remote sensing

MDPI

Article

Semi-Supervised Deep Learning Classification for Hyperspectral Image Based on Dual-Strategy Sample Selection

Bei Fang [1], Ying Li [1,*], Haokui Zhang [1] and Jonathan Cheung-Wai Chan [2]

[1] School of Computer Science, Northwestern Polytechnical University, Xi'an 710129, China; kkbei@mail.nwpu.edu.cn (B.F.); hkzhang1991@mail.nwpu.edu.cn (H.Z.)

[2] Department of Electronics and Informatics, Vrije Universiteit Brussel, Brussel 1050, Belgium; jcheungw@etrovub.be

* Correspondence: lybyp@nwpu.edu.cn; Tel.: +86-138-9143-3893

Received: 24 January 2018; Accepted: 3 April 2018; Published: 8 April 2018

Abstract: This paper studies the classification problem of hyperspectral image (HSI). Inspired by the great success of deep neural networks in Artificial Intelligence (AI), researchers have proposed different deep learning based algorithms to improve the performance of hyperspectral classification. However, deep learning based algorithms always require a large-scale annotated dataset to provide sufficient training. To address this problem, we propose a semi-supervised deep learning framework based on the residual networks (ResNets), which use very limited labeled data supplemented by abundant unlabeled data. The core of our framework is a novel dual-strategy sample selection co-training algorithm, which can successfully guide ResNets to learn from the unlabeled data by making full use of the complementary cues of the spectral and spatial features in HSI classification. Experiments on the benchmark HSI dataset and real HSI dataset demonstrate that, with a small number of training data, our approach achieves competitive performance with maximum improvement of 41% (compare with traditional convolutional neural network (CNN) with 5 initial training samples per class on Indian Pines dataset) for HSI classification as compared with the results from those state-of-the-art supervised and semi-supervised methods.

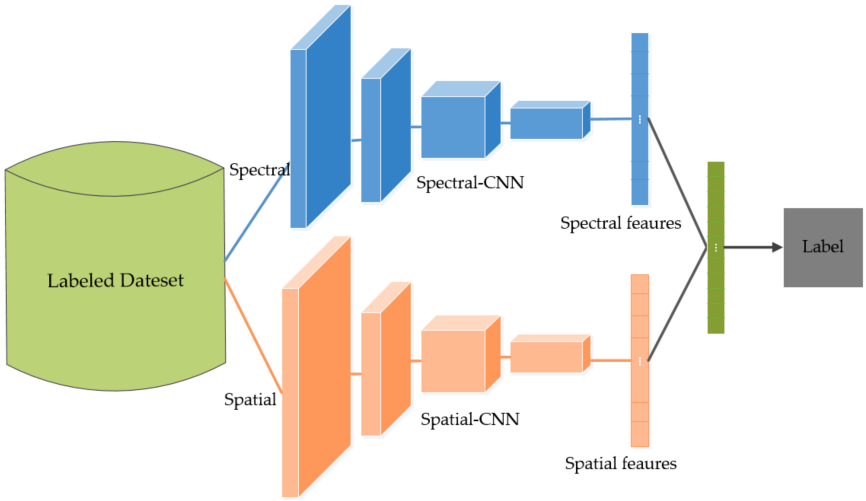
Keywords: hyperspectral image classification; deep learning; residual networks; co-training; sample selection

1. Introduction

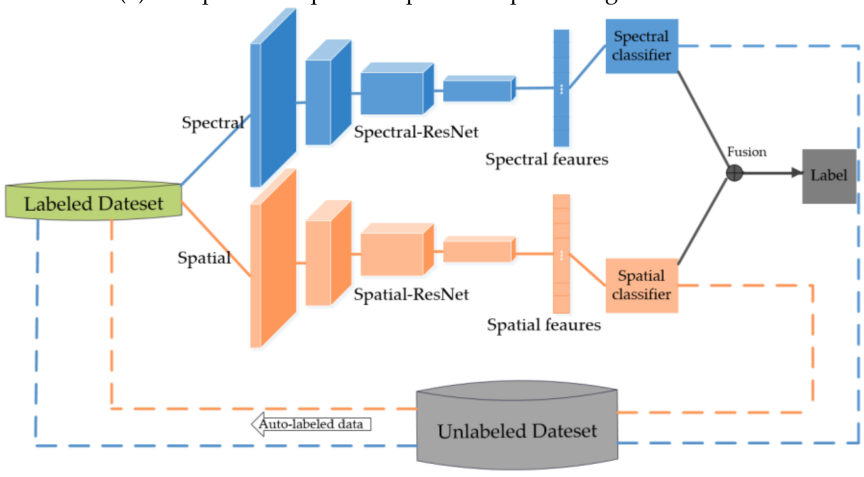
Hyperspectral image (HSI) collected by imaging spectrometers captured rich spectral and spatial information simultaneously [1]. For this reason, hyperspectral data are used on a wide range of applications such as environmental sciences [2], agriculture [3], and mineral exploitation [4]. HSI classification is one of the most important topics in remote sensing. Specifically, combining the rich spectral information and spatial information as complementary cues represents an opportunity to dramatically improve the performance of HSI classification.

Most recently, deep learning has emerged as the state-of-the-art machine learning technique with a great potential for HSI classification. Instead of depending on manually-engineered shallow features, deep learning techniques automatically learn hierarchical features (from low-level to high-level) from raw input data [5,6]. Inspired by the great success of deep learning for image classification, remarkable efforts have been invested for spectral-spatial HSI classification by deep learning techniques in the last few years [7–12]. These deep learning algorithms falls into two broad categories. The first category includes features learning and classification steps. For example, Chen et al. [7] applied deep feature learning by including stacked autoencoder (SAE) and deep belief network (DBN) [8] for spectral–spatial

feature extraction. Then, multiclass logistic regression is used for classification. The main drawback of these approaches is that it has not built a unified solution for feature learning and classification. The second category tries to concatenate the spectral and spatial features as well as classification, as shown in Figure 1a. For example, Yue et al. and Chen et al. [9–11] have utilized convolutional neural networks (CNNs) to jointly learn the features and classifiers in an end-to-end fashion. Specifically, as described by Yue et al. [9], a 2D-CNN with two convolutional layers and three subsampling layer is used to extract deep features from HSI with a spatial size of 42 × 42. Chen et al. [10] used deep CNN with three convolutional layers and one fully connected layer to extract deep features from HSI with a spatial size of 27 × 27. Li et al. [11] proposed a 3D-CNN framework to extract deep spectral-spatial combined features with two 3D convolution layers, one fully connected layer and one classification layer; the spatial size is empirically set to 5 × 5.



(**a**) A supervised spectral–spatial deep learning framework.



(**b**) Our semi-supervised spectral–spatial deep learning framework.

Figure 1. The structures of (**a**) supervised and (**b**) semi-supervised spectral–spatial deep learning for HSI classification.

It should be noted that deep neural networks always require a large number of datasets for supervised training, e.g., ImageNet with millions of annotated images [12]. However, labeling a large archive of hyperspectral data for classification task is very expensive and time consuming. To address this challenge, Yang et al. [13] proposed a deep CNN with two-branch architecture to extract the joint spectral-spatial features from HSI which is reportedly beneficial when the number of training samples

is limited. Ma et al. [14] developed a spatial updated deep autoencoder, in order to deal with the small training set using deep features, a collaborative representation-based classification is applied.

Although previous works use the supervised method with a small training samples, they do not benefit from the massive unlabeled data to promote the classification performance. Thus, it is necessary to develop a new effective training framework for deep learning to benefit from the massive unlabeled data which is already available. To make full use of massive unlabeled samples, some semi-supervised methods exist in the literature [15,16], we are particularly interested in the co-training algorithm, which is an important paradigm of semi-supervised methods [17–19]. Blum and Mitchell [20] have given theoretical proofs to guarantee the success of co-training in utilizing the unlabeled samples. At each iteration of the co-training process, two learners are trained independently from two views and are required to label some unlabeled examples for each other to augment the training set [19]. The co-training strategy has already been considered to solve HSI classification on the conditions that: (1) Each example contains two views, either of which is able to depict the example well; and (2) the two views should not be highly correlated. Hyperspectral data matches the two conditions well by providing the spectral features and spatial features that are conditionally independent [21], and co-training can exploit the limited labeled data with the massive unlabeled data to improve the performance. Romaszewski et al. [18] used co-training approach with the P-N learning scheme, which P-expert assumes the same class labels for spatially close pixels and the N-expert detects pixels with similar spectra. P-expert and N-expert take advantage of the spatial structure and the spectral structure respectively. Tan et al. [17] used to tri-training to exploit spectral and spatial information for hyperspectral data classification is presented based on an active learning and a multi-scale homogeneity. In order to make accurate predictions of the unknown labels of a sparsely labeled image, Appice et al. [21] applied a transductive learning approach with a co-training schema.

To the best of our knowledge, in previous co-training algorithms, unlabeled samples selected for augmenting the training set are the ones with the highest confidence from a single view (spectral or spatial) sample selection criteria, such as the spatial neighbors sample selection with active learning [17], the spectral neighbors sample selection with Euclidean spectral distance [18], spatial information extracted with segmentation algorithm [22], the spatial example selection with the diversity class criterion [19]. However, when only few training data are available, as is the case with spectral-spatial HSI classification, this strategy is not appropriate because the training samples are too few to describe adequately the distribution of the data from either the spectral view or spatial view. To address this issue, we propose a new sample selection scheme for co-training process based on spectral features and spatial features views.

Another obstacle of deep neural networks training is that when deeper networks are able to start converging, a degradation problem has been exposed [23]. Fortunately, degradation problem with the increase of convolutional layers can be solved by adding shortcut connections between every other layer and propagating the value of features by the latest residual networks learning framework (ResNet) as proposed by He et al. [23]. Zhong et al. [24] have used ResNet for supervised HSI classification, but the method has not exploited the unlabeled data. In this paper, therefore, the goal is to develop a semi-supervised deep learning classification framework based on co-training. The framework is illustrated in Figure 1b.

The pipeline of the framework can be summarized as follows. First, the spectral-ResNet and spatial-ResNet models are trained on the given labeled data of the respective views. Then at each iteration of co-training, two models are applied to predict the unlabeled sets, and the most confident labeled samples are used to augment the training set of the other model (view). The iterative process is repeated until some stopping criterion has been reached. Finally, the classification result of the spectral features is fused with that of the spatial features to obtain the label of the test data.

The main contribution of this paper can be summarized as three aspects. Firstly, ResNets are used to extract the spectral features and spatial features for HSI classification. The identity mapping of the ResNet can alleviate the degradation of the classification performance of deep learning models caused

by increased depth. Secondly, in order to select a set of informative and high confident samples from the unlabeled datasets to update the next round training of the deep learning models effectively, a new sample selection scheme for co-training process based on spectral features and spatial features views is proposed. Finally, we verify the advantages of our method by testing it on several benchmark HSI datasets and a selected Hyperion dataset.

The remainder of this paper is organized as follows. In Section 2, the general framework is presented, and a sample selection scheme is presented in detail. We present the experimental results and discuss about the experimental results in Sections 3 and 4, respectively. Finally, in Section 5, the paper is summarized, and the future works are suggested.

2. Method

2.1. Overview

The proposed framework aims at learning a powerful semi-supervised deep learning framework for HSI classification based on limited labeled data and the wealth of unlabeled data. To be specific, we have a small labeled pool L, and we have a large-scale unlabeled hyperspectral dataset U. The proposed framework is shown in Figure 2, where a spectral-spatial co-training algorithm based on deep learning is introduced to learn from the unlabeled data. Now, three important phases of the framework will be introduced.

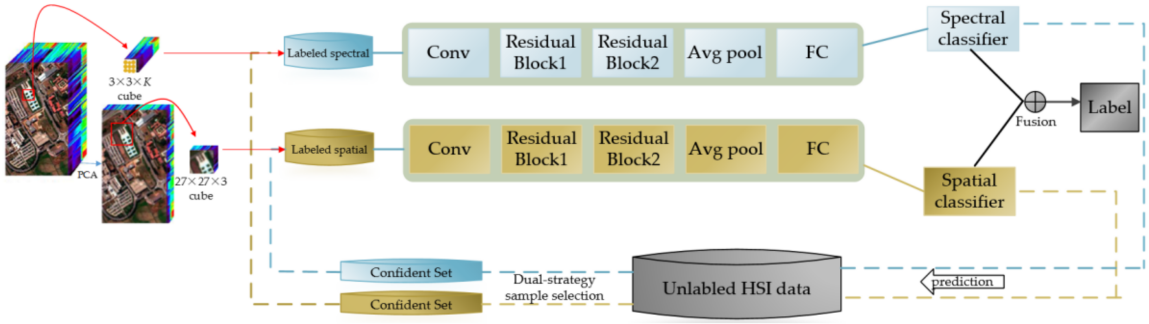


Figure 2. Overview of semi-supervised deep learning framework for hyperspectral image (HSI) classification. The training of the framework mainly two iterative steps: (1) Training the spectral- and spatial- models over the respective data based on the labeled pool (indicated as solid lines); (2) applying each model to predict the unlabeled HSI data and use respective sample selection strategy to select the most confident samples for the other (indicated as dashed lines. See details in the text). After all iterations of co-training are completed, the classification results of the test dataset which obtained through two training networks were fused, and then the label of the test dataset was obtained (indicated as solid black lines).

Network Architectures. Building a suitable network architecture is the first prerequisite of the whole system. This paper adopts the architecture of ResNet (Section 2.2) to extract both the spectral and spatial features. Residual Networks can be regarded as an extension of CNN's with skipped connections that facilitate the propagation of gradients and perform robustly with very deep architecture [24]. However, the extremely limited number of training samples for such deep learning models is difficult. To address this problem, we utilized the regularization method Batch Normalization (BN) [25] to prevent the learning process from overfitting.

Training Process. Training of the semi-supervised deep learning framework mainly involves two iterative steps: Training each ResNet model and updating the labeled pool as illustrated in Figure 2. More specifically, we denote the state of the system at the t-th iteration of co-training as L_t and U_t are denoted as labeled training samples and unlabeled samples respectively. To effectively select informative and confident samples from U_t to update the next round training set of the deep learning

models, a dual-strategy sample selection co-training algorithm based on spectral and spatial features is introduced in Section 2.3.

The goal of the proposed dual-strategy sample selection method is that labeling and selecting the unlabeled samples for each model are based on both spectral and spatial features. To this end, for spectral view of the co-training, we propose a new similarity metric based on deep spectral feature learning, it is a measurement to define the relationship between two samples. In particular, we extract the hierarchical features from a deep network on all available samples (labeled samples and unlabeled samples), and then the distance between labeled samples and unlabeled samples is given by the Euclidean distance. Using this method, we can select the most confident spectral samples with high similarity for each labeled sample to be included in the new training set on the condition that the spectral-ResNet agree on the labeling of these unlabeled samples. For spatial view of the co-training, we use a spatial neighborhood information extraction strategy to select the most confident spatial neighbors as the new training set based on the condition that spatial-ResNet agree on the labeling of these unlabeled samples. Such dual-strategy is believed to select the most useful and informative samples to update the training set for the next round of training of the deep learning models.

Testing Process. The iterative process is repeated until some stopping criterion has been reached. After the fully connected layers, the output of the fully connected layers represent the spectral features and spatial features, which are followed by a softmax regression classifier (defined in this work as spectral classifier and spatial classifier) to predict the probability distribution of each class. Finally, the prediction probability vector of the test dataset from the two channels are summed to get the final classification result, and then the label of the test dataset was obtained, as shown in Figure 2 indicated as solid black lines.

2.2. Networks Architectures Based on Spectral and Spatial Features

ResNet is constructed via stacking residual blocks, and it skips blocks of convolutional layers by using shortcut connections to form residual blocks. By using shortcut connections, residual networks perform residual mapping fitted by stacked nonlinear layers, which is easier to be optimized than the original mapping [23]. These stacked residual blocks significantly improve training efficiency and largely resolve the degradation problem by employing batch normalization BN [25]. Inspired by the latest residual networks learning framework proposed by He et al. [23], the architecture of our networks for each model, as shown in Figure 3, contains one convolutional layer and two "bottleneck" building blocks, and each building block has one shortcut connection. Each residual block can be expressed in a general form as follows:

$$\begin{aligned} x_{l+1} &= h(x_l) + F(x_l, W_l), \\ x_{l+1} &= f(y_l), \end{aligned} \tag{1}$$

where x_l and x_{l+1} are input and output of the l-th block, respectively. W_l denotes the parameters of the residual structure. $F(x)$ is a residual mapping function and $h(x_l) = x_l$ is an identity mapping function, and f is a Rectified Linear Units (ReLU) [26] function. For each residual mapping function F, the "bottleneck" building block has a stack of 3 layers. Take spatial-ResNet in the right of Figure 3 for example, the three layers are $[1 \times 1, 20]$, $[3 \times 3, 20]$, and $[1 \times 1, 80]$ convolutions, where 1×1 convolutions layers are responsible for reducing and then increasing (restoring) dimensions, leaving the 3×3 convolutions layer a bottleneck with smaller input/output dimensions [23]. To regularize and speed up the training process, we adopt batch normalization BN [25] right after each convolution and before activation. BN standardizes the mean and variance of hidden layers for each mini-batch, is defined as follows.

$$\hat{x}^{(i)} = \frac{x^{(i)} - E(x^{(i)})}{\mathrm{VAR}(x^{(i)})}, \tag{2}$$

where $\hat{x}^{(i)}$ is the i-th dimension of feature batch x, $E(\cdot)$ represents the expected value and $\mathrm{VAR}(\cdot)$ is the variance of the features. In order to prevent overfitting, 'dropout' (random omission of part of the feature during each training case) is employed in our method after the average pooling in each branch, the dropout rate is set to 0.5.

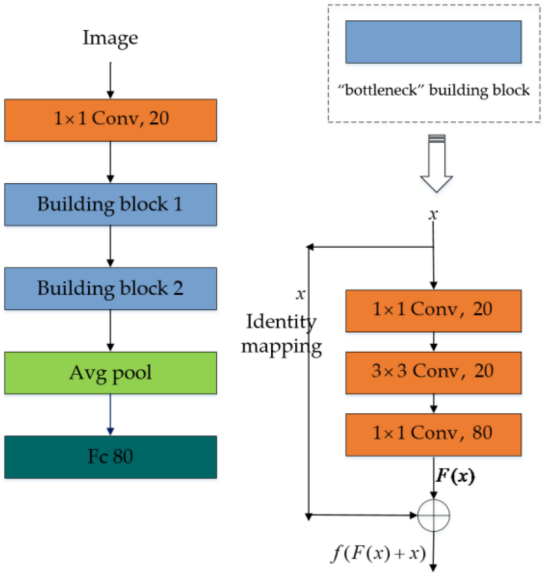


Figure 3. A residual network for spatial-ResNet model, which contains two "bottleneck" building blocks, and each building block has one shortcut connection. The number on each building block is the number of output feature map. *F(x)* is the residual mapping and *x* is the identity mapping, for each residual function, we use a stack of 3 layers. The original mapping is represented as *F(x)* + *x*.

In the spectral-ResNet model, for one pixel of HSI which is to be dealt with, a $3 \times 3 \times K$-sized cube is extracted from its eight neighborhoods as its original input data (the size of spatial neighborhood is empirically determined). To meet the input requirement of spectral-ResNet, the original data is re-arranged into nine pixel vectors, and the length of each pixel vector is K, K is the number of bands, as shown in Figure 2. It should be noted that the 1D kernels are exploited to effectively capture intrinsic spectrum content along the 1D spectral dimensions. In the 1D convolution operation, the input data is convolved with 1D kernels, and then the convolved input data go through the activation function to form the feature vectors. The data is re-arranged in the spectral-ResNet to extract the high-level abstract spectral features. Suppose the set of labeled samples is $L_t^{spectral}$, and the set of the unlabeled samples is denoted as $U_t^{spectral}$.

In the spatial-ResNet model, for a certain pixel in the original HSI, it is natural to consider its neighboring pixels for the extraction of spatial features. However, due to the hundreds of bands along the spectral dimension of HSI, the region-based feature vector will result in too large as an input dimension. This problem could be solved by principal component analysis (PCA), and as PCA is conducted on the pixel-spectrum, the spatial information remains intact. We reduce the spectral dimension of the original HSI to three which is empirically chosen as a trade-off between accuracy and computational complexity with minimum information loss. Then, for each pixel, we choose a relatively large image patch (the patch is 27×27 in our experiment) from its neighborhood window as the input of the spatial-ResNet model, as shown in Figure 2. In each 2D convolutional layer, the image patch is convolved with 2D kernels, then goes through the activation function to form the feature maps. Then the high-level spatial features can be extracted by the spatial-ResNet model. Suppose the set of labeled samples is denoted as $L_t^{spatial}$, and the set of the unlabeled samples is denoted as $U_t^{spatial}$.

2.3. Dual-Strategy Sample Selection Co-Training

The goal of the dual-strategy sample selection co-training algorithm is to select highly confident examples with predicted labels from the unlabeled pool based on spectral and spatial features. These newly labeled examples by each model can boost the performance in the next round of training. Now we introduce the three main components of the iteration algorithm. For clarity, we omit the iteration number t of co-training in the equations below.

2.3.1. New Sample Selection Mechanism Based on Spectral Feature

For spectral-ResNet model, all bands of the labeled data are used to train the model, so we take full advantage of the spectral characteristics and the inherent deep features to select the most confident samples. Thus we proposed a new sample selection mechanism based on spectral feature and deep learning.

Since the spectral information of the same class is similar and the labeled samples is limited, we intend to take the samples with the highest similarity to be the most confident samples for this class. First, we define a distance metric from the test sample and a class dataset. For the t-th iteration of co-training, we get a candidate set for each class after the unlabeled samples $U_t^{spectral}$ was classified by spectral-ResNet. A candidate set is denoted as $H_t^{spectral} = \{(X_1, y_1), \ldots, (X_M, y_M)\}$, where X_M is the candidate set with label y_M, M is the number of the class. Then we try to search for the most confident samples for each class with the spectral similarity metric from the candidate set. For each sample x_M of X_M and labeled set $L_t^{spectral} = \{(L_1, y_1), \ldots, (L_M, y_M)\}$, where L_M is the training set with label y_M, we define the distance metric from x_M to L_M as:

$$d(x_M, L_M) = \inf\{d(x_M, l_M) : l_M \in L_M\}, \tag{3}$$

where inf represents the infimum, l_M is each sample in the training set with label y_M.

Then, the main problem is how to define the distance between two samples. In order to take advantage of the deep features inherent to describe the distribution of the hyperspectral data, we give a definition of a new metric between two hyperspectral samples based on deep learning. Some research [27,28] have been shown that combining the features from lower layers can capture finer features. Moreover, using the hierarchical feature to describe the distribution of the data can alleviate the problem of intra- and inter-class variation in data [29]. Inspired by this observation, for the spectral-ResNet described in Section 2.2, we can extract a multi-level representation for each of the unlabeled samples. The multi-level representation consists of the output of the first convolutional layer and the two building blocks, which are denoted as r_1, r_2, r_3 respectively. Then, the hierarchical representation of each sample is represented as $r = [r_1; r_2; r_3]$. The distance between two samples such as x_M and l_M is given by the distance between two hierarchical representations:

$$d(x_M, l_M) = \|r^{(x_M)} - r^{(l_M)}\|_2^{\frac{1}{2}}. \tag{4}$$

Last, we define a similarity metric between the x_M and the training set L_M based on this distance metric as:

$$s(x_M, L_M) = \exp(-d(x_M, L_M)). \tag{5}$$

For this similarity metric, the most confident samples belonging to L_M are those with distance close to zero, and the corresponding similarity is close to one.

2.3.2. Sample Selection Mechanism Based on Spatial Feature

For spatial-ResNet model, PCA is executed to map the hyperspectral data in the first step, this step can keep spatial information intact but cast away part of the redundant spectral information. Since spatial consistency has been found among neighboring pixels. Neighbors of the labeled samples

are identified using a second-order spatial connectivity by the spatial consistency assumption. Then the most confident samples are selected by the classification of the spatial-ResNet and neighbors of labeled samples.

For illustrative purpose, Figure 4 shows a toy example illustrating the process of sample selection mechanism based on spatial feature. In the left of Figure 4, we display the available labeled samples for two different classes, 1 and 2. These labeled samples are used to train the Spatial-Resnet, and the second-order neighborhood of the labeled samples are labeled by the trained Spatial-Resnet, as illustrated in the upper middle part of Figure 4. In the middle of the lower part of Figure 4, we label the neighbors of the labeled samples by the spatial consistency assumption. Finally, the most confident samples are selected as shown in the right of Figure 4.

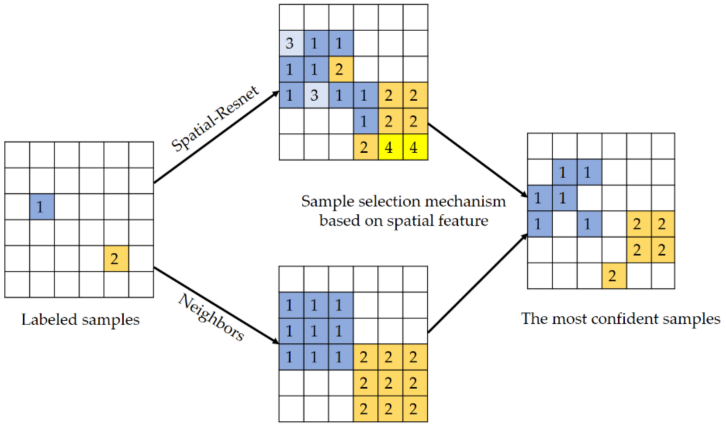


Figure 4. The process of sample selection mechanism based on spatial feature.

2.3.3. Co-Training

Finally, the two well-trained ResNet models are utilized to predict the unlabeled pool over the respective modalities. A highly confident set $H^{spectral}$ and $H^{spatial}$ can be selected with the new sample selection mechanism. Now we attach the spectral data $H^{spectral}$ and spatial data $H^{spatial}$ to update the labeled pool as

$$\begin{aligned} L_{t+1}^{spectral} &= L_t^{spectral} \cup H_t^{spatial}, \\ L_{t+1}^{spatial} &= L_t^{spatial} \cup H_t^{spectral}. \end{aligned} \tag{6}$$

This step basically identifies the labeled samples with the highest confidence scores combined from both views. During the next round training, $H^{spectral}$ will improve the spatial-ResNet model as they are new and informative labeled samples, which is the same with the $H^{spatial}$ for the spectral-ResNet.

3. Experimental Results and Analyses

In this section, the effectiveness of the proposed method is tested in the classification of three open source hyperspectral datasets, namely, the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) Indian Pines datasets, the Reflective Optics System Imaging Spectrometer (ROSIS-03) University of Pavia datasets, the AVIRIS Salinas Valley datasets and one selected Hyperion dataset. In our experiments, firstly, the performance of the proposed method is compared with three state-of-the-art HSI classification methods: (1) CNN [10], a supervised classification using deep CNN to extract the joint spectral–spatial features from HSI; (2) CDL-MD-L [15], a self-training semi-supervised classification approach based on contextual deep learning classification (CDL) and multi-decision labeling (MD-L); (3) Co-DC-CNN, DC-CNN [30] is a dual-channel CNN with non-residual networks from our previous work, we extend it to a co-training approach denoted as Co-DC-CNN. Then, we compared our results against three semi-supervised classification methods based on spectral-spatial feature and co-training: (1) PNGrow [18], a semi-supervised classification algorithm using co-training approach

with the P-N learning scheme, P-expert and N-expert take advantage of the spatial structure and the spectral structure respectively; (2) TT_AL_MSH_MKE [17], tri-training technique for spectral-spatial HSI classification based on an active learning (AL) and a multi-scale homogeneity(MSH), MKE is a combination of MLR, KNN and ELM, and (3) S^2CoTraC [21], the semi-supervised classification algorithm using co-training approach with both spectral information and spatial information which are iteratively extracted at the pixel level via collective inference. All the parameters in the compared methods are set according to the authors' suggestion or tuned to achieve the best performance. It should be noted that the result of TT_AL_MSH_MKE are taken from the results reported by Tan et al. [17]. Additionally, the quantitative comparisons of the classification results are based on class-specific accuracy, overall accuracy (OA), average accuracy (AA), kappa coefficient (κ) and F1-meaure [31].

3.1. Dataset Description and Experimental Settings

The Indian Pine image was recorded by the AVIRIS sensor over the Indian Pines site in Northwestern Indiana. It consists of 145 × 145 pixels and 220 spectral reflectance bands in the wavelength range 0.4–2.5 μm. Twenty spectral bands were removed due to noise and water absorption, and the remaining 200 bands was used for the experiments. The ground-truth data contains sixteen classes, and the false-color composite of the Indian Pines image and the corresponding reference image are shown in Figure 5a,b, respectively.

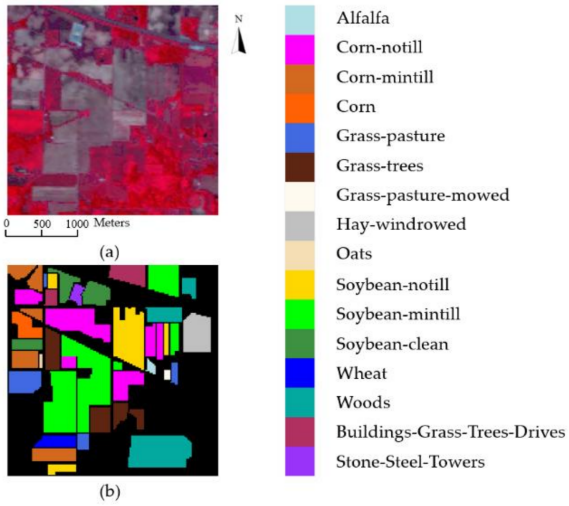


Figure 5. AVIRIS Indian Pines image. (**a**) False-color image. (**b**) Reference image. Black area denotes unlabeled pixels.

The Pavia University image was gathered by the ROSIS-03 sensor during a flight campaign over Pavia, northern Italy, having 610 × 340 pixels. A total of 115 spectral bands were collected, at the range 0.43–0.86 μm. Twelve spectral bands were removed due to noise and the remaining 103 bands were used for classification. Nine land-cover classes were selected, and the true color composite of the University of Pavia image and the corresponding reference image are shown in Figure 6a,b, respectively.

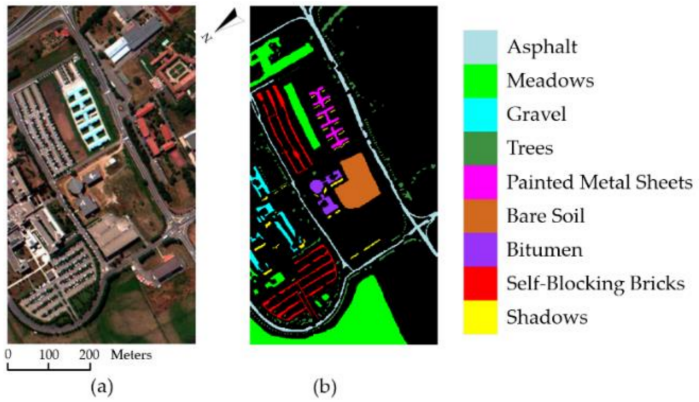


Figure 6. ROSIS-03 University of Pavia image. (**a**) True color image. (**b**) Reference image. Black area denotes unlabeled pixels.

The Salinas Valley image was captured by the AVIRIS sensor over Salinas Valley, California. The image consists of 512 × 217 pixels and 224 spectral bands in the range from 0.4 to 2.5 µm, where 20 water absorption bands were removed. The reference of this image contains sixteen ground-truth classes. Figure 7 shows the true color composite of the Salinas Valley image and the corresponding reference data.

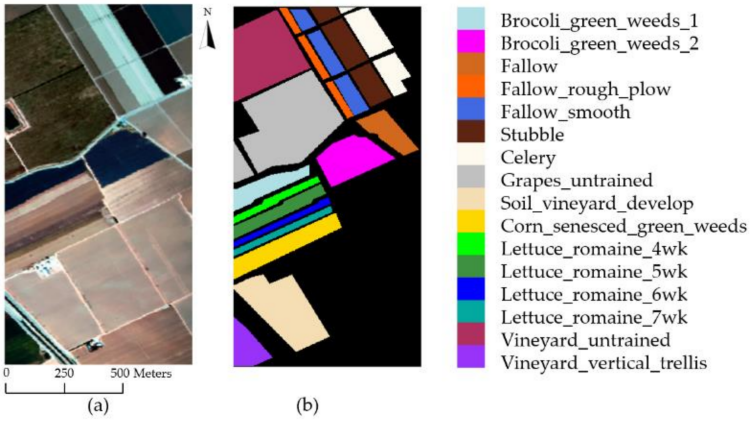


Figure 7. AVIRIS Salinas Valley image. (**a**) True color image. (**b**) Reference image. Black area denotes unlabeled pixels.

The Hyperion image was gathered over Lafayette, Indiana, USA on 14 October 2015. There are 242 spectral bands in the spectral range of 0.4–2.5 µm. After removing the noisy bands and water absorption bands, there are 175 bands remained. A sub-image with size 341 × 307 pixels and thirteen land-cover classes is used as study area. The true color composite of the image and the corresponding reference image are shown in Figure 8a,b, respectively.

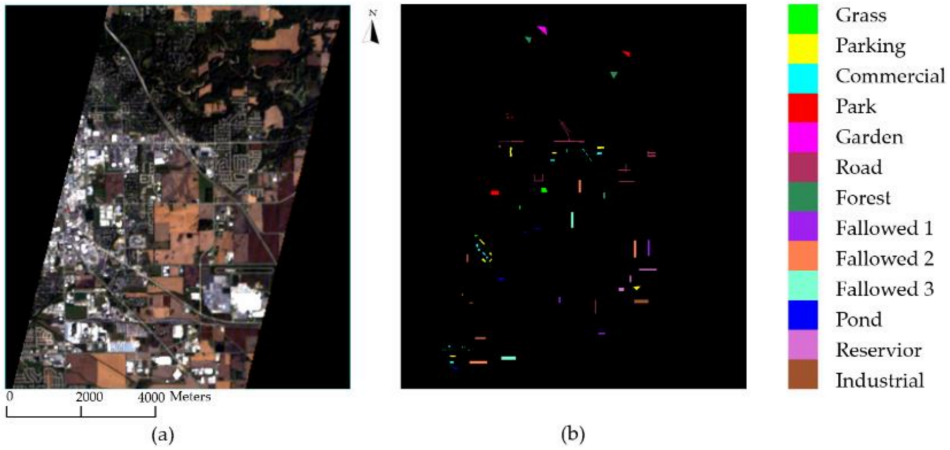


Figure 8. Hyperion image. (**a**) True color image. (**b**) Reference image. Black area denotes unlabeled pixels.

For all the experiments with the four HSI datasets, limited training samples were randomly selected from each class, and the rest of the samples were set as unlabeled data for co-training. To obtain a more convincing estimate of the capabilities of the proposed method, we run the experiment 10 times for each dataset. The training sample sizes of all the experiments were quite limited, which is a challenge for the classification task.

For the four datasets, the structure of the networks was set to the same depth and same width with a fair comparison. For the spectral-ResNet, it contains one convolutional layer and two building blocks. The first layer contains $[3 \times 1, 20]$ convolutional kernels, is followed by one pooling layer with pooling size [2,1] and stride [2,1]. For each building block, the three stacked layers contains $[1 \times 1, 20]$, $[3 \times 1, 20]$ and $[1 \times 1, 80]$ convolutional kernels. Finally, the network ends with a global average pooling, a fully connected layer, and softmax. In training a network, one epoch means one pass of the full training set. This network is trained over 240 epochs (160 epochs with learning rate 0.01, 80 epochs with learning rate 0.001). Each iteration of training network randomly takes 20 samples, where weight decay, momentum and dropout rate are set to 0.0005, 0.9 and 0.5, respectively. For the spatial-ResNet, three Principal Components are extracted from the original HSI and then $27 \times 27 \times 3$-sized image patches are extracted as the input data. The network structure is same as spectral-ResNet. The first layer contains $[3 \times 3, 20]$ convolutional kernels, is followed by one pooling layer with pooling size [2,2] and stride [2,2]. For each building block, the three stacked layers contains $[1 \times 1, 20]$, $[3 \times 3, 20]$ and $[1 \times 1, 80]$ convolutional kernels. It is trained over 200 epochs (140 epochs with learning rate 0.01, 60 epochs with learning rate 0.001). At each iteration of training network, 20 samples are randomly selected from the training set. Weight decay, momentum and dropout rate are set to 0.0005, 0.9 and 0.5, respectively. In order to prevent overfitting, 'dropout' is employed in our method after the average pooling in each branch, the dropout rate is set to 0.5. In the dual-strategy sample selection method, the spectral similarity between the tested samples and training set is empirically set as $s \geq 0.9$ for selection, the eight neighbors of labeled training samples are used as the candidate set.

3.2. Experimental Results on the AVIRIS Indian Pines Dataset

The first group of experiments was conducted on AVIRIS Indian Pines dataset. Firstly, we tested the proposed method in different scenarios, where an increased amount of initial training samples was used, respectively ({5, 10, 15, 20} samples per class). In particular, for Grass-pasture-mowed and Oats, the number of initial training samples is at most 10. For the co-training progress, three iterations of co-training were performed, the iteration number of co-training is denoted as $t = 3$. The detailed results are listed in Table 1, and classifications are shown in Figure 9. We make two observations on Table 1: First, as the initial training samples increase, the OA is in an upward trend until it becomes

stable, and the OA of 20 training samples improves a little bit as compared with 15. Furthermore, in the case of 15 initial training samples, the network structure and sample selection strategy used in the proposed method can be used to train the network well. Second, we analyze the classification results of each class. For the classes with very few samples, Alfalfa, Grass-pasture-mowed and Oats, the classification accuracy had already reached 100% when the initial training samples is 10. Furthermore, when the initial training samples is set as 5 or 10, the classification accuracy for each class is not stable, especially for Corn-notill, Soybean-mintill and Woods, as for those classes the number of samples is large. However, the results are more stable when the initial training samples size is set as 15 and 20.

Table 1. Classification accuracy (%) of the proposed algorithm for AVIRIS Indian Pines with 5, 10, 15 and 20 initial training samples per class and $t = 3$ iterations of co-training.

Class		Labeled Samples Per Class			
No.	**Number of Samples**	**5**	**10**	**15**	**20**
1	46	**100.00 ± 0.00**	**100.00 ± 0.00**	**100.00 ± 0.00**	**100.00 ± 0.00**
2	1428	82.99 ± 6.41	94.56 ± 2.12	97.27 ± 1.07	**97.74 ± 0.52**
3	830	86.16 ± 8.97	94.07 ± 0.99	**97.57 ± 1.39**	96.69 ± 1.14
4	237	**99.57 ± 0.67**	98.40 ± 2.57	99.48 ± 0.72	95.08 ± 3.34
5	483	95.68 ± 1.85	98.87 ± 1.17	99.22 ± 0.66	**99.35 ± 0.41**
6	730	98.83 ± 0.95	99.64 ± 0.76	**99.86 ± 0.22**	99.74 ± 0.27
7	28	97.83 ± 3.64	99.31 ± 1.97	**100.00 ± 0.00**	**100.00 ± 0.00**
8	478	99.83 ± 0.25	99.97 ± 0.07	**100.00 ± 0.00**	**100.00 ± 0.00**
9	20	95.56 ± 5.44	**100.00 ± 0.00**	**100.00 ± 0.00**	**100.00 ± 0.00**
10	972	87.94 ± 3.53	96.09 ± 2.11	98.38 ± 0.92	**98.67 ± 1.34**
11	2455	76.40 ± 11.03	92.07 ± 2.17	96.26 ± 1.64	**97.26 ± 0.46**
12	593	93.31 ± 4.47	95.22 ± 2.36	**98.82 ± 0.70**	98.14 ± 1.52
13	205	**99.42 ± 0.80**	84.93 ± 13.47	92.98 ± 6.97	96.04 ± 3.24
14	1265	96.93 ± 3.67	95.45 ± 2.03	**97.41 ± 0.74**	97.20 ± 0.69
15	386	96.33 ± 2.25	96.44 ± 2.54	**97.71 ± 1.53**	93.03 ± 6.84
16	93	98.86 ± 1.44	**99.55 ± 0.90**	97.44 ± 3.89	99.32 ± 1.15
	OA	88.42 ± 3.07	95.07 ± 1.02	97.66 ± 0.63	**97.66 ± 0.85**
	AA	94.11 ± 1.12	96.54 ± 0.88	98.28 ± 0.71	**98.39 ± 0.49**
	κ	86.90 ± 3.44	94.39 ± 1.16	97.34 ± 0.72	**97.36 ± 0.40**
	F1-measure	90.09 ± 0.01	95.66 ± 0.01	97.87 ± 0.01	**97.27 ± 0.01**

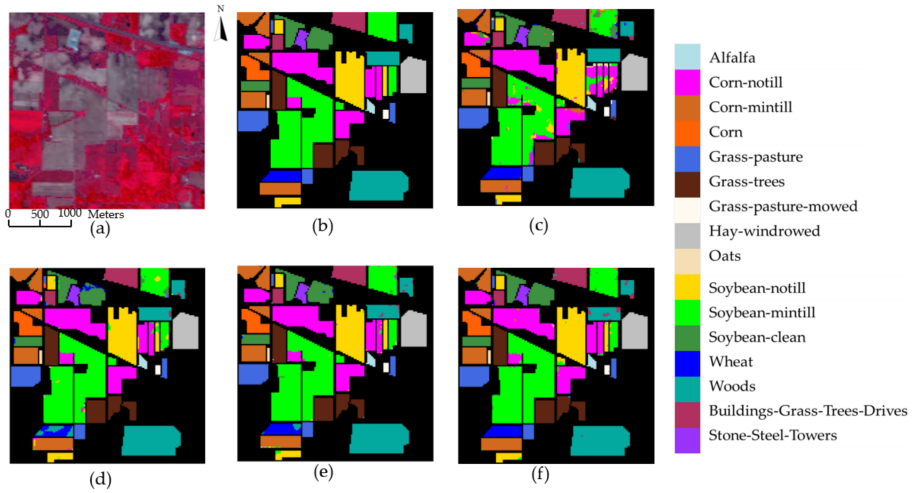


Figure 9. Classification results of AVIRIS Indian Pines. (**a**) False-color image. (**b**) Reference image. (**c**) Label = 5, OA = 92.35%. (**d**) Label = 10, OA = 96.34%. (**e**) Label = 15, OA = 98.38%. (**f**) Label = 20, OA = 98.2%.

Then we compare the proposed method with three HSI classifiers algorithms in Table 2. In order to evaluate the performance of co-training in the proposed algorithm, firstly, we compare the proposed algorithm with a state-of-the-art supervised spectral–spatial deep learning algorithm CNN and a self-training based semi-supervised classification approach CDL-MD-L. It is obvious especially when there is a very small initial training samples that the performance between our method and other two algorithms shows the significant advantage (maximum improvement of 41% in 5 initial training samples per class). To validate the effectiveness of the residual network in the proposed framework, we also compare the proposed algorithm with a co-training CNN with non-residual networks Co-DC-CNN, it can be found in Table 2 that the proposed algorithm achieves a better performance using the residual network.

Table 2. Classification accuracy (%) of state-of-the-art HSI classifiers algorithms on AVIRIS Indian Pines with 5, 10, 15 and 20 initial training samples per class.

Algorithm		**Labeled Samples Per Class**			
		5	**10**	**15**	**20**
CNN	OA	47.33 ± 4.19	64.09 ± 2.76	68.90 ± 1.73	79.62 ± 1.06
	AA	57.71 ± 2.74	78.5 ± 2.61	83.38 ± 1.25	88.73 ± 0.89
	κ	41.95 ± 4.45	59.77 ± 2.74	65.15 ± 1.26	79.93 ± 0.93
CDL-MD-L	OA	74.85	86.46 ± 1.78	90.22	91.54
	AA	72.98	79.30 ± 1.66	85.12	88.02
	κ	74.13	84.63 ± 2.00	88.94	91.06
Co-DC-CNN	OA	85.81 ± 3.33	92.31 ± 1.23	95.13 ± 0.79	94.89 ± 1.02
	AA	91.58 ± 1.32	93.53 ± 1.09	95.37 ± 0.87	95.40 ± 0.79
	κ	84.64 ± 3.39	91.40 ± 1.54	94.88 ± 0.95	94.01 ± 0.74
Proposed	OA	**88.42 ± 3.07**	**95.07 ± 1.02**	**97.66 ± 0.63**	**97.66 ± 0.85**
	AA	**94.11 ± 1.12**	**96.54 ± 0.88**	**98.28 ± 0.71**	**98.39 ± 0.49**
	κ	**86.90 ± 3.44**	**94.39 ± 1.16**	**97.34 ± 0.72**	**97.36 ± 0.40**

Moreover, we compare the proposed method with four different semi-supervised approaches. As the results show in Table 3, the proposed method provides the best performance even with small number of initial training samples. In particular, when the initial training sample is 5, the proposed method obtained the best result OA of 88.42%, which is 6.31% higher than the second best (82.11%) achieved by PNGrow.

Table 3. Classification accuracy (%) of semi-supervised classifiers algorithms on AVIRIS Indian Pines with 5, 10, 15 and 20 initial training samples per class.

Algorithm		**Labeled Samples Per Class**			
		5	**10**	**15**	**20**
PNGrow	OA	82.11 ± 2.69	89.18 ± 1.54	91.80 ± 2.07	93.22 ± 1.10
	AA	88.60 ± 1.2	92.58 ± 1.1	94.26 ± 1.1	94.96 ± 0.7
	κ	79.74 ± 3.0	87.70 ± 1.7	90.66 ± 2.3	92.27 ± 1.2
TT_AL_MSH_MKE	OA	71.05 ± 7.76	79.36 ± 6.95	83.44 ± 5.45	n/d
	AA	80.48 ± 5.86	86.45 ± 4.93	89.38 ± 3.55	n/d
	κ	67.88 ± 8.17	77.02 ± 7.46	81.45 ± 5.91	n/d
S^2CoTraC	OA	69.15 ± 2.25	79.18 ± 0.56	90.40 ± 1.65	93.53 ± 0.69
	AA	82.96 ± 2.64	88.93 ± 1.38	94.83 ± 0.94	94.20 ± 0.41
	κ	65.97 ± 2.30	76.69 ± 0.53	89.10 ± 1.85	92.61 ± 0.78
Proposed	OA	**88.42 ± 3.07**	**95.07 ± 1.02**	**97.66 ± 0.63**	**97.66 ± 0.85**
	AA	**94.11 ± 1.12**	**96.54 ± 0.88**	**98.28 ± 0.71**	**98.39 ± 0.49**
	κ	**86.90 ± 3.44**	**94.39 ± 1.16**	**97.34 ± 0.72**	**97.36 ± 0.40**

3.3. Experimental Results on the ROSIS-03 University of Pavia Dataset

The second experiment was conducted on the ROSIS-03 University of Pavia Dataset. The results are listed in Table 4, and the visual classification results are shown in Figure 10. In this dataset, as the initial training samples increases, the OA improves and the per-class accuracy is more stable. The comparison between our proposed method and state-of-the-art HSI classifiers algorithms are presented in Table 5. The comparison between our proposed method and other semi-supervised methods are presented in Table 6. The proposed method obtained the best result on the 10, 15, 20 initial training samples, but on the 5 initial training samples, the OA and κ is lower than PNGrow. However, the iteration of proposed method is only three where the iteration of PNGrow is ten. In Section 4, we will discuss the relationship between classification results and the number of iterations in co-training.

Table 4. Classification accuracy (%) of the proposed algorithm for ROSIS Pavia University data with 5, 10, 15 and 20 initial training samples per class and $t = 3$ iterations of co-training.

Class		Labeled Samples Per Class			
No.	**Number of Samples**	**5**	**10**	**15**	**20**
1	6631	78.96 ± 9.94	93.40 ± 3.26	97.31 ± 0.66	**98.51 ± 0.72**
2	18,649	83.10 ± 5.93	96.40 ± 2.28	98.88 ± 0.55	**99.28 ± 0.53**
3	2099	89.14 ± 3.76	97.57 ± 1.52	98.55 ± 0.66	**99.18 ± 0.49**
4	3064	97.49 ± 0.35	97.54 ± 0.75	98.03 ± 0.75	**98.71 ± 0.42**
5	1345	**100.00 ± 0.00**	**100.00 ± 0.00**	99.97 ± 0.06	99.99 ± 0.03
6	5029	90.17 ± 5.20	97.58 ± 1.34	99.74 ± 0.28	**99.96 ± 0.04**
7	1330	98.68 ± 0.97	98.35 ± 0.68	**99.93 ± 0.09**	99.90 ± 0.11
8	3682	91.33 ± 2.27	93.15 ± 4.35	97.65 ± 0.69	**98.27 ± 0.74**
9	947	99.05 ± 0.66	99.08 ± 1.02	**99.95 ± 0.08**	99.87 ± 0.25
	OA	86.69 ± 2.94	96.16 ± 1.05	98.64 ± 0.21	**99.16 ± 0.24**
	AA	91.99 ± 1.04	97.01 ± 0.66	98.89 ± 0.12	**99.30 ± 0.14**
	κ	82.94 ± 3.57	94.95 ± 1.36	98.21 ± 0.27	**98.89 ± 0.32**
	F1-measure	86.19 ± 0.01	96.23 ± 0.01	97.76 ± 0.01	**98.60 ± 0.00**

Table 5. Classification accuracy (%) of state-of-the-art HSI classifiers algorithms on ROSIS Pavia University data with 5, 10, 15 and 20 labeled samples per class.

Algorithm		**Labeled Samples Per Class**			
		5	**10**	**15**	**20**
CNN	OA	55.40 ± 3.89	69.02 ± 2.26	72.38 ± 1.26	79.34 ± 0.51
	AA	55.89 ± 3.31	63.13 ± 1.77	69.79 ± 1.37	77.52 ± 0.58
	κ	44.16 ± 3.84	59.86 ± 1.95	64.62 ± 1.48	73.84 ± 0.43
CDL-MD-L	OA	72.85	82.61 ± 2.95	88.04	91.89
	AA	78.58	85.10 ± 2.45	89.12	91.32
	κ	63.71	0.7807 ± 0.03	83.64	88.42
Co-DC-CNN	OA	83.47 ± 3.01	94.99 ± 1.49	95.33 ± 0.32	97.45 ± 0.45
	AA	88.52 ± 1.39	95.51 ± 1.13	96.68 ± 0.27	98.72 ± 0.23
	κ	81.78 ± 3.81	93.63 ± 1.31	95.72 ± 0.49	98.67 ± 0.40
Proposed	OA	**86.69 ± 2.94**	**96.16 ± 1.05**	**98.64 ± 0.21**	**99.16 ± 0.24**
	AA	**91.99 ± 1.04**	**97.01 ± 0.66**	**98.89 ± 0.12**	**99.30 ± 0.14**
	κ	**82.94 ± 3.57**	**94.95 ± 1.36**	**98.21 ± 0.27**	**98.89 ± 0.32**

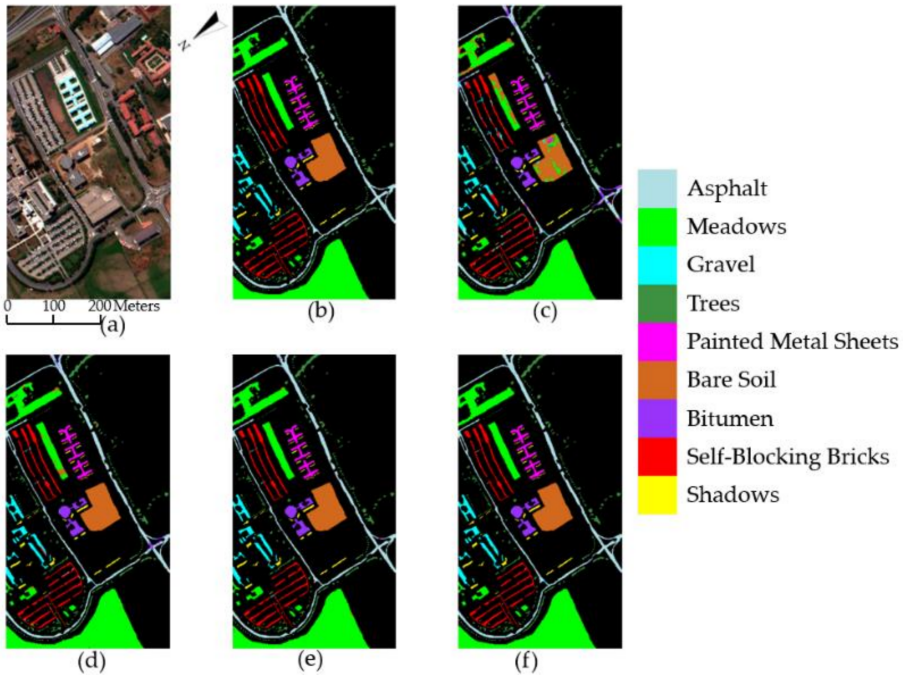


Figure 10. Classification results of ROSIS Pavia University data. (**a**) True color image. (**b**) Reference image. (**c**) Label = 5, OA = 89.7%. (**d**) Label = 10, OA = 97.13%. (**e**) Label = 15, OA = 98.9%. (**f**) Label = 20, OA = 99.40%.

Table 6. Classification accuracy (%) of semi-supervised classifiers algorithms on ROSIS Pavia University data with 5, 10, 15 and 20 labeled samples per class.

Algorithm		Labeled Samples Per Class			
		5	**10**	**15**	**20**
PNGrow	OA	**88.11 ± 2.87**	93.85 ± 2.23	93.77 ± 3.42	96.90 ± 0.90
	AA	91.53 ± 1.3	95.32 ± 0.6	95.96 ± 1.1	97.47 ± 0.4
	κ	**84.64 ± 3.5**	91.95 ± 2.8	91.89 ± 4.3	95.90 ± 1.2
TT-AL-MSH-MKE	OA	79.04 ± 3.95	86.00 ± 3.04	90.20 ± 2.51	n/d
	AA	85.99 ± 3.84	89.80 ± 2.74	92.16 ± 1.92	n/d
	κ	85.99 ± 3.84	82.05 ± 3.87	87.24 ± 3.20	n/d
S^2CoTraC	OA	50.76 ± 1.68	80.75 ± 0.35	82.87 ± 1.42	93.67 ± 1.51
	AA	62.37 ± 1.96	82.37 ± 1.29	90.06 ± 0.84	92.45 ± 0.30
	κ	42.62 ± 1.81	75.11 ± 1.21	78.56 ± 1.73	91.69 ± 1.88
Proposed	OA	86.69 ± 2.94	**96.16 ± 1.05**	**98.64 ± 0.21**	**99.16 ± 0.24**
	AA	**91.99 ± 1.04**	**97.01 ± 0.66**	**98.89 ± 0.12**	**99.30 ± 0.14**
	κ	82.94 ± 3.57	**94.95 ± 1.36**	**98.21 ± 0.27**	**98.89 ± 0.32**

3.4. Experimental Results on the AVIRIS Salinas Valley Dataset

The third experiment was conducted on the AVIRIS Salinas Valley. We tested the proposed method on different initial training samples sizes same as the AVIRIS Indian Pines Dataset. The detailed results are listed in Table 7, and the classifications are shown in Figure 11. In this dataset, as the initial training samples increases, the OA have been improved and the per-class accuracies are more stable. The comparison between our proposed method and other methods are presented in Tables 8 and 9. For each class, the number of sample is large. With large number of samples, the classification results after three iterations of co-training are not ideal, especially on the two classes with high number of samples, Grapes_untrained and Vinyard_untrained.

Table 7. Classification accuracy (%) of the proposed algorithm for AVIRIS Salinas Valley with 5, 10, 15 and 20 initial labeled samples per class and $t = 3$ iterations of co-training.

Class		Labeled Samples Per Class			
No.	**Number of Samples**	**5**	**10**	**15**	**20**
1	2009	99.80 ± 0.15	99.68 ± 0.75	99.89 ± 0.26	**99.92 ± 0.07**
2	3726	98.80 ± 1.80	99.96 ± 0.09	99.76 ± 0.64	**100.00 ± 0.00**
3	1976	99.98 ± 0.04	99.99 ± 0.02	**100.00 ± 0.00**	**100.00 ± 0.00**
4	1394	99.89 ± 0.08	99.98 ± 0.04	**100.00 ± 0.00**	**100.00 ± 0.00**
5	2678	99.15 ± 0.39	99.21 ± 0.33	99.45 ± 0.19	**99.69 ± 0.29**
6	3959	99.91 ± 0.23	**100.00 ± 0.00**	**100.00 ± 0.00**	**100.00 ± 0.00**
7	3579	99.62 ± 0.35	99.92 ± 0.13	99.99 ± 0.02	**100.00 ± 0.00**
8	11,271	84.79 ± 5.97	91.39 ± 3.77	94.11 ± 1.77	**95.45 ± 0.85**
9	6203	99.38 ± 0.46	99.75 ± 0.16	99.84 ± 0.12	**99.89 ± 0.12**
10	3278	98.46 ± 0.61	99.34 ± 0.27	**99.62 ± 0.25**	99.35 ± 0.26
11	1068	99.87 ± 0.16	99.91 ± 0.09	**99.91 ± 0.11**	99.81 ± 0.21
12	1927	99.98 ± 0.03	99.76 ± 0.41	**100.00 ± 0.00**	**100.00 ± 0.00**
13	916	99.73 ± 0.39	99.85 ± 0.17	**99.91 ± 0.10**	99.80 ± 0.22
14	1070	98.51 ± 1.54	99.51 ± 0.27	99.80 ± 0.17	**99.84 ± 0.18**
15	7268	78.20 ± 13.77	94.55 ± 2.20	96.77 ± 3.88	**98.42 ± 0.80**
16	1807	99.10 ± 0.52	99.74 ± 0.35	99.83 ± 0.29	**99.90 ± 0.17**
	OA	93.50 ± 1.40	97.31 ± 0.60	98.23 ± 0.39	**98.75 ± 0.22**
	AA	97.20 ± 0.62	98.91 ± 0.18	99.30 ± 0.20	**99.50 ± 0.10**
	κ	92.77 ± 1.57	97.01 ± 0.66	98.03 ± 0.43	**98.61 ± 0.24**
	F1-measure	96.11 ± 0.01	98.53 ± 0.01	99.03 ± 0.01	**99.46 ± 0.01**

Table 8. Classification accuracy (%) of state-of-the-art HSI classifiers algorithms on AVIRIS Salinas Valley with 5, 10, 15 and 20 initial training samples per class.

Algorithm		**Labeled Samples Per Class**			
		5	**10**	**15**	**20**
CNN	OA	75.96 ± 2.48	76.33 ± 1.24	86.89 ± 0.98	87.67 ± 0.33
	AA	80.19 ± 2.19	85.91 ± 1.75	92.82 ± 0.87	94.4 ± 0.48
	κ	73.26 ± 2.51	73.79 ± 1.51	85.46 ± 0.93	86.38 ± 0.31
Co-DC-CNN	OA	90.18 ± 1.59	94.45 ± 1.12	95.16 ± 0.34	95.72 ± 0.29
	AA	93.63 ± 1.22	95.60 ± 0.68	96.54 ± 0.31	96.68 ± 0.15
	κ	89.87 ± 1.21	94.64 ± 1.03	94.99 ± 0.54	95.70 ± 0.55
Proposed	OA	**93.50 ± 1.40**	**97.31 ± 0.60**	**98.23 ± 0.39**	**98.75 ± 0.22**
	AA	**97.20 ± 0.62**	**98.91 ± 0.18**	**99.30 ± 0.20**	**99.50 ± 0.10**
	κ	**92.77 ± 1.57**	**97.01 ± 0.66**	**98.03 ± 0.43**	**98.61 ± 0.24**

Table 9. Classification accuracy (%) of state-of-the-art semi-supervised classifiers algorithms on AVIRIS Salinas Valley with 5, 10, 15 and 20 initial training samples per class.

Algorithm		**Labeled Samples Per Class**			
		5	**10**	**15**	**20**
PNGrow	OA	**95.35 ± 1.3**	**97.36 ± 0.5**	**98.30 ± 0.4**	98.61 ± 0.2
	AA	96.48 ± 1.0	98.13 ± 0.4	98.60 ± 0.3	98.75 ± 0.2
	κ	**94.83 ± 1.5**	**97.06 ± 0.5**	**98.11 ± 0.5**	98.45 ± 0.2
TT-AL-MSH-MKE	OA	89.32 ± 2.02	90.72 ± 1.38	92.34 ± 1.00	n/d
	AA	93.88 ± 1.11	94.79 ± 0.77	95.67 ± 0.49	n/d
	κ	88.14 ± 3.84	89.68 ± 1.53	91.48 ± 1.11	n/d
S^2CoTraC	OA	77.46 ± 1.06	82.22 ± 0.85	95.82 ± 1.39	94.62 ± 1.17
	AA	89.14 ± 2.16	92.88 ± 2.04	98.59 ± 0.34	98.00 ± 0.66
	κ	75.61 ± 1.21	80.51 ± 0.91	95.36 ± 1.54	94.02 ± 1.31
Proposed	OA	93.50 ± 1.40	97.31 ± 0.60	98.23 ± 0.39	**98.75 ± 0.22**
	AA	**97.20 ± 0.62**	**98.91 ± 0.18**	**99.30 ± 0.20**	**99.50 ± 0.10**
	κ	92.77 ± 1.57	97.01 ± 0.66	98.03 ± 0.43	**98.61 ± 0.24**

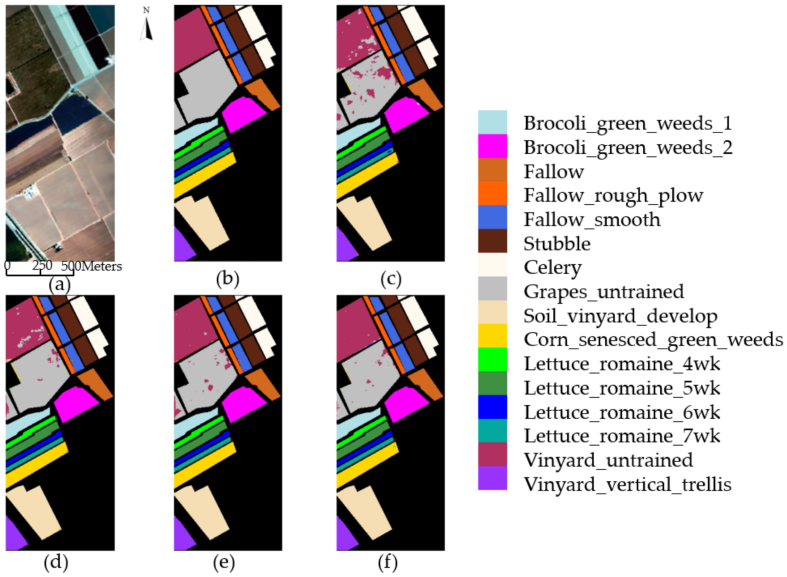


Figure 11. Classification results of AVIRIS Salinas Valley. (**a**) True color image. (**b**) Reference image. (**c**) Label = 5, OA = 95.69%. (**d**) Label = 10, OA = 98.09%. (**e**) Label = 15, OA = 98.59%. (**f**) Label = 20, OA = 99.08%.

3.5. Experimental Results on Hyperion Dataset

The fourth experiment was conducted on a Hyperion dataset. We tested the proposed method on different initial training samples, respectively ({5, 10, 15} samples per class). Because the number of samples per class is small, we didn't test the experiment on 20 initial training samples. The number of iterations in co-training is two. The results are listed in Table 10, and the visual classification results are shown in Figure 12. In this dataset, as the initial training samples increases, the OA have been improved and more stable. The comparison between our proposed method and the state-of-the-art HSI classifiers method are presented in Table 11. According to Table 11, the proposed method performs best.

Table 10. Classification accuracy (%) of the proposed algorithm for Hyperion data with 5, 10 and 15 initial training samples per class and $t = 2$ iterations of co-training.

Class		Labeled Samples Per Class		
No.	**Number of Samples**	**5**	**10**	**15**
1	24	86.84 ± 11.89	89.80 ± 9.09	**91.11 ± 14.49**
2	61	68.45 ± 14.49	84.87 ± 9.72	**89.13 ± 2.66**
3	54	62.92 ± 28.32	77.92 ± 2.16	**86.67 ± 4.21**
4	51	79.06 ± 35.51	90.86 ± 5.36	**98.33 ± 1.52**
5	32	**100.00 ± 0.00**	**100.00 ± 0.00**	**100.00 ± 0.00**
6	148	63.29 ± 13.17	76.71 ± 2.76	**85.71 ± 1.68**
7	49	86.36 ± 7.04	**100.00 ± 0.00**	**100.00 ± 0.00**
8	39	**100.00 ± 0.00**	**100.00 ± 0.00**	**100.00 ± 0.00**
9	82	83.34 ± 11.18	**100.00 ± 0.00**	**100.00 ± 0.00**
10	67	90.59 ± 17.74	99.25 ± 1.38	**100.00 ± 0.00**
11	20	72.23 ± 13.61	91.43 ± 12.15	**100.00 ± 0.00**
12	53	68.75 ± 12.29	**100.00 ± 0.00**	**100.00 ± 0.00**
13	79	64.64 ± 12.66	82.19 ± 8.99	**88.44 ± 1.408**
	OA	77.04 ± 4.48	89.17 ± 1.68	**93.26 ± 0.83**
	AA	80.71 ± 4.01	91.80 ± 1.48	**95.34 ± 0.89**
	κ	74.66 ± 4.90	87.92 ± 1.85	**92.40 ± 0.93**
	F1-measure	78.30 ± 0.05	89.77 ± 0.03	**94.57 ± 0.00**

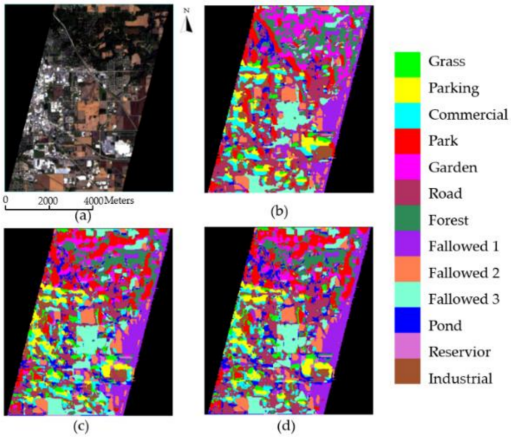


Figure 12. Classification results of Hyperion data with 5, 10 and 15 initial labeled samples per class. (**a**) True color image. (**b**) Label = 5, OA = 82.56%. (**c**) Label = 10, OA = 91.89%. (**d**) Label = 15, OA = 93.97%.

Table 11. Classification accuracy (%) of state-of-the-art HSI classifiers algorithms on Hyperion data with 5, 10 and 15 initial training samples per class.

Algorithm		Labeled Samples Per Class		
		5	10	15
CNN	OA	39.77 ± 4.84	74.56 ± 3.14	72.16 ± 2.26
	AA	37.70 ± 4.61	77.78 ± 2.98	78.72 ± 1.99
	κ	33.09 ± 5.22	71.71 ± 3.27	68.89 ± 2.17
Co-DC-CNN	OA	72.58 ± 4.27	82.47 ± 2.14	90.84 ± 1.30
	AA	75.63 ± 4.09	84.70 ± 1.95	92.63 ± 1.28
	κ	70.44 ± 4.95	81.99 ± 1.89	91.87 ± 1.34
S^2CoTraC	OA	61.35 ± 1.06	79.54 ± 0.85	84.49 ± 1.39
	AA	47.85 ± 2.16	70.51 ± 2.04	79.59 ± 0.34
	κ	56.50 ± 1.21	77.26 ± 0.91	82.71 ± 1.54
Proposed	OA	**77.04 ± 4.48**	**89.17 ± 1.68**	**93.26 ± 0.83**
	AA	**80.71 ± 4.01**	**91.80 ± 1.48**	**95.34 ± 0.89**
	κ	**74.66 ± 4.90**	**87.92 ± 1.85**	**92.40 ± 0.93**

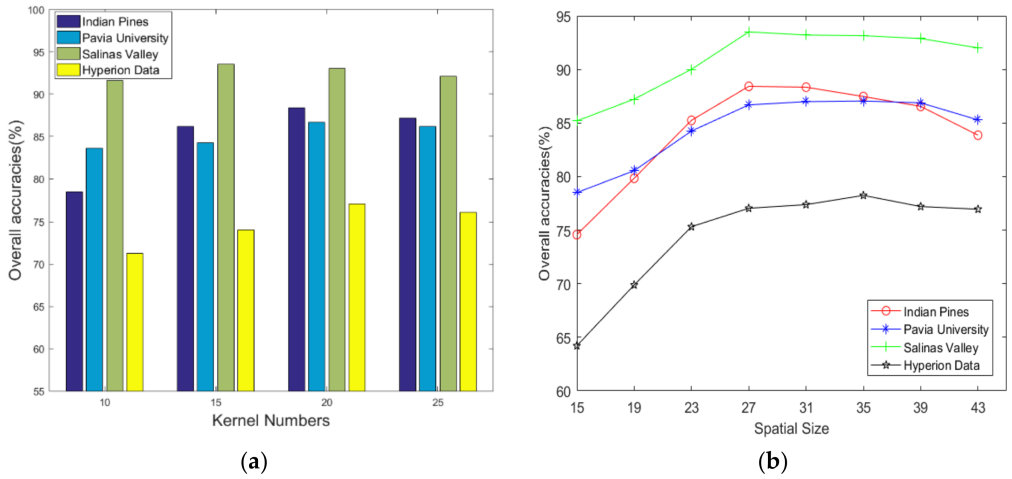
4. Discussion

4.1. Influence of Network Hyper-Parameters

For the proposed classification framework, the choice of network hyper-parameters has an effect on the training process and classification performance. In this section, we investigate the impact on the proposed framework from two aspects: The kernel number of convolutional layers and the spatial size of the input image patch in the spatial-Resnet.

For the "bottleneck" building block, the number of kernel and the quadruple operation is designed by referring to the literature [23]. The two blocks do not use pooling and therefore it does not increase the number of dimensions and feature channels to compensate for the loss of information after pooling, so the adjacent two blocks are with the same width. Then we experimentally verified the kernel number of convolutional layers (the width of the network). We assessed different kernel numbers from 10 to 25 in an interval of 5 in each convolutional layer to find a general framework. The classification results of all datasets using different number of kernels with 5 initial training samples per class in Figure 13a. The framework with 20 kernels in each convolutional layer achieved the highest classification accuracy in the Indian Pines, Pavia University datasets and Hyperion data, and the framework with 15 kernels

obtained the best performance in the Salinas Valley dataset, but it is a little bit higher than the 20 kernels. To maintain data consistency, we use 20 kernels for all datasets.



(**a**) (**b**)

Figure 13. Influence of network hyper-parameters. (**a**) Overall accuracies of different kernel numbers. (**b**) Overall accuracies of different spatial size.

In order to get an appropriate size of spatial neighborhood in the spatial-Resnet, we tested 15 × 15, 19 × 19, 23 × 23, 27 × 27, 31 × 31, 35 × 35, 39 × 39 and 43 × 43. Figure 13b shows the classification results of all datasets using different sizes of spatial neighborhood with 5 initial training samples per class. It can be seen that the accuracy increases quickly with the increase of spatial size at first, then plateaued when the spatial size reached 27 × 27. Therefore, as a trade-off between accuracy and amount of data involved, we empirically choose 27 × 27 as the spatial neighborhood size.

4.2. Effect of the Number of Iterations in Co-Training

As mentioned in Section 3, when the dataset is large as well as the initial training samples is extremely small, the classification results with 3 iterations of co-training are not ideal. Therefore, it is interesting to understand the relationship between classification results and the number of iterations in co-training, using OA and computation time. Experiments are performed on the ROSIS-03 University of Pavia Dataset and the AVIRIS Salinas Valley Dataset by co-training strategy, the initial training labeled set is with 5 per class. The accuracy and computation time are plotted in Figure 14a–c. From the results we see that with large dataset, increase of iteration of co-training will improve the classification results but drastically increase the time costs. Specifically, the first iteration of the co-training actually is a supervised classification. From Figure 14a we can see, on the first iteration of the method, the accuracies are very low. With the samples added based on our sample selection mechanism and sample addition mechanism of co-training, the accuracies are greatly improved. Moreover, as we can see from Figure 14b,c, the computation time increases drastically with the increase of number of iterations of co-training, the main factor comes from the network training time. Therefore, we suggest a moderate number (e.g., three) of iterations in co-training, which keeps satisfactory performance and relatively low time cost simultaneously. We can always set the number of iterations of co-training as 6 to achieve better classification results, if the computational cost is still affordable.

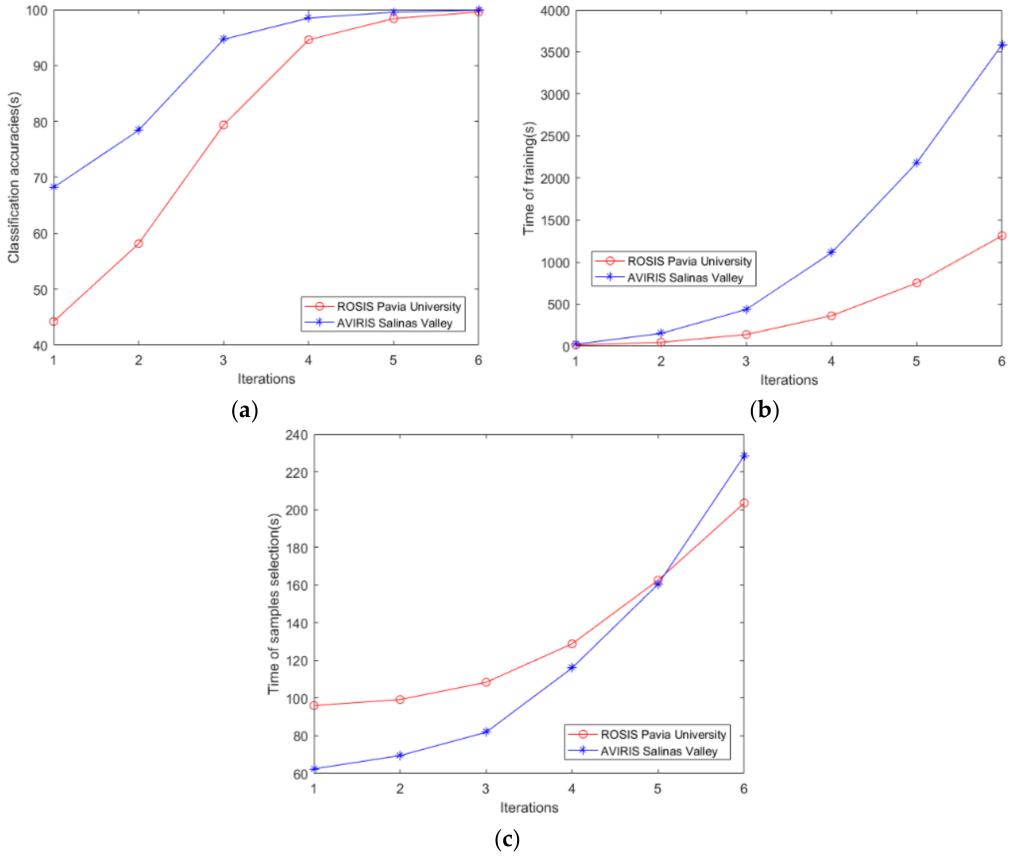


Figure 14. Performance analysis of effect of the number of iterations of co-training. (**a**) Classification accuracies of different iterations. (**b**) Time cost of the network training progress of different iterations of co-training. (**c**) Time cost of the sample selection progress of different iterations of co-training.

4.3. Sample Selection Mechanism Analysis in Co-Training

The effectiveness of the proposed co-training method has been validated, as it not only uses the labeled data, but also exploits and labels the most confident unlabeled data to help learning. In this part, we analyze the performance of the dual-strategy sample selection method in co-training. Experiments are carried out on two HSI data which have relatively large number of samples-the ROSIS-03 University of Pavia Dataset and the AVIRIS Salinas Valley Dataset. Tables 12 and 13 display for each iteration of co-training method on the two HSI data with 5 per class initial training data set, the number of initial training samples, the number of selected samples from spectral-spatial feature and classification results. It should be noted that the accuracies of the selected samples are also listed next to the number of selected samples in Tables 12 and 13.

Table 12. Sample selection mechanism analysis on ROSIS Pavia University data.

	Iteration 1	**Iteration 2**	**Iteration 3**	**Iteration 4**
training samples (spectral)	45 ± 0.00	108 ± 3	537 ± 8	1136 ± 8
training samples (spatial)	45 ± 0.00	215 ± 1	462 ± 12	1439 ± 43
selected samples (spectral)	170 ± 1 (99.19%)	248 ± 13 (99.45%)	977 ± 55 (99.88%)	1167 ± 47 (99.98%)
selected samples (spatial)	63 ± 3 (100.00%)	429 ± 11 (99.83%)	598 ± 16 (99.96%)	1459 ± 76 (99.91%)
OA (%)	54.2 ± 2.12	67.81 ± 4.85	86.69 ± 2.94	97.58 ± 0.13

Table 13. Sample selection mechanism analysis on the AVIRIS Salinas Valley.

	Iteration 1	Iteration 2	Iteration 3	Iteration 4
training samples (spectral)	80 ± 0.00	462 ± 66	1510 ± 19	3659 ± 149
training samples (spatial)	80 ± 0.00	557 ± 18	1473 ± 13	3936 ± 105
selected samples (spectral)	477 ± 18 (99.26%)	916 ± 31 (99.52%)	2463 ± 115 (99.48%)	3552 ± 102 (99.84%)
selected samples (spatial)	382 ± 66 (100.00%)	1048 ± 47 (99.79%)	2149 ± 130 (99.93%)	3816 ± 80 (99.87%)
OA (%)	67.02 ± 2.29	80.44 ± 1.87	93.5 ± 1.42	98.32 ± 0.52

Firstly, as is shown in first column of Table 12, the number of initial training samples and selected samples are small and the corresponding classification results (54.2%) are relatively poor, which is normal. However, after two iterations of co-training with new samples added, the classification results have been greatly improved. It can be seen that the selected samples by the proposed sample selection mechanism can effectively promote the training of the network and improve the classification results. Secondly, again from the first column of the Table 12, in the first iteration of co-training, the number of samples selected from the two models have a large difference (e.g., 170 and 63). However, the selected samples are used to augment the training set of the other model in the next iteration of co-training, which pushed the network performance to excel. Furthermore, the selected samples from the spatial view with local distributions and the selected samples from the spectral view with global distributions, they added into each other's strength and facilitate the selection of samples next round. Third, considering that the proposed method is based on the deep learning framework, the corresponding training time is relatively longer. The number of selected samples of each iteration of co-training is also relatively larger than other co-training based algorithms [17,18]. Although there are mislabeled samples, its number is extremely small compared to the total training samples, and the impact on network training is negligible. Overall, it shows the robustness of our proposed network and the effectiveness of the co-training structure. The same conclusion can be obtained by analyzing the AVIRIS Salinas Valley in Table 13.

5. Conclusions

This paper proposed a semi-supervised deep learning framework for HSI classification, which is capable of reducing the dependence of deep learning method on large-scale manually labeled HSI data. The key to the framework are two parts: (1) The spectral- and spatial-ResNet for extracting the spectral features and spatial features and (2) the dual-strategy sample selection co-training algorithm for effective semi-supervised learning. Experimental results on the benchmark HSI data and a selected Hyperion data demonstrate the effectiveness of our approach. In terms of future research, we plan to design an adaptive network structure to better classify different HSI data. Moreover, it would be interesting to investigate a completely unsupervised setting where an advanced clustering algorithm can be used to initialize the network parameters.

Acknowledgments: This work was supported by National Key Research and Development Program (2016YFB0502502), Foundation Project for Advanced Research Field (614023804016HK03002), Shaanxi International Scientific and Technological Cooperation Project (2017KW-006). The authors wish to thank Annalisa Appice for providing the code of S^2CoTraC.

Author Contributions: All the authors made significant contributions to this work. Bei Fang and Ying Li devised the approach and analyzed the data; Jonathan Cheung-Wai Chan helped design the experiments and provided advice for the preparation and revision of the work; Bei Fang performed the experiments; and Haokui Zhang helped with the experiments.

Conflicts of Interest: The authors declare no competing financial interests. The founding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

References

1. Landgrebe, D. Hyperspectral image data analysis. *IEEE Signal Proc. Mag.* **2002**, *19*, 17–28. [CrossRef]
2. Bioucas-Dias, J.M.; Plaza, A.; Camps-Valls, G.; Scheunders, P. Hyperspectral remote sensing data analysis and future challenges. *IEEE Geosci. Remote Sens. Mag.* **2013**, *1*, 6–36. [CrossRef]
3. Agapiou, A.; Hadjimitsis, D.G.; Alexakis, D.D. Evaluation of broadband and narrowband vegetation indices for the identification of archaeological crop marks. *Remote Sens.* **2012**, *4*, 3892–3919. [CrossRef]
4. Yokoya, N.; Chan, J.C.-W.; Segl, K. Potential of resolution-enhanced hyperspectral data for mineral mapping using simulated EnMAP and Sentinel-2 images. *Remote Sens.* **2016**, *8*, 172. [CrossRef]
5. Lecun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
6. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
7. Chen, Y.; Lin, Z.; Zhao, X.; Wang, G. Deep learning-based classification of hyperspectral data. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 2094–2107. [CrossRef]
8. Chen, Y.; Zhao, X.; Jia, X. Spectral–spatial classification of hyperspectral data based on deep belief network. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2015**, *8*, 1–12. [CrossRef]
9. Yue, J.; Zhao, W.; Mao, S.; Liu, H. Spectral-spatial classification of hyperspectral images using deep convolutional neural networks. *Remote Sens. Lett.* **2015**, *6*, 468–477. [CrossRef]
10. Chen, Y.; Jiang, H.; Li, C.; Jia, X.; Ghamisi, P. Deep feature extraction and classification of hyperspectral images based on convolutional neural networks. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 6232–6251. [CrossRef]
11. Li, Y.; Zhang, H.; Shen, Q. Spectral-spatial classification of hyperspectral imagery with 3D convolutional neural network. *Remote Sens.* **2017**, *9*, 67. [CrossRef]
12. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Li, F.F. ImageNet: A large-scale hierarchical image database. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
13. Yang, J.; Zhao, Y.Q.; Chan, J.C.-W. Learning and transferring deep joint spectral-spatial features for hyperspectral classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 4729–4742. [CrossRef]
14. Ma, X.; Wang, H.; Geng, J. Spectral–spatial classification of hyperspectral image based on deep auto-encoder. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2016**, *9*, 4073–4085. [CrossRef]
15. Ma, X.; Wang, H.; Wang, J. Semisupervised classification for hyperspectral image based on multi-decision labeling and deep feature learning. *ISPRS J. Photogramm. Remote Sens.* **2016**, *120*, 99–107. [CrossRef]
16. Zhao, W.; Du, S. Spectral–spatial feature extraction for hyperspectral image classification: A dimension reduction and deep learning approach. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 4544–4554. [CrossRef]
17. Tan, K.; Zhu, J.; Du, Q.; Wu, L.; Du, P. A novel tri-training technique for semi-supervised classification of hyperspectral images based on diversity measurement. *Remote Sens.* **2016**, *8*, 749. [CrossRef]
18. Romaszewski, M.; Głomb, P.; Cholewa, M. Semi-supervised hyperspectral classification from a small number of training samples using a co-training approach. *ISPRS J. Photogramm. Remote Sens.* **2016**, *121*, 60–76. [CrossRef]
19. Zhang, X.; Song, Q.; Liu, R.; Wang, W.; Jiao, L. Modified co-training with spectral and spatial views for semisupervised hyperspectral image classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 2044–2055. [CrossRef]
20. Blum, A.; Mitchell, T. Combining labeled and unlabeled data with co-training. In Proceedings of the 11th Annual Conference on Computational Learning Theory, Madison, WI, USA, 24–26 July 1998; pp. 92–100.
21. Appice, A.; Guccione, P.; Malerba, D. A novel spectral-spatial co-training algorithm for the transductive classification of hyperspectral imagery data. *Pattern Recognit.* **2017**, *63*, 229–245. [CrossRef]
22. Tan, K.; Li, E.; Du, Q.; Du, P. An efficient semi-supervised classification approach for hyperspectral imagery. *ISPRS J. Photogramm. Remote Sens.* **2014**, *97*, 36–45. [CrossRef]
23. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
24. Zhong, Z.; Li, J.; Ma, L.F.; Jiang, H.; Zhao, H. Deep residual networks for hyperspectral image classification. In Proceedings of the IEEE International Geoscience & Remote Sensing Symposium, Fort Worth, TX, USA, 23–28 July 2017.

25. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 448–456.
26. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105. [CrossRef]
27. Liu, W.; Anguelov, D.; Eran, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 21–37.
28. Huang, G.; Liu, Z.; Weinberger, K.Q.; Laurens, V.D.M. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
29. Hu, J.; Lu, J.; Tan, Y.P. Discriminative deep metric learning for face verification in the wild. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 24–27 June 2014; pp. 1875–1882.
30. Zhang, H.K.; Li, Y.; Zhang, Y.Z.; Shen, Q. Spectral-spatial classification of hyperspectral imagery using a dual-channel convolutional neural network. *Remote Sens. Lett.* **2017**, *8*, 438–447. [CrossRef]
31. Powers, D.M. Evaluation: From precision, recall and F-measure to ROC, Informedness, Markedness and Correlation. *J. Mach. Learn. Technol.* **2011**, *2*, 37–63.