing: A comprehensive review and list of resources," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 4, pp. 8–36, Dec. 2017.

[27] 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.

[28] P. Zhong, Z. Gong, S. Li, and C.-B. Schönlieb, "Learning to diversify deep belief networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 6, pp. 3516–3530, Jun. 2017.

[29] Z. Gong, P. Zhong, Y. Yu, W. Hu, and S. Li, "A CNN with multiscale convolution and diversified metric for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 6, pp. 3599–3618, Jun. 2019. doi: 10.1109/TGRS.2018.2886022.

[30] S. Hao, W. Wang, Y. Ye, T. Nie, and L. Bruzzone, "Two-stream deep architecture for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 4, pp. 2349–2361, Apr. 2018.

[31] L. Jiao, M. Liang, H. Chen, S. Yang, H. Liu, and X. Cao, "Deep fully convolutional network-based spatial distribution prediction for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 10, pp. 5585–5599, Oct. 2017.

[32] Z. Zhong, J. Li, Z. Luo, and M. Chapman, "Spectral–spatial residual network for hyperspectral image classification: A 3-D deep learning framework," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 2, pp. 847–858, Feb. 2018.

[33] 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.

[34] H. Lee and H. Kwon, "Going deeper with contextual CNN for hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 26, no. 10, pp. 4843–4855, Oct. 2017.

[35] S. Zhou, Z. Xue, and P. Du, "Semisupervised stacked autoencoder with cotraining for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 6, pp. 3813–3826, Jun. 2019. doi: 10.1109/TGRS.2018.2888485.

[36] Z. Chen, Y. Fu, Y. Zhang, Y.-G. Jiang, X. Xue, and L. Sigal, "Multi-level semantic feature augmentation for one-shot learning," Aug. 2018, *arXiv:1804.05298*. [Online]. Available: https://arxiv.org/abs/1804.05298

[37] D. J. Rezende, S. Mohamed, I. Danihelka, K. Gregor, and D. Wierstra, "One-shot generalization in deep generative models," in *Proc. ICML*, New York, NY, USA, Jun. 2016, pp. 1521–1529.

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

[39] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," Nov. 2017, *arXiv:1711.06025*. [Online]. Available: https://arxiv.org/abs/1711.06025

[40] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2017, pp. 4080–4090.

Xiaorui Ma (M'17) received the B.S. degree in applied mathematics from Lanzhou University, Lanzhou, China, in 2008, and the Ph.D. degree in communication and information system from the Dalian University of Technology, Dalian, China, in 2017.

She is currently a Lecturer with the Dalian University of Technology. Her research interests include processing and analysis of remote sensing images, especially hyperspectral image classification and synthetic aperture radar image classification.

Sheng Ji received the B.S. degree in electronic and information engineering from the Dalian University of Technology, Dalian, China, in 2017, where he is currently pursuing the M.E. degree with the School of Information and Communication Engineering.

His research interests include hyperspectral image classification and target detection, remote sensing image analysis and interpretation, and machine learning.

Jie Wang (M'12–SM'18) received the B.S. degree in electronic engineering from the Dalian University of Technology, Dalian, China, in 2003, the M.S. degree in electronic engineering from Beihang University, Beijing, China, in 2006, and the Ph.D. degree in electronic engineering from the Dalian University of Technology, in 2011.

He is currently a Full Professor with Dalian Maritime University, Dalian. His research interests include wireless localization and tracking, radio tomography, wireless sensing, cognitive radio networks, and machine learning.

Jie Geng received the B.S. and Ph.D. degrees in electronic and information engineering from the Dalian University of Technology, Dalian, China, in 2013 and 2018, respectively.

He is currently an Assistant Professor with the School of Electronics and Information, Northwestern Polytechnical University, Xi'an, China. His research interests include processing and analysis of remote sensing images, especially hyperspectral image classification and synthetic aperture radar image classification.

Hongyu Wang received the B.S. degree in electronic engineering from the Jilin University of Technology, Changchun, China, in 1990, the M.S. degree in electronic engineering from the Graduate School of Chinese Academy of Sciences, Beijing, China, in 1993, and the Ph.D. degree in precision instrument and optoelectronics engineering from Tianjin University, Tianjin, China, in 1997.

He is currently a Professor with the Dalian University of Technology, Dalian, China. His research interests include image processing, image analysis, and remote sensing image classification.