


Article

Spectral-Spatial Attention Networks for Hyperspectral Image Classification

Xiaoguang Mei ^{1,2} , Erting Pan ¹, Yong Ma ^{1,2}, Xiaobing Dai ^{1,2}, Jun Huang ^{1,2}, Fan Fan ^{1,2,*}, Qinglei Du ¹, Hong Zheng ¹ and Jiayi Ma ^{1,2,3}

¹ Electronic Information School, Wuhan University, Wuhan 430072, China; meixiaoguang@gmail.com (X.M.); panerting@whu.edu.cn (E.P.); mayong@whu.edu.cn (Y.M.); xiaobing@whu.edu.cn (X.D.); junhwong@whu.edu.cn (J.H.); dql822@163.com (Q.D.); zh@whu.edu.cn (H.Z.); jyma2010@gmail.com (J.M.)

² Institute of Aerospace Science and Technology, Wuhan University, Wuhan 430072, China

³ Hubei Key Laboratory of Advanced Control and Intelligent Automation for Complex Systems, Wuhan 430074, China

* Correspondence: fanfan@whu.edu.cn

Received: 12 March 2019; Accepted: 20 April 2019; Published: 23 April 2019



Abstract: Many deep learning models, such as convolutional neural network (CNN) and recurrent neural network (RNN), have been successfully applied to extracting deep features for hyperspectral tasks. Hyperspectral image classification allows distinguishing the characterization of land covers by utilizing their abundant information. Motivated by the attention mechanism of the human visual system, in this study, we propose a spectral-spatial attention network for hyperspectral image classification. In our method, RNN with attention can learn inner spectral correlations within a continuous spectrum, while CNN with attention is designed to focus on saliency features and spatial relevance between neighboring pixels in the spatial dimension. Experimental results demonstrate that our method can fully utilize the spectral and spatial information to obtain competitive performance.

Keywords: hyperspectral image classification; attention mechanism; RNN; CNN

1. Introduction

Hyperspectral imaging, also known as imaging spectroscopy, captures the electromagnetic energy reflected or emitted from the same area over hundreds of narrow, continuous spectral bands from the visible to infrared wavelength ranges [1–4]. Hyperspectral images (HSIs) captured from land surface-observing aircrafts or satellites have become increasingly important in environmental monitoring, urban planning, mining, defense and agriculture due to their rich spectral information [2,5,6]. These images are combined to form a three-dimensional (x, y, λ) hyperspectral data cube for processing and analyzing, where x and y represent two spatial dimensions of the scene, and λ represents the spectral dimension (comprising of a range of wavelengths).

Hyperspectral image classification, which assigns every pixel vector to a certain set of classes, is one of the major tasks in the analysis of HSIs, and it has received much attention from researchers. Numerous traditional methods, such as support vector machine (SVM) [7] and k-nearest neighbor (KNN) [2], have been proposed. However, these approaches disregard the correlations among pixels in spatial axes and cause a waste of spatial information. Jiang et al. [8] proposed an unsupervised superpixel wise principle component analysis to learn the intrinsic low-level features of different homogeneous regions by segmenting the entire HSI based on superpixel segmentation. It takes full advantage of spatial information contained in the HSIs. Thus, spectral-spatial based methods improve classification performance because they incorporate additional spatial information from an HSI. For example, Roscher et al. [9] took spectral as well as spatial information by an incremental

learning strategy for import vector machines and discriminative random fields. Another highlight of this work was the concept of self-training for sequential classification of HSI, which was comprised of the inclusion of new training samples to increase the classification accuracy and the deletion of non-informative samples to be memory- and runtime-efficient. Li et al. [10] constructed a family of generalized composite kernels by utilizing spectral and spatial information from HSI data. Jiang et al. [11] developed a random label propagation algorithm, which constructed a spectral-spatial probability transfer matrix that simultaneously considered the spectral similarity and superpixel based spatial information to cleanse the label noise under the label propagation framework.

1.1. Motivation

Deep learning algorithms have been introduced to modern HSI analysis due to their outstanding predictive power, and they can extract more discriminative features and achieve a better performance than traditional shallow classifiers [2,12]. Deep models, such as networks with 1D [13,14], 2D [15], and 3D [16] convolutional layers, have been proposed for hyperspectral data analysis.

Methods with a 1D network take spectra as input and only use spectral information to learn features. Mou et al. [13] utilized recurrent neural network (RNN) to model pixel spectra in an HSI as 1D sequence for classification, and they found that the modified gated recurrent unit (GRU) outperforms traditional approaches and the baseline convolutional neural network (CNN). Given that spatial information has been proven to be useful in improving the interpretation of HSI classification results, the study of classification models based on deep spectral-spatial features has been promoted. For example, Yang et al. [15] designed a two-CNN model to learn the spectral features and spatial features jointly. Cao et al. [17] used a CNN in combination with a Markov random field in a unified Bayesian framework to classify HSI pixel vectors. Spatial-spectral unified network [18] combined a spectral dimensional band grouping-based long short-term memory (LSTM) model with 2D CNN for spatial features and integrated the spectral finite element (FE), spatial FE, and classifier training into a unified neural network. The result showed that the full use of spectral and spatial information can considerably improve accuracy.

The attention mechanism, which becomes a vital part in human perception, is based on a reasonable assumption that human vision does not process an entire image at once, and it only focuses on specific parts of the entire visual space at “high resolution” while perceiving the surrounding in “low resolution” [19,20]. Hence, this mechanism heightens the sensitivity to features containing the most valuable information. Several attempts have been exerted to incorporate attention mechanism as an effective technique processing into visual tasks to strengthen some features and to improve the performance as a result. It has been proven to be productive in many applications, including image captioning [21], matching [22–25] and saliency detection [26].

Attention mechanism enables models to focus on key pieces of the feature space and differentiate irrelevant information [27]. It was first introduced for language translation [28], which learned to focus on particular words or phrases when translating sentences, showing large performance gains especially on long sequences. Considering the spectral dimension data in HSIs as sequence data, attention mechanism can capture the high spectral correlation between adjacent spectra by the above method completely.

Self-attention proposed by Lin et al. [29] uses attention scores to weight all features to obtain salient features. Pei et al. [30] designed a specific spatially-indexed attention mechanism among the convolutional layers to extract the salient facial regions in each individual image and a temporal attention layer to assign attention weights to each frame. Inspired by them, since local features at neighboring spatial positions have high relevance in HSI's spatial domain, adding attention mechanism is helpful in learning spatial dependence and saliency features.

1.2. Contribution

The major contributions in this paper involve the following three aspects:

- We design a joint network with a spectral attention bi-directional RNN branch and a spatial attention CNN branch to extract spectral-spatial features for HSI classification. An attention mechanism is used to emphasize meaningful features along the two branches, as shown in Figure 1. Our goal is to improve representation ability by using the attention mechanism, namely, to focus on the correlations between adjacent spectral dimensions and the spatial dependency in spatial domain, as well as to suppress unnecessary features.
- A bi-directional RNN with an attention mechanism is designed for spectral information in both backward and forward directions. For each pixel, a spectral vector is decomposed into a set of ordered single data and fed into GRU units one by one. Additional attention weights strengthen the spectral correlation between spectrum channels. We compare the attention RNN to the ordinary bi-directional RNN, and the experimental results in Tables 7–9 have proven its effectiveness for the classification with spectral information.
- For spatial axes, we add attention to 2D CNN and train this model on the image patch around the pixel. Compared with the average consideration of each image region, the attention parameter assigns a greater weight to the key parts to make the model focus on the primary features. The classification results of attention CNN and CNN in Tables 7–9 show that the central pixel is classified better by adding attention weight.

The remainder of this paper is organized as follows. Section 2 briefly introduces the related works. Section 3 describes the proposed method for HSI classification, including the two-branch network and co-training. The information of datasets used in this work and the experimental results are given in Section 4. Finally, Section 5 concludes this paper briefly.

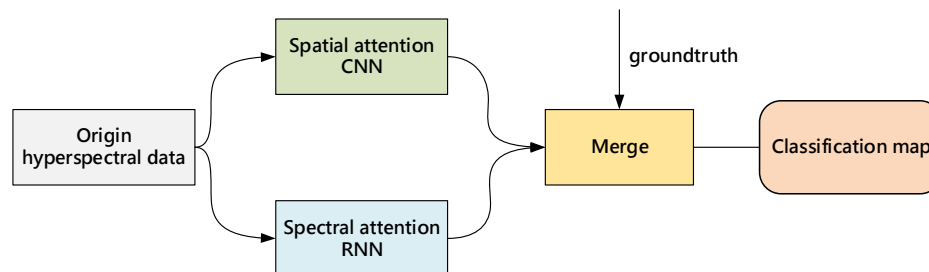


Figure 1. Our framework for hyperspectral images (HSI) classification. It consists of a spatial attention convolutional neural network (CNN) branch and a spectral attention recurrent neural network (RNN) branch, followed by a merge block which learns spectral-spatial features jointly.

2. Related Works

In this section, we mainly recall the background information of bidirectional RNN, CNN and attention mechanism.

2.1. Bi-Directional Recurrent Network

RNN, which extends conventional feedforward neural networks with loops in connections, has gained significant attention for solving many challenging problems involving sequential data analysis, such as speech recognition and language modeling [31,32]. Unlike feedforward neural networks, RNN is called recurrent because of its recurrent hidden state, whose activation at each step depends on the previous computations. RNN has a memory function, which can remember the information about what has been calculated so far.

The architecture of RNN is illustrated in the left part of Figure 2. For a hidden layer in RNN, which maintains a hidden state at each time iteration, it receives the input vector x , and generates the output vector y . The unfolded structure of a bi-directional RNN (Bi-RNN), shown in the right part of Figure 2, presents the calculation process. Bi-RNN connects two hidden layers running in opposite directions to a single output, allowing them to receive information from both past and future states. Neither of these output states is connected to inputs of the opposite directions. By simultaneously employing both directions of input data, information both from the past and future can be used to calculate the output.

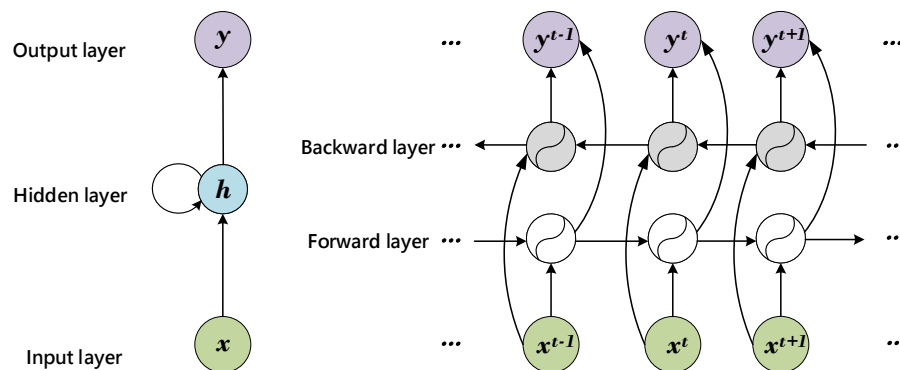


Figure 2. The left part is a brief illustration of RNN, and the right part shows the unfold bidirectional RNN (Bi-RNN) structure.

LSTM [33] and GRU [34] are introduced to learn long-term dependencies and alleviate the vanishing gradient problem. These two architectures do not have any fundamental difference from RNN, but they use different functions to compute the hidden state. Compared with LSTM, GRU does not maintain a cell state C and uses two gates instead of three. GRUs have fewer parameters and thus may be trained a bit faster and need less data to generalize.

2.2. CNN

Another popular deep learning model for vision tasks is CNN [35]. Fundamentally, the mammalian visual system has a spatial hierarchy. Inspired by this, CNN has a trainable multilayer architecture composed of a series of convolution layers, non-linearity layers, and pooling layers stacked alternately. It is used to learn low-level features such as edges or textures and high-level features with more discriminative information [36–40]. A typical CNN structure is shown in Figure 3.

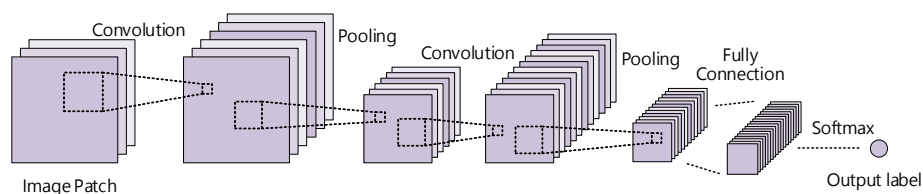


Figure 3. Illustration of typical CNN structure.

In the convolution layer, rather than being fully connected to the input, each hidden layer unit is connected via shared weights to the local receptive field around the input, which might be k two-dimensional feature maps of size $m \times n$. The convolution layer computes convolution of input feature maps x^i with convolutional kernel W^i of size $l \times l \times q$, followed by an element-wise nonlinear activation function. Activity of the i_{th} feature map is $C^i = \sum_j^q W^i \times X^j + b^i$ in the i_{th} layer, where b^i is the bias term for the i_{th} feature map, X^j is the j_{th} channel of the previous layer.

The non-linear activation function summarizes the responses at several input locations, and it computes the output feature map $p^i = f(c^i)$ commonly via a rectified linear unit (ReLU) $f(x) = \max(0, x)$. The pooling layer computes the maximum or average value within a small patch of each feature map, and the most common type is max pooling. The pooling operation offers invariance by reducing the resolution of feature maps. After completing the stacked layers, fully connected layers and a softmax layer are usually adopted to predict the classification labels. Compared with other neural networks, CNN is easier to train with its fewer connections and parameters because of weight sharing and local connection scheme.

2.3. Attention Mechanism

Using attention mechanism, neural networks focus on a certain part of the given information, and every pixel has an independent weight, highlighting discriminative and effective features, and weakening information detrimental to classification.

Spatial Transformer Networks [41] which intelligently focused on a particular area of image was a special case of attention. Moreover, Kim et al. [42] used a joint residual attention model which utilized the attention mechanism to select the most valuable visual information so as to enhance language feature selection and feature extraction for visual question-answering problems. Additionally, Yang et al. [43] proposed an attention mechanism to extract additional meaningful information on the transition layer and passed to the next feature extraction block for subsequent feature exploitation. As for HSI classification, a proposed network [20] was constructed by stacking the proposed attention inception module and it could adaptively learn the network architecture by dynamically routing between the attention inception modules. They designed a novel neural network which has a “trunk branch” with a feedback attention mechanism and a “mask branch” with a gate control attention mechanism to perform pixel-wise classification for very high-resolution remote sensing images.

As noted in previous studies, many networks have adopted the attention mechanism to model the internal relationships and dependencies of the original data through global information, assigning higher priority to more informative areas. The acquired attention weight map can be used for feature recalibration. It simulates the biological process which causes human visual systems to be instantly attracted to a tiny piece of important information in an intricate image. We can relearn feature-based weights for more relevant and noteworthy information.

3. Methods

There are three subsections playing crucial roles in our methodology: A bidirectional RNN-based spectral attention feature learner, a CNN-based spatial attention feature learner and a co-training model.

In our work, attention is of much concern. For spectral classification, considering that each pixel can be represented as a continuous spectral curve that contains rich spectrum characters, we can focus on the inter-band relationship of features by attention. In spatial dimensions, we regard spatial features as complements to spectral ones; this branch improves the representation of interests and focuses on the inter-spatial relationships of features by exploiting spatial attention to CNN. Then, we concatenate two branches and feed them to the fully connected layers to learn high-level joint spectral-spatial features and acquire a prediction class after a softmax layer.

3.1. Attention with RNN for Spectral Classification

RNNs are popular architectures for modeling various sequential problems, and Bi-RNN is proposed to make full use of both latter and previous information. By considering all spectra of a hyperspectral pixel as a sequence, we develop a Bi-RNN model, containing a forward GRU layer and a backward GRU layer, as illustrated in Figure 4. Our model processes the input in both forward and backward directions to the same output layer with two separate hidden layers.

Its input is a spectral vector of one hyperspectral vector x , $x = (x_1, x_2, \dots, x_n)$, and the bi-directional hidden vector is calculated as:

Forward hidden state:

$$\vec{h}_t = f(\vec{W}x_t + \vec{V}h_{t-1} + \vec{b}), \tag{1}$$

Backward hidden state:

$$\overleftarrow{h}_t = f(\overleftarrow{W}x_t + \overleftarrow{V}h_{t+1} + \overleftarrow{b}), \tag{2}$$

where t ranges from the first spectral band 1 to the last n_{th} one, the coefficient matrices \overleftarrow{W} and \vec{W} are from the input at the present step, \vec{V} is from the hidden state h_{t-1} at the previous step, \overleftarrow{V} is from h_{t+1} at the succeeding step, f is the nonlinear activation of the hidden layer, and the memory of the input as the output of this encoder is g_t :

$$g_t = \text{concat}[\vec{h}_t, \overleftarrow{h}_t], \tag{3}$$

where $\text{concat}(\cdot)$ is a function of concatenation between the forward hidden state and backward hidden state.

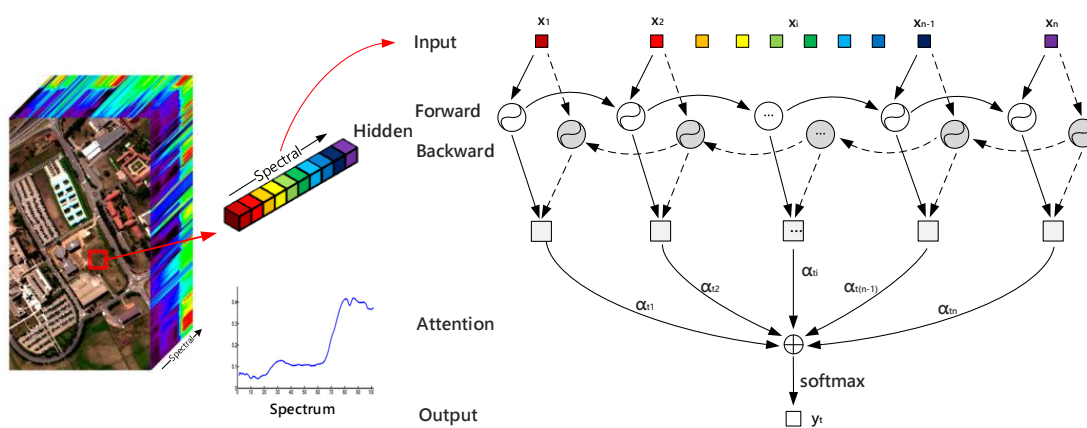


Figure 4. Bi-RNN model with attention mechanism for spectral classification. Every pixel vector is regarded as a sequence, the hundreds of spectral bands are input into the gated recurrent unit (GRU) cell one by one. Both forward and backward features are captured by Bi-RNN and re-weighted by the attention layer.

Bi-RNN allows the spectral vector to be fed into one by one to learn continuous spectrum features with forward and backward directions. If we directly sum and average the data of each spectral band, it means that each spectral channel contributes equally to the classification task. The fact is that the spectrum is a continuous curve with peaks and troughs, rather than a straight line with a fixed value. Therefore, some bands in the spectrum should have a smaller weight while those key spectral bands should have a greater weight. Introducing the attention mechanism into the Bi-RNN, our model assigns an appropriate weight to each spectral channel and makes the model capture inner spectral relationships and classify much better.

Compared with the traditional RNN model that treats the input in the same manner, we add an attention layer to decode different spectral information to learn many characteristics. Our attention layer can be defined as follows:

$$e_{it} = \tanh(W_i g_t + b_i), \tag{4}$$

$$\alpha_{it} = \text{softmax}(W_i' e_{it} + b_i'), \tag{5}$$

where W_i and W_i' are transformation matrices, b_i and b_i' are bias terms, and the $\text{softmax}(\cdot)$ is to map the non-normalized output to a probability distribution and constrain output to be in the interval $(0, 1)$.

So we can compute the predicted label y_t of pixel x as follows:

$$y_t = U[g_t, \alpha], \quad (6)$$

where $U(\cdot)$ is a function of summing over all states which are weighted by their corresponding attention weights.

Equation (4) is a one-layer neural network. This layer rearranges the state of Bi-RNN in its current vector space, and then the tanh activation transforms it to get e_{it} as a new hidden representation of h_i . The attention weight α is produced through the softmax layer, formulated as Equation (5), where we measure the importance of input based on the relevance of e_t with another channel-wise vector. After obtaining the newly learned attention weights, we update the label representation vector y using the soft-attention operation show in Equation (6).

With the attention mechanism, our model adopts a more reasonable explanation that some spectral data play a key role and some of them are meaningless. Meanwhile, our Bi-RNN model can better characterize the spectrum features of hyperspectral origin data, pay more attention to the correlation of adjacent bands, and make the training model more accurate.

3.2. Attention with CNN for Spatial Classification

Our CNN model aims to extract robust spatial features. The attention mechanism we added on the spatial CNN focuses on the dependence between spatial neighboring pixels and the significant features on the entire input patch. Experiments in Section 4 for CNN and attention CNN show that attention weights contribute greatly to spatial classification. The attention CNN architecture is shown in Figure 5.

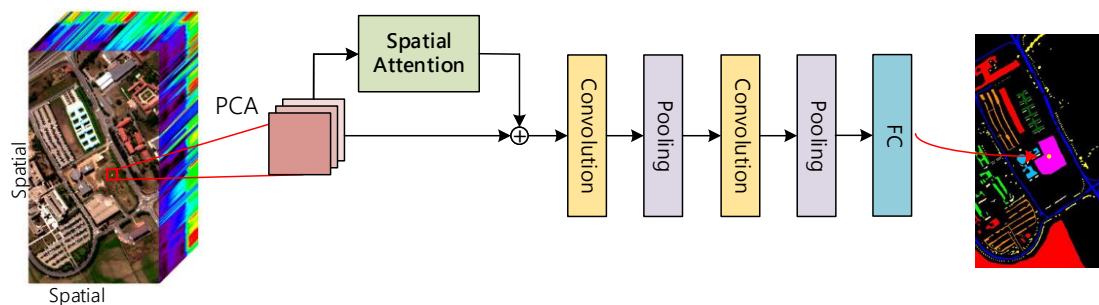


Figure 5. CNN model with attention mechanism for spatial classification. The original HSI is firstly processed by principal component analysis (PCA) for dimensionality reduction. Spatial attention map is calculated from an initial input patch for CNN and superimposed on the patch to the subsequent network.

First of all, in order to fuse the spatial information of all bands and suppress noise, we reduce the dimensions of an HSI to the low-dimensional subspace via principal component analysis (PCA). The tighter the relationship between the target pixel and its neighborhood, the smaller the patch created for the target pixel. After PCA, for instance, the first three components of the Pavia University dataset are reserved because they have almost 99.3% information. Around each pixel, we create a patch of size $k \times k \times 3$ as a neighbor region as the input of the spatial branch. With the addition of the attention mechanism, our CNN model can estimate the salience and correlation inner different image regions.

Different from the spectral attention, the spatial attention focuses on where the informative part is, which is a kind of complementary. CNN attention is added before the convolution layer. For the input neighbor region S of size $m \times n \times c$, we generate an efficient feature descriptor by utilizing the

inner-spatial relationships of features. The spatial attention is denoted by a weight matrix α with the same size $m \times n$ as the feature map, and the element α_{ij} of α bespeaks the attention weight for the pixel vector S_{ij} composed of c PCA channels located at (i, j) in the neighbor region.

Particularly, the spatial attention weight map α is calculated by two steps. The first step is to get a distributed representation through a single-layer neural network, as Equation (7). In the second step, a sigmoid function calculates α_{ij} which evaluates the impact between the i_{th} position and j_{th} position. Moreover, the more similar feature representations indicate the greater relevance of the two positions contributed. The definition of the spatial attention model is shown below:

$$s_{ij} = \tanh(W_s \cdot S_{ij} + b_s), \quad (7)$$

$$\alpha_{ij} = \sigma(W_z \cdot s_{ij} + b_z). \quad (8)$$

where $W_s \in \mathbb{R}^{k \times C}$ and $W_z \in \mathbb{R}^k$ are transformation matrices that map image visual features, $b_s \in \mathbb{R}^k$ and $b_z \in \mathbb{R}^l$ are model biases, and $\sigma(\cdot)$ is a sigmoid function which also could constrain the attention weight to lie in the interval $(0, 1)$.

For each patch, the convolutional layer uses a sliding window as a kernel to move across, and it can locate similar features in this patch by calculating the point-to-point inner product. The pooling layer selects values to reduce the feature map dimension. The kernels of the convolutional layers are 5×5 , and the strides of the max pooling layers are 2. The fully connected (FC) layer owns 1024 units. Table 1 lists the rest settings about the spatial attention network.

Table 1. Network settings in the spatial attention CNN.

Layer	Size	Activation	Strides
conv1	$32 \times 5 \times 5$	ReLU	1
max pooling	2×2	/	2
conv2	$64 \times 5 \times 5$	ReLU	1
max pooling	2×2	/	2
FC	1024	/	/

3.3. Merge

In our method, the last step concatenates the two branches to co-training them, and the complete framework is shown in Figure 6. The proposed Bi-RNN with attention network and the CNN with attention network are adopted as the spectral feature learner and the spatial feature learner, respectively. In order to exploit both spectral correlation and spatial features and extract the intergrated spectral-spatial features, we concatenate the last fully connected layer in the Bi-RNN with the one in the CNN to form a new fully connected layer, which is followed by another FC layer to represent the joint spectral-spatial features and a softmax regression layer to predict the probability distribution of each class.

Spectral RNN with attention mechanism focuses more on distinguishable essential characteristics and the inner spectral correlations, but the attentive spatial CNN supplements the neighbor information with spatial structure features and internal spatial relevance, enabling a more homogeneous classification map and a higher accuracy. The merge layer fuses and balances the spatial and spectral information, and its result has the largest diversity in class probability estimation.

Compared to the hand-crafted features, the deep joint spectral-spatial features trained in this end-to-end framework are more discriminative and robustness. The co-training network consisting of Bi-RNN and CNN, both of which have added attention mechanism, enhances the effectiveness of extracting features and promotes hyperspectral classification accuracy.

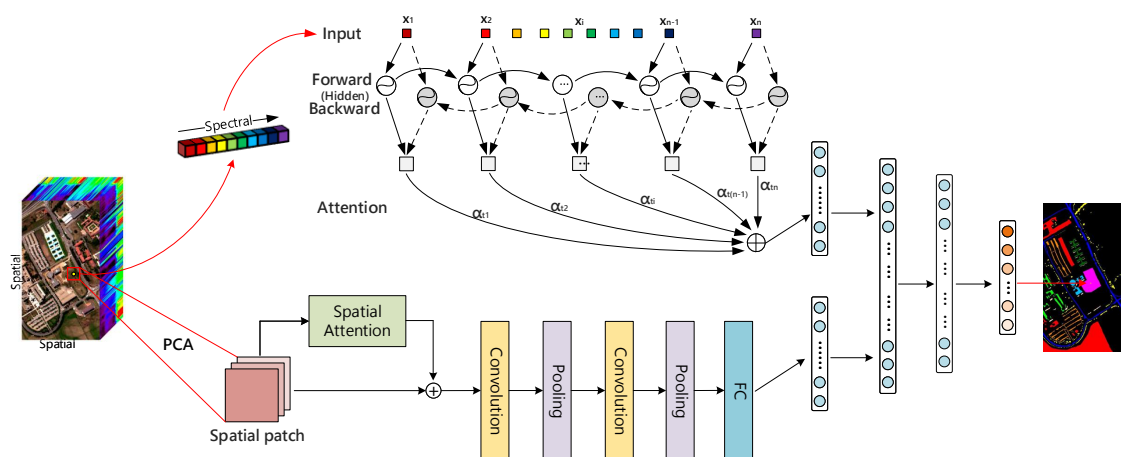


Figure 6. The whole structure of our proposed model. The spectral attention Bi-RNN branch and the spatial attention CNN branch are followed by a multi-layer merge network to extract conjoint spatial-spectral characteristics.

4. Experiment Results

In this section, we introduce three public datasets used in our experiment and the configuration of the proposed spectral-spatial attention network (SSAN). In addition, classification performance based on the proposed method and other comparative methods are presented. All the experiments are implemented with an NVIDIA RTX 2080Ti GPU, tensorflow-gpu 1.9.0 and Keras 2.1.0 with python 3.6.

4.1. Data Description

To evaluate our method, we train and test it on three public HSI classification datasets, namely, the Pavia University dataset, the Pavia Center dataset and the Indian Pines dataset, which are widely used to evaluate classification algorithms.

- **Pavia Center:** The first dataset is gained by ROSIS. We utilize 102 spectral bands after removing 13 noisy channels. The image is of 1096×715 pixels covering the center of Pavia. The available training samples contain nine urban land-cover classes.
- **Pavia University:** The second dataset is obtained by the ROSIS sensor during a flight campaign over Pavia. The ROSIS-03 sensor recorded the original image in 115 spectral channels ranging from 430 to 860nm. Removing 12 noisy bands, the left 103 bands are adopted. The spatial size of the image is 610×340 pixels. The ground truth map contains nine different urban land-cover types with more than 1000 labeled pixels for each class.
- **Indian Pines:** The third dataset is gathered by AVIRIS sensor over the Indian Pines test site in Northwestern Indiana. Removing bands that cover water absorption features, the remaining 200 bands with 145×145 pixels are used in this paper. The original data consists of observations from 16 identified classes representing the land cover types.

In our experiment, the training set is generated randomly from the ground reference data and the remaining reference samples consist of testing sets. For deep learning models, the training set consists of labeled samples and validation samples. To overcome the categories' imbalance problem, instead of splitting dataset by an average percentage of each class, we randomly select 100 labeled samples and 100 validation samples of each annotated class for training set in the Pavia Center dataset and Pavia University dataset, details are listed in Tables 2 and 3. As for the same problem in the Indian Pines dataset, some class samples of this dataset are less than 100. Table 4 provides the detail information about different classes and the corresponding training sets and testing sets.

Table 2. Number of training and testing samples in Pavia Center dataset.

Label	Class Name	Training		Testing
		Labeled	Validation	
1	Waters	100	100	65,771
2	Trees	100	100	7398
3	Asphalt	100	100	2890
4	Self-Blocking Bricks	100	100	2485
5	Bitumen	100	100	6384
6	Tiles	100	100	9048
7	Shadows	100	100	7087
8	Meadows	100	100	42,626
9	Bare soil	100	100	2663
Total		900	900	146,352

Table 3. Number of training and testing samples in Pavia University dataset.

Label	Class Name	Training		Testing
		Labeled	Validation	
1	Asphalt	100	100	6431
2	Meadows	100	100	18,449
3	Gravel	100	100	1899
4	Trees	100	100	2864
5	Painted mental sheets	100	100	1145
6	Bare Soil	100	100	4829
7	Bitumen	100	100	1130
8	Self-Blocking Bricks	100	100	3482
9	Shadows	100	100	747
Total		900	900	40,976

Table 4. Number of training and testing samples in Indian Pines dataset.

Label	Class Name	Training		Testing
		Labeled	Validation	
1	Alfalfa	8	4	34
2	Corn-notill	100	100	1228
3	Corn-mintill	100	100	630
4	Corn	50	50	137
5	Grass-pasture	50	50	383
6	Grass-trees	100	100	530
7	Grass-pasture-mowed	8	4	16
8	Hay-windowed	50	50	378
9	Oats	8	4	8
10	Soybean-notill	100	100	772
11	Soybean-mintill	100	100	2255
12	Soybean-clean	100	100	393
13	Wheat	50	50	105
14	Woods	100	100	1065
15	Buildings-Grass-Trees-Drives	100	100	186
16	Stone-Steel-Towers	20	10	63
Total		1044	922	8283

In the Pavia Center dataset, we choose four principal components that could consist of 99% information of the original data, and then extract image patches as CNN branch input. Similarly, three principal components are chosen in the Pavia Center dataset and four principle components for the Indian Pines dataset.

4.2. Parameter Setting

There are three main parameters that have significant impact on our experiment: Learning rate, spatial size and dropout. In this section, we evaluate the sensitivity of performance to different parameter settings of our proposed model in detail.

(1) Learning rate: Firstly, we test the impact of different learning rates. The learning rate controls the learning process and the amount of allocate error when updating model weights each time. At extremes, a learning rate could be too large and results in an oscillation over training epochs, or too small to be converged. The learning rate of our model is chosen from [0.0003, 0.0005, 0.0008, 0.001, 0.003, 0.005, 0.01], and the optimal learning rate based on the classification accuracy is 0.005 for the Pavia Center dataset, 0.0005 for the Pavia University dataset, and 0.0005 for the Indian Pines dataset.

(2) Spatial size: Spatial features learning from CNN badly depend on the size of the spatial neighbor region. As we have fixed the reduced channel number, we test spatial sizes [15 × 15, 19 × 19, 23 × 23, 27 × 27, 31 × 31] to capture sufficient spatial information. The results are listed in Table 5, and all of them are acquired in 10,000 training iterations with batch size 128 and the optimal learning rate of each dataset. Larger size of spatial input would supply more chance to learn more spatial features. Nevertheless, a larger size of spatial region would also bring negative effect with unnecessary information and a possibility of over-smoothing phenomenon. To make a fair comparison, we fix the spatial size of 27 × 27 in different classification methods.

Table 5. Overall accuracy (OA) of the proposed method with different spatial sizes.

Spatial Size	Overall Accuracy		
	Pavia Center	Pavia University	Indian Pines
15 × 15	97.71	94.18	92.69
19 × 19	98.24	96.57	94.70
23 × 23	98.66	98.21	96.55
27 × 27	99.25	98.87	97.23
31 × 31	98.51	96.33	97.01

(3) Dropout: During training, the neural network develops co-dependency among neurons which lead to over-fitting of training data. Dropout is a regularization approach in neural networks which helps reduce interdependent learning and preventing over-fitting. We test it with different dropout proportions. The results in Table 6 represent that 60% dropout for the Pavia University dataset and 50% dropout for the Pavia Center dataset, and that the Indian Pines dataset acquires the highest accuracy.

Table 6. OA of proposed method with different spatial sizes.

Dropout	Overall Accuracy		
	Pavia Center	Pavia University	Indian Pines
0.2	94.78	90.32	92.55
0.3	95.33	95.49	91.74
0.4	98.11	97.12	95.26
0.5	97.32	98.66	97.14
0.6	99.13	96.45	96.32

4.3. Classification Results

To demonstrate the superiority and effectiveness of the proposed SSAN model, we compare it with traditional methods such as KNN and SVM, and advanced machine-learning methods such as CNN, RNN, RNN with attention (ARNN), and CNN with attention (ACNN). The comparative methods are summarized as follows:

- (1) **KNN**: k nearest neighbors, the parameter k is set to $[3, 5, 5]$ for the Pavia Center dataset, the Pavia University dataset and the Indian Pines dataset, respectively.
- (2) **SVM**: Support vector machine with radial basis function kernel.
- (3) **RNN**: GRU-based bi-directional RNN, which is the base RNN model in our proposed SSAN. Learning rate and the training step have been optimized to fulfill a great classification accuracy.
- (4) **CNN**: Two-dimensional CNN, which has the same structure with CNN branch in SSAN. Learning rate and the spatial input size are optimized on validation samples.
- (5) **ARNN**: Our attention Bi-RNN branch in SSAN.
- (6) **ACNN**: Our attention CNN branch in SSAN.
- (7) **SSAN**: The proposed spectral-spatial attention network.

For a fair comparison, we utilize the same training and testing datasets for all methods, and all algorithms are executed twenty times. The average results which add the standard deviation obtained from the 20 runs are reported to reduce random selection effects. Overall accuracy (OA), average accuracy (AA), and the kappa coefficient k are used as the evaluation measurements for the compared methods.

4.4. Results on the Pavia Center Dataset

The classification maps of Pavia Center dataset from deep learning models and our proposed model are provided in Figure 7, and the corresponding accuracy indexes including OAs, AAs and kappa coefficients are presented in Table 7. Obviously, the performance of our proposed method is much better than other methods, and SSAN generates the highest OA, AA, kappa and the best classification results. From Table 7, comparing the OAs and AAs, we can see that most results are unbalanced, such as class Bitumen in SVM, and class Self-Blocking Bricks in RNN. Our method SSAN acquires more smooth and homogeneous results, and it proves that only using spectral or spatial information is insufficient for this task. Comparing ARNN and RNN, Self-Blocking Bricks is improved obviously from 50.46% to 76.69%, and classification accuracies of Asphalt, Tiles and Shadows in ACNN are increased respectively compared to CNN. Taking the accuracy of all classes into consideration, our method shows more robustness even with a small number of training samples and unbalance among classes.

4.5. Results on the Pavia University Dataset

Figure 8 shows the qualitative classification maps of deep learning networks and our method. Table 8 lists the index results and evaluation measurements quantitatively. It is obvious that the proposed SSAN surpasses other methods and owns the highest accuracy on most classes except class Asphalt and class Meadows, where the results of these two classes in our method have slightly lower precisions than ACNN. By adding attention mechanism, classification accuracies of Meadows and Self-Blocking Bricks in ARNN are improved significantly compared with RNN, while Trees, Bitumen and Shadows are better classified in ACNN in comparison to CNN. Viewing the classification maps, we notice that most ground objects are classified well and the house and road edges are clear. Nevertheless, a few scattered and diverse misclassifications are inside the natural vegetation area, which destroy the object integrity, especially class Bitumen and class Bare Soil. By adding attention mechanism and combining ARNN with ACNN, the results show that our SSAN model outperforms other approaches in acquiring more homogenized and favorable classification maps.

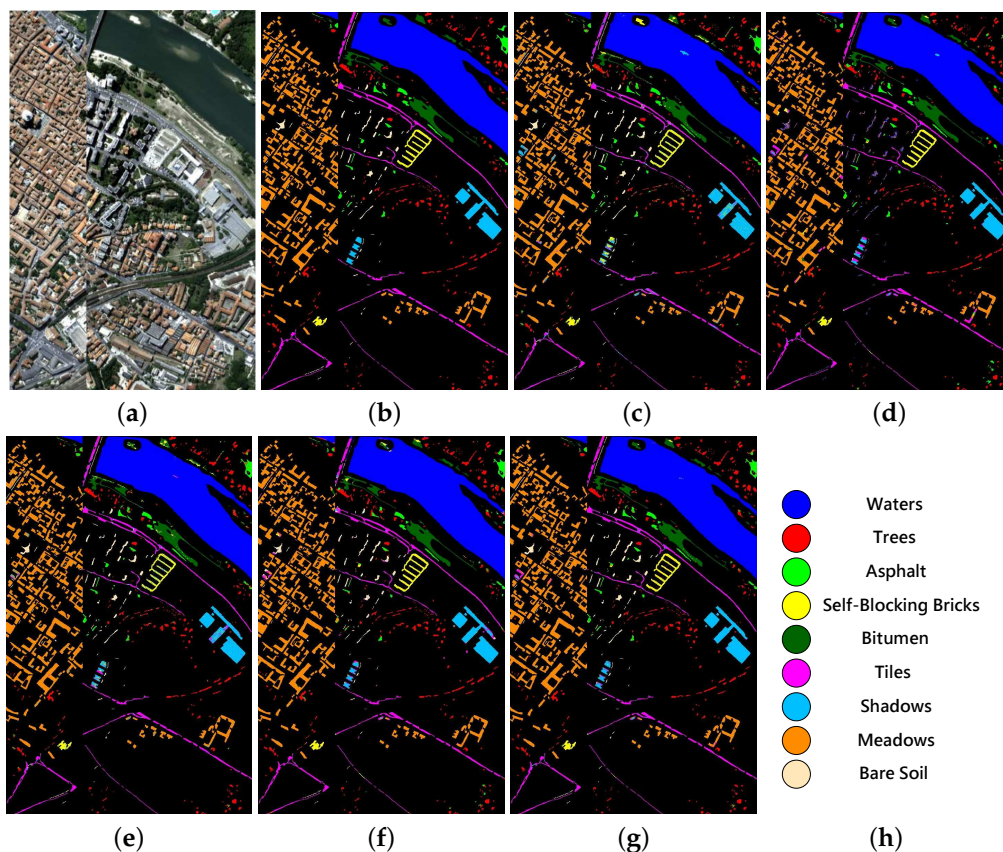


Figure 7. Classification maps using different methods on the Pavia Center dataset: (a) False-color image, (b) Ground-truth map, (c) RNN, (d) CNN, (e) RNN with attention (ARNN), (f) CNN with attention (ACNN), (g) spectral-spatial attention network (SSAN).

Table 7. Classification performance of different methods for the Pavia Center dataset. Bold indicates the best result.

Label	Class Name	KNN	SVM	RNN	CNN	ARNN	ACNN	SSAN
1	Waters	99.15	99.17	99.63	98.22	98.87	99.63	99.97
2	Trees	88.76	80.54	86.34	89.37	91.32	92.15	98.37
3	Asphalt	76.34	94.22	85.46	77.92	86.28	84.63	93.04
4	Self-Blocking Bricks	81.92	83.27	50.46	86.39	76.69	87.35	94.22
5	Bitumen	86.39	51.33	94.29	89.77	94.05	98.07	98.40
6	Tiles	91.44	93.24	91.75	79.89	95.88	87.36	97.11
7	Shadows	81.31	75.47	81.63	87.28	84.21	96.42	97.03
8	Meadows	93.67	94.55	97.18	98.27	98.75	97.91	98.22
9	Bare soil	97.18	98.16	99.31	93.87	99.13	89.22	99.63
	OA	92.66 ±0.34	93.11 ±0.67	99.04 ±0.25	99.13 ±0.19	99.24 ±0.08	99.39 ±0.11	99.69 ±0.05
	AA	88.22 ±0.42	85.74 ±0.33	89.21 ±0.16	90.26 ±0.25	92.56 ±0.11	93.72 ±0.20	98.31 ±0.17
	Kappa	90.13 ±0.21	90.03 ±0.44	97.34 ±0.23	97.37 ±0.30	97.48 ±0.24	98.33 ±0.17	99.18 ±0.09

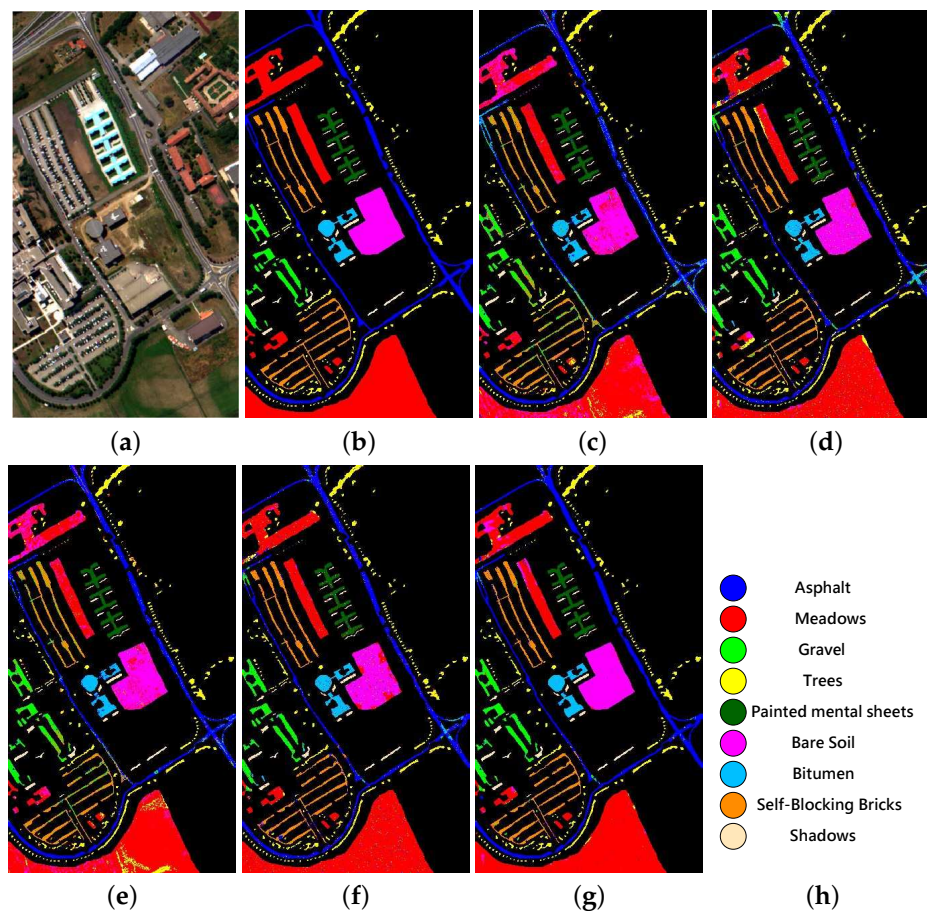


Figure 8. Classification maps using different methods on the Pavia University dataset: (a) False-color image, (b) Ground-truth map, (c) RNN, (d) CNN, (e) ARNN, (f) ACNN, (g) SSAN.

Table 8. Classification performance of different methods for the Pavia University dataset. Bold indicates the best result.

Label	Class Name	KNN	SVM	RNN	CNN	ARNN	ACNN	SSAN
1	Asphalt	76.00	78.90	80.86	92.13	88.94	98.46	98.73
2	Meadows	69.33	82.31	68.69	94.23	83.56	98.59	95.72
3	Gravel	84.00	81.41	80.23	76.83	82.99	77.36	95.27
4	Trees	95.44	93.99	95.23	74.67	94.22	94.66	96.34
5	Painted mental sheets	96.88	99.21	99.21	91.95	99.46	99.23	99.66
6	Bare Soil	71.34	83.67	82.35	86.34	86.25	87.07	98.78
7	Bitumen	95.26	92.20	86.98	72.84	90.45	97.08	96.34
8	Self-Blocking Bricks	73.00	86.56	78.22	88.86	82.29	90.39	95.26
9	Shadows	97.42	99.74	99.84	87.40	99.63	99.79	99.89
	OA	84.89 ±0.23	84.03 ±0.88	95.42 ±0.34	97.36 ±0.26	97.37 ±0.31	98.98 ±0.15	99.24 ±0.17
	AA	84.89 ±0.46	88.59 ±0.67	87.66 ±0.26	86.34 ±0.18	90.71 ±0.25	94.07 ±0.32	98.07 ±0.23
	Kappa	83.57 ±0.28	79.94 ±0.35	87.21 ±0.19	93.77 ±0.07	92.22 ±0.37	97.17 ±0.14	98.17 ±0.22

4.6. Results on the Indian Pine Dataset

The false-color images of the Indian Pine dataset and their corresponding ground-truth maps along with classification maps of the models are represented in Figure 9, and the corresponding accuracy indexes are shown in Table 9. The traditional approaches barely utilize the shallow spectral feature and neglect abundant spatial features, which leads to a fairly unimpressive classification

performance. The classification maps of other methods present many noisy points and confuse class Soybean-mintill and class Building-Grass-Trees-Drives with other classes. Attention layer in RNN effectively improves classification accuracy of almost all categories in this dataset according to the results in ARNN and RNN. Similarly, with attention mechanism, Alfalfa, Grass-pasture-mowed, Soybean-clean and Buildings-Grass-Trees-Drives are classified much better in ACNN than CNN. From these classification maps we can see that some classes are hard to be correctly classified, and it brings challenges to the effectiveness and robustness of classifier.

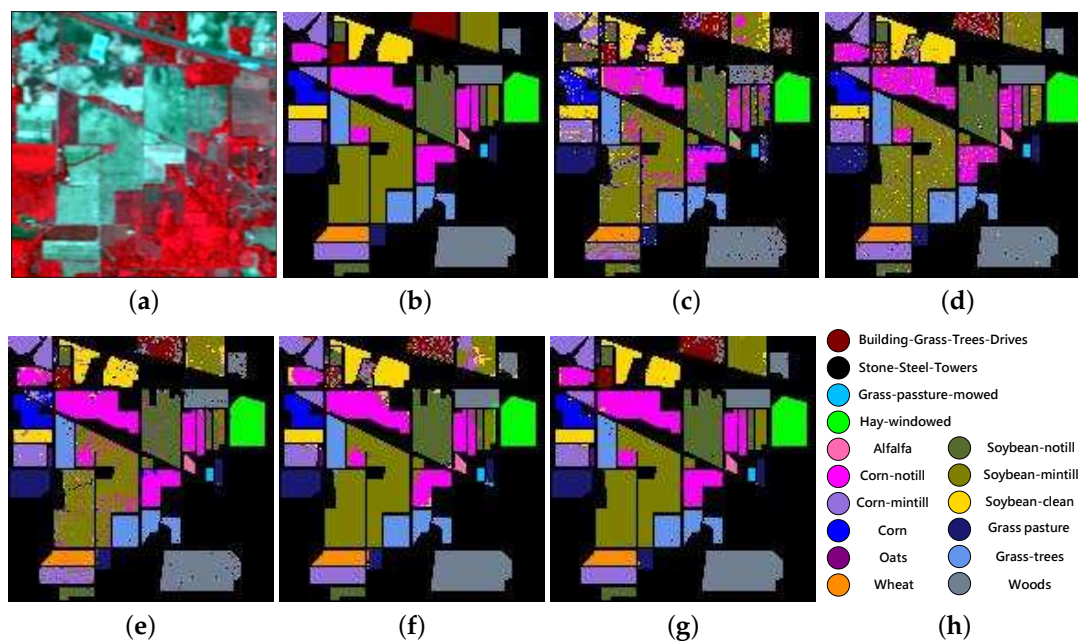


Figure 9. Classification maps using different methods on the Indian Pines dataset: (a) False-color image, (b) Ground-truth map, (c) RNN, (d) CNN, (e) ARNN, (f) ACNN, (g) SSAN.

Table 9. Classification performance of different methods for the Indian Pines dataset. Bold indicates the best result.

Label	Class Name	KNN	SVM	RNN	CNN	ARNN	ACNN	SSAN
1	Alfalfa	35.91	84.89	83.43	76.52	79.32	95.44	97.76
2	Corn-notill	47.32	71.78	77.87	87.19	85.27	92.76	98.83
3	Corn-mintill	51.68	63.07	76.21	82.91	84.23	87.83	99.25
4	Corn	68.77	89.57	55.39	88.32	67.65	87.29	99.67
5	Grass-pasture	87.32	93.86	87.98	91.68	92.14	96.82	99.24
6	Grass-trees	90.24	94.62	94.57	94.35	95.57	96.37	98.36
7	Grass-pasture-mowed	82.49	92.77	89.42	81.67	98.04	99.73	100
8	Hay-windowed	96.75	98.79	94.65	87.02	99.26	91.42	99.32
9	Oats	63.26	92.36	42.47	83.14	84.41	90.87	99.76
10	Soybean-notill	61.89	76.22	68.49	85.29	83.85	89.64	98.79
11	Soybean-mintill	54.71	62.33	86.52	94.31	94.62	95.39	99.47
12	Soybean-clean	49.37	77.56	78.61	85.14	86.33	97.66	98.65
13	Wheat	96.84	98.39	93.58	94.32	95.79	99.13	100
14	Woods	84.39	93.64	94.91	96.42	96.03	98.81	99.46
15	Buildings-Grass-Trees-Drives	47.85	69.48	74.82	76.59	89.28	92.27	99.32
16	Stone-Steel-Towers	81.09	86.78	82.68	62.33	96.82	83.15	97.53
	OA	64.52	78.03	91.31	95.24	94.87	97.27	99.67
		±0.47	±0.69	±0.39	±0.24	±0.37	±0.14	±0.21
	AA	69.03	83.92	82.05	86.78	89.92	94.32	99.08
		±0.56	±0.73	±0.28	±0.19	±0.41	±0.25	±0.16
	Kappa	62.39	75.26	87.67	93.05	92.71	95.86	98.37
		±0.82	±0.40	±0.26	±0.07	±0.29	±0.18	±0.32

Comparing ARNN with RNN and ACNN with CNN, the attention weight, which captures spatial correlations between adjacent channels and spatial inner dependency, helps a lot in focusing on strongly related features and correcting severe misclassified pixels. Our proposed SSAN enhances the accuracy of indistinguishable classes and gains a more uniform and smooth result. One possible reason for misclassification is that some indistinguishable classes may have similar features either in the spectral or in the spatial domain. Another point worth considering is that some classes in the Indian Pine dataset are too unbalanced to learn sufficient differentiable features. In order to overcome these problems, our method surpasses other approaches in acquiring more homogeneous classification maps and manifests the highest accuracy.

The results indicate that the proposed method with the attention mechanism in two branches is effective in HSI classification. Obviously, the aforementioned traditional methods, such as SVM and KNN, demonstrate poor performance. Deep learning methods, such as CNN and RNN, are effective because of their discriminative features. A comparison of RNN and ARNN or CNN and ACNN indicates that the attention mechanism plays a significant role in our method. Within the attention weights, CNN focuses more on saliency features in the spatial domain, and RNN attempts to learn spectral correlations from adjacent spectrum. Our fusion network combines spatial and spectral dimensions and exhibits well-balanced results among all compared methods in all scenarios.

5. Conclusions

In this study, a novel two-branch co-training method is proposed to extract spectral-spatial features based on ARNN and ACNN for HSI classification. Inspired by the way humans perceive images that emphasizes informative features and suppresses unnecessary information, known as attention mechanism, we incorporate this mechanism into our model. ARNN and ACNN are trade on learning characteristics from spectral and spatial information, respectively, and they can grasp numerous interspectral correlations in the continuous spectrum domain and focus on similar spatial features between neighboring pixels in spatial dimension by adding attention weights. Specifically, we use bi-directional RNN in ARNN to learn forward and backward information in spectra. The co-training network can learn higher-level spatial-spectral joint characteristics and inherit features from both ARNN and ACNN. Analysis of experimental results on three public datasets demonstrates that our method not only performs better than the other methods, but also extracts more homogeneous discriminative feature representations.

Our work has proven the effectiveness of attention mechanism in HSI classification in this paper, and we plan to generalize our method for other more complex remote sensing applications, such as unmixing and change detection, in the near future.

Author Contributions: All the authors made significant contributions to the work. X.M., E.P. and Y.M. conceived and designed the research and performed the experiments. E.P., X.D. and F.F. analyzed the results and wrote the manuscript, and J.M., J.H., Q.D. and H.Z. provided insightful advices to this work and revised the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China under Grant 61805181, Grant 61773295, Grant 61705170 and Grant 61605146.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Paoletti, M.; Haut, J.; Plaza, J.; Plaza, A. A new deep convolutional neural network for fast hyperspectral image classification. *ISPRS J. Photogramm. Remote Sens.* **2018**, *145*, 120–147. [[CrossRef](#)]
2. Zhang, L.; Zhang, L.; Du, B. Deep learning for remote sensing data: A technical tutorial on the state of the art. *IEEE Geosci. Remote Sens. Mag.* **2016**, *4*, 22–40. [[CrossRef](#)]
3. Ma, L.; Crawford, M.M.; Zhu, L.; Liu, Y. Centroid and Covariance Alignment-Based Domain Adaptation for Unsupervised Classification of Remote Sensing Images. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 2305–2323. [[CrossRef](#)]

4. Tao, C.; Wang, Y.; Cui, W.; Zou, B.; Zou, Z.; Tu, Y. A transferable spectroscopic diagnosis model for predicting arsenic contamination in soil. *Sci. Total. Environ.* **2019**, *669*, 964–972. [[CrossRef](#)]
5. Fan, F.; Ma, Y.; Li, C.; Mei, X.; Huang, J.; Ma, J. Hyperspectral image denoising with superpixel segmentation and low-rank representation. *Inf. Sci.* **2017**, *397*, 48–68. [[CrossRef](#)]
6. Ma, J.; Ma, Y.; Li, C. Infrared and visible image fusion methods and applications: A survey. *Inf. Fusion* **2019**, *45*, 153–178. [[CrossRef](#)]
7. Chen, Y.; Lin, Z.; Zhao, X.; Wang, G.; Gu, Y. Deep learning-based classification of hyperspectral data. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 2094–2107. [[CrossRef](#)]
8. Jiang, J.; Ma, J.; Chen, C.; Wang, Z.; Cai, Z.; Wang, L. SuperPCA: A Superpixelwise PCA Approach for Unsupervised Feature Extraction of Hyperspectral Imagery. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 4581–4593. [[CrossRef](#)]
9. Roscher, R.; Waske, B.; Forstner, W. Incremental import vector machines for classifying hyperspectral data. *IEEE Trans. Geosci. Remote Sens.* **2012**, *50*, 3463–3473. [[CrossRef](#)]
10. Li, J.; Marpu, P.R.; Plaza, A.; Bioucas-Dias, J.M.; Benediktsson, J.A. Generalized composite kernel framework for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2013**, *51*, 4816–4829. [[CrossRef](#)]
11. Jiang, J.; Ma, J.; Wang, Z.; Chen, C.; Liu, X. Hyperspectral Image Classification in the Presence of Noisy Labels. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 851–865. [[CrossRef](#)]
12. Hu, W.; Huang, Y.; Wei, L.; Zhang, F.; Li, H. Deep convolutional neural networks for hyperspectral image classification. *J. Sens.* **2015**, *2015*. [[CrossRef](#)]
13. Mou, L.; Ghamisi, P.; Zhu, X.X. Deep recurrent neural networks for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 3639–3655. [[CrossRef](#)]
14. Wu, H.; Prasad, S. Convolutional recurrent neural networks for hyperspectral data classification. *Remote Sens.* **2017**, *9*, 298. [[CrossRef](#)]
15. 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](#)]
16. 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](#)]
17. Cao, X.; Zhou, F.; Xu, L.; Meng, D.; Xu, Z.; Paisley, J. Hyperspectral image classification with Markov random fields and a convolutional neural network. *IEEE Trans. Image Process.* **2018**, *27*, 2354–2367. [[CrossRef](#)]
18. Xu, Y.; Zhang, L.; Du, B.; Zhang, F. Spectral-Spatial Unified Networks for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 5893–5909. [[CrossRef](#)]
19. Chen, L.C.; Yang, Y.; Wang, J.; Xu, W.; Yuille, A.L. Attention to scale: Scale-aware semantic image segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 3640–3649.
20. Xu, R.; Tao, Y.; Lu, Z.; Zhong, Y. Attention-Mechanism-Containing Neural Networks for High-Resolution Remote Sensing Image Classification. *Remote Sens.* **2018**, *10*, 1602. [[CrossRef](#)]
21. Chen, L.; Zhang, H.; Xiao, J.; Nie, L.; Shao, J.; Liu, W.; Chua, T.S. Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6298–6306.
22. Nam, H.; Ha, J.W.; Kim, J. Dual attention networks for multimodal reasoning and matching. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 299–307.
23. Ma, J.; Zhao, J.; Jiang, J.; Zhou, H.; Guo, X. Locality preserving matching. *Int. J. Comput. Vis.* **2019**, *127*, 512–531. [[CrossRef](#)]
24. Yan, J.; Li, C.; Li, Y.; Cao, G. Adaptive discrete hypergraph matching. *IEEE Trans. Cybern.* **2018**, *48*, 765–779. [[CrossRef](#)]
25. Ma, J.; Jiang, J.; Zhou, H.; Zhao, J.; Guo, X. Guided locality preserving feature matching for remote sensing image registration. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 4435–4447. [[CrossRef](#)]
26. Kuen, J.; Wang, Z.; Gang, W. Recurrent Attentional Networks for Saliency Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2016; pp. 3668–3677.

27. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
28. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv* **2014**, arXiv:1409.0473.
29. Lin, Z.; Feng, M.; Santos, C.N.d.; Yu, M.; Xiang, B.; Zhou, B.; Bengio, Y. A structured self-attentive sentence embedding. *arXiv* **2017**, arXiv:1703.03130.
30. Pei, W.; Dibeklioğlu, H.; Baltrušaitis, T.; Tax, D.M. Attended End-to-end Architecture for Age Estimation from Facial Expression Videos. *arXiv* **2017**, arXiv:1711.08690.
31. Graves, A.; Mohamed, A.R.; Hinton, G. Speech recognition with deep recurrent neural networks. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, DC, Canada, 26–31 May 2013; pp. 6645–6649.
32. Sundermeyer, M.; Ney, H.; Schluter, R. From Feedforward to Recurrent LSTM Neural Networks for Language Modeling. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2015**, *23*, 517–529. [[CrossRef](#)]
33. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)]
34. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv* **2014**, arXiv:1406.1078 .
35. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1097–1105.
36. Wang, Z.; Yi, P.; Jiang, K.; Jiang, J.; Han, Z.; Lu, T.; Ma, J. Multi-memory convolutional neural network for video super-resolution. *IEEE Trans. Image Process.* **2019**, *28*, 2530–2544. [[CrossRef](#)]
37. Ma, J.; Yu, W.; Liang, P.; Li, C.; Jiang, J. FusionGAN: A generative adversarial network for infrared and visible image fusion. *Inf. Fusion* **2019**, *48*, 11–26. [[CrossRef](#)]
38. Tian, T.; Li, C.; Xu, J.; Ma, J. Urban area detection in very high resolution remote sensing images using deep convolutional neural networks. *Sensors* **2018**, *18*, 904. [[CrossRef](#)]
39. Li, Y.; Zhang, Y.; Huang, X.; Ma, J. Learning source-invariant deep hashing convolutional neural networks for cross-source remote sensing image retrieval. *IEEE Trans. Geosci. Remote. Sens.* **2018**, 1–16. [[CrossRef](#)]
40. Ma, J.; Zhao, J. Robust topological navigation via convolutional neural network feature and sharpness measure. *IEEE Access* **2017**, *5*, 20707–20715. [[CrossRef](#)]
41. Jaderberg, M.; Simonyan, K.; Zisserman, A.; Kavukcuoglu, K. Spatial transformer networks. In Proceedings of the Neural Information Processing Systems 2015, Montreal, QC, Canada, 7–12 December 2015; pp. 2017–2025.
42. Kim, J.H.; Lee, S.W.; Kwak, D.H.; Heo, M.O.; Kim, J.; Ha, J.W.; Zhang, B.T. Multimodal Residual Learning for Visual QA. In Proceedings of the 30th International Conference on Neural Information Processing Systems (NIPS'16), Barcelona, Spain, 5–10 December 2016; pp. 361–369.
43. Yang, Y.; Zhong, Z.; Shen, T.; Lin, Z. Convolutional Neural Networks with Alternately Updated Clique. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 2413–2422.

