remote sensing

MDPI

Article

Alternately Updated Spectral–Spatial Convolution Network for the Classification of Hyperspectral Images

Wenju Wang [1], **Shuguang Dou** [1,*] and **Sen Wang** [2]

[1] College of Communication and Art Design, University of Shanghai for Science and Technology, Shanghai 200093, China
[2] Institute of Information Technology, Shanghai Baosight Software Co., Ltd., Shanghai 200940, China
* Correspondence: 173732497@st.usst.edu.cn; Tel.: +86-138-1834-7625

Received: 10 July 2019; Accepted: 24 July 2019; Published: 26 July 2019

Abstract: The connection structure in the convolutional layers of most deep learning-based algorithms used for the classification of hyperspectral images (HSIs) has typically been in the forward direction. In this study, an end-to-end alternately updated spectral–spatial convolutional network (AUSSC) with a recurrent feedback structure is used to learn refined spectral and spatial features for HSI classification. The proposed AUSSC includes alternating updated blocks in which each layer serves as both an input and an output for the other layers. The AUSSC can refine spectral and spatial features many times under fixed parameters. A center loss function is introduced as an auxiliary objective function to improve the discrimination of features acquired by the model. Additionally, the AUSSC utilizes smaller convolutional kernels than other convolutional neural network (CNN)-based methods to reduce the number of parameters and alleviate overfitting. The proposed method was implemented on four HSI data sets, as follows: Indian Pines, Kennedy Space Center, Salinas Scene, and Houston. Experimental results demonstrated that the proposed AUSSC outperformed the HSI classification accuracy obtained by state-of-the-art deep learning-based methods with a small number of training samples.

Keywords: convolutional neural network (CNN); deep learning; hyperspectral image (HSI) classification

1. Introduction

Hyperspectral images (HSIs) contain both spectral and spatial information and generally consist of hundreds of spectral bands for the same observed scene [1]. Due to the vast amounts of information they contain, HSIs have found important applications in a variety of fields, such as the non-contact analysis of food materials [2], the detection and identification of plant diseases [3], multispectral change detection [4], and medicine [5]. HSI classification is the core technology in these applications. However, since HSIs include inherently high-dimensional structures, their classification remains a challenging task in the remote sensing community.

Traditional classification methods involve feature engineering using a classifier. This process aims to extract or select features from original HSI data, typically producing a classifier based on low-dimensional features. Support vector machines (SVMs) are the most commonly used method in the early stages of HSI classification, due to their low sensitivity to high dimensionality [6]. Spectral–spatial classification methods have become predominant in recent years [7]. Mathematical-morphology-based techniques [8], Markov random fields (MRFs) [9], and sparse representations [10] are also commonly used branches. However, many of these techniques suffer from low classification accuracy due to shallow feature extraction.

Deep learning, a popular tool in multiple areas including remote sensing, has recently been applied to HSI classification [11]. Traditional feature extraction methods have struggled to identify high-level features in HSIs. However, deep learning frameworks have been proposed, in which stacked auto-encoders (SAEs) were used to obtain useful deep features [11]. Deep learning-based methods can extract deep spectral and spatial features from HSIs to obtain higher classification accuracies than those of most traditional methods [12]. Consequently, in recent years, a variety of deep learning-based methods have been used for classification [7]. For example, one study used a deep belief network (DBN) that combined PCA with logistic regression to perform HSI classification, achieving competitive classification accuracy [13].

Among these methods, deep convolutional neural network (CNN) algorithms have achieved particularly high accuracy. Deep supervised methods using randomized PCA have also been proposed to reduce the dimensionality of raw HSIs. Additionally, two-dimensional (2D) CNNs have been used to encode spectral data, spatial information, and a multilayer perceptron (MLP) for classification tasks [14]. Three-dimensional (3D) CNNs have also been used as feature extraction models to acquire spectral–spatial features from HSIs [15]. Two-layer 3D CNNs have performed far better than 2D CNN-based methods [16].

Recently, two deep convolutional spectral–spatial networks, the spectral–spatial residual network (SSRN) [17] and the fast and dense spectral–spatial convolutional network (FDSSC) [18], achieved unprecedented classification accuracy. This was due in part to the inclusion of deeper 3D CNN architectures. SSRN and FDSSC achieved an overall accuracy of above 99% across three widely used HSI data sets. As such, there appears to be little room for improvement in HSI classification. However, deep supervised methods require large quantities of data. For example, SAE logistic regression (SAE-LR) requires 60% of a data set to be labeled [11] and DBNs [13] and 3D CNNs [16] require 50% to be labeled. In contrast, SSRN and FDSSC require only 20% and 10% of a data set to be labeled, respectively. However, even a minimal labeling requirement (e.g., 10%) typically includes more than a thousand samples. As a result, the cost of sample labeling remains high in remote sensing studies.

In contrast, semi-supervised methods require only limited labeled samples. Recently, a semi-supervised model was introduced that labels samples based on local, global, and self-decisions. As a result, test samples were labeled based on multiple decisions [19]. Generative adversarial networks (GANs) can also be used for HSI classification. Real labeled HSIs and fake data generated by a generative network can be used as inputs to a discriminative network. Trained discriminative networks can then classify unlabeled samples [20]. Although GANs require only 200 real labeled samples to train, their classification accuracy remains relatively low.

Attention mechanisms [21], a popular research topic in network structures, have also proven to be effective for image classification [22]. These mechanisms mimic the internal processes of biological systems by aligning internal experiences with objective sensations, thereby increasing the observational fineness of subregions. When humans view a digital image, they do not observe every pixel in the image simultaneously. Most viewers focus on specific regions according to their requirements. Additionally, while viewing, their attentional focus is influenced by previously observed images. Attention mechanisms implemented through feedback connections [23] in a network structure can enable the network to re-weight target information and ignore background information and noise. Cross-entropy loss is the most commonly used loss function in multi-objective classification tasks and has achieved excellent performance. It increases the inter-class distance, yet neglects the intra-class distance. However, sometimes the intra-class distance is even greater than the inter-class distance, which reduces the discrimination of the extracted features. The objective function must ensure that these extracted features are distinguishable. Furthermore, the center loss function [24], which is designed to reduce the intra-class distance, has been shown to help the network extract more discriminant features. However, to prevent the degradation of classification accuracy, center loss can only be used as an auxiliary loss function.

This study introduces an attention mechanism and a center loss function for HSI classification. Inspired by previous studies [25], we propose a deep supervised method with an end-to-end alternately updated convolutional spectral–spatial network (AUSSC). Unlike 3D CNN, SSRN, and FDSSC, which include only forward connections in the convolutional layers, the AUSSC includes both forward and feedback connections. Additionally, the convolutional kernels of the AUSSC are smaller than those of 3D CNN, SSRN, or FDSSC, as the kernels are decomposed into smaller kernels. Deeper spectral and spatial features can be obtained in the AUSSC using a fixed number of parameters, due to the alternate updating of blocks.

Due to the inclusion of attention mechanisms and factorization into smaller convolutions, the AUSSC is more capable of spectral and spatial feature learning than other CNN-based methods. Both forward and feedback connections are densely connected within the alternately updated blocks. Consequently, spectral and spatial features are optimally learned and feature maps from different blocks are repeatedly refined by attention. The classification results obtained using the proposed method demonstrate that this AUSSC has been optimized for classification with a limited number of training samples. The four principal contributions of this study are as follows:

(1) The proposed method includes a recurrent feedback spectral–spatial structure with fixed parameters, in order to learn not only deep but also refined spectral and spatial features to improve HSI classification accuracy.

(2) The effectiveness of the center loss function is validated as an auxiliary loss function used to improve the results of hyperspectral image classification.

(3) The AUSSC decomposes a large 3D convolutional kernel into three smaller 1D convolutional kernels, thereby saving a large number of parameters and reducing overfitting.

(4) The AUSSC achieves state-of-the-art classification accuracy across four widely used HSI data sets, using limited training data with a fixed spatial size.

The remainder of this paper is organized as follows. Section 2 presents the framework of the proposed AUSSC. Section 3 describes the experimental data sets. The details of the experimental results and a discussion are given in Section 4. Conclusions and suggestions for future work are presented in Section 5.

2. Methods

In this section, an alternately updated spectral–spatial convolutional network is proposed for HSI classification. Figure 1 shows an overview of the proposed method. For HSI data with L channels and a size of $H \times W$, a spatial size of $s \times s$ was selected from the raw HSI data and used as the input to the AUSSC network. First, the AUSSC uses three smaller convolutional kernels to learn spectral and spatial features from an original HSI patch. Second, the alternately updated spectral and spatial blocks refine the deep spectral and spatial features using recurrent feedback. Finally, the model parameters are optimized using the cross-entropy loss and center-loss loss functions. Details of each stage are elaborated in the following subsections.

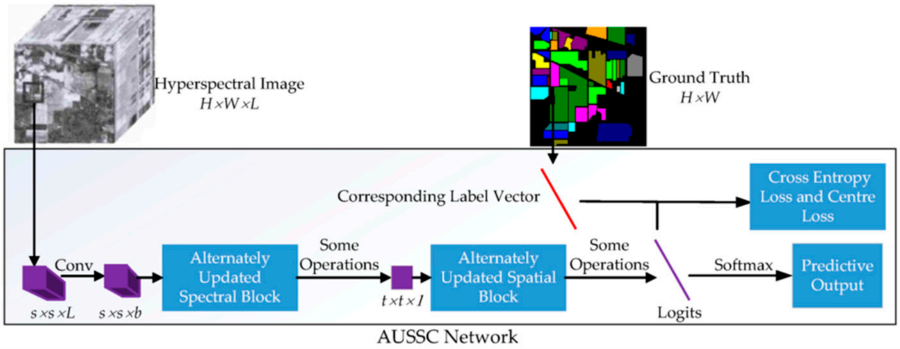

Figure 1. An overview of the proposed end-to-end alternately updated spectral–spatial convolutional network (AUSSC). "Conv" refers to the convolution operation. The operations denoted by "Some operations" are presented in detail in Section 2.4. "Logits" refers to the output of the last fully connected layer. Classification results are acquired after the Softmax operation.

2.1. *Learning Spectral and Spatial Features with Smaller Convolutional Kernels*

During HSI classification, deep CNN-based methods typically utilize preprocessing technology such as PCA. This is often followed by several convolutional layers with multiple activation functions and a classifier for obtaining classification maps. The convolution and activation can be formulated as

$$X_i^{l+1} = f(\sum_{j=1}^{N} X_j^l * k_{ji}^{l+1} + b_i^{l+1}), \tag{1}$$

where X_j^l is the ith input feature map for the $(l+1)$th layer, N is the number of feature maps in the $(l+1)$th layer, * is the convolution operation, $f(\cdot)$ is an activation function, and k_{ji}^{l+1} and b_i^{l+1} are learnable parameters that can be fine-tuned using the back-propagation (BP) algorithm.

The 3D CNN, SSRN, and FDSSC algorithms all demonstrate that an end-to-end 3D-CNN-based framework outperforms 2D-CNN-based methods that include preprocessing or post-processing, as well as other deep learning-based methods. One reason for this is that an end-to-end framework can reduce pre-processing and subsequent post-processing, allowing the connection between the original input and the final output to be as close as possible. The model then includes more space that can be adjusted automatically by the data, thereby increasing the degree of fitness. Additionally, when applied to HSIs with a 3D structure, 1D convolution operations focus on spectral features. 2D convolution operations focus on spatial features and 3D convolution operations can learn both spatial and spectral features. However, 3D kernel parameters are larger than 2D or 1D kernel parameters when the number of convolutional layers and kernels is the same. As such, a large number of model parameters can lead to overfitting.

As such, we propose an end-to-end CNN-based framework that uses smaller convolutional kernels compared to other CNN-based methods. As shown in Figure 2, the AUSSC utilizes kernels for HSI classification, ignoring other specific architectures. The 3D CNN method uses two similar convolutional kernels with sizes of $a \times a \times m_1$ and $a \times a \times m_2$, with the two convolutional kernels differing only in spectral dimension. SSRN uses a spectral kernel with a size of $1 \times 1 \times m$ and a spatial kernel with a size of $a \times a \times d$ to learn spectral and spatial representations, respectively. Convolutional kernels dictate model parameters and determine which features are learned by the CNN. In contrast, we introduce the idea of factorization into smaller convolutions from InceptionV3 [26]. In this process, a larger 3D convolutional kernel with a size of $a \times a \times m$ was divided into three smaller convolutional kernels with sizes of $1 \times 1 \times m$, $1 \times a \times 1$, and $a \times 1 \times 1$. This substantially reduced the number of parameters, accelerated the operation, and reduced the possibility of overfitting. As shown in Table 1, in the absence of bias (with all other conditions remaining the same), the convolutional kernel with a size of $a \times a \times m$ included a^2m parameters. The smallest convolutional kernel only included parameters, which is more economical than the other two. This increased the nonlinear representation capabilities of the model due to the use of multiple nonlinear activation functions.

Table 1. Parameters for different convolutional kernels.

Convolutional Kernels			**Parameters**
$a \times a \times m$			$a^2 m$
$1 \times 1 \times m$		$a \times a \times 1$	$a^2 + m$
$1 \times 1 \times m$	$a \times 1 \times 1$	$1 \times a \times 1$	$a + a + m$

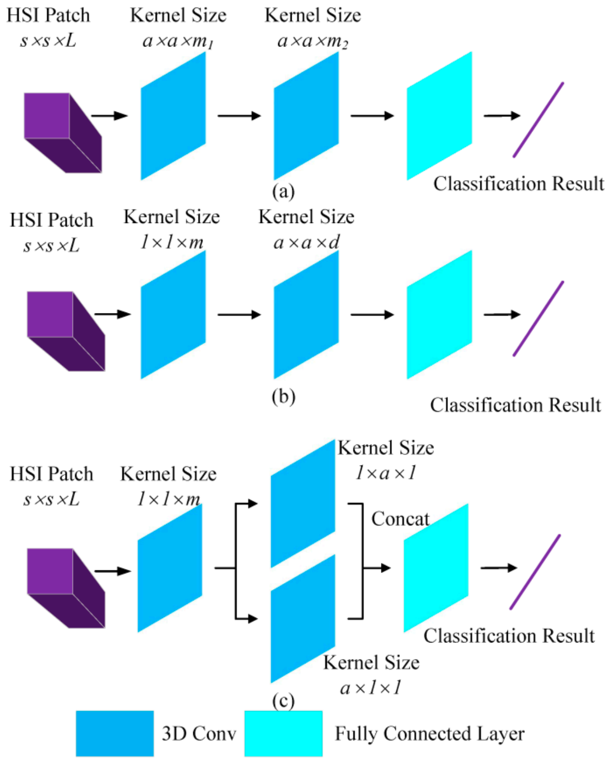

Figure 2. The structure of convolutional kernels in the 3D CNN-based method without a specific architecture. (**a**) 3D-CNN; (**b**) SSRN; (**c**) The proposed method. "Concat" refers to the concatenate operation.

2.2. *Refining Spectral and Spatial Features via Alternately Updated Blocks*

Deep CNN architectures have been used for HSI classification and have produced competitive classification results [17]. However, the connection structure in the convolutional layers is typically in the forward direction. Additionally, the convolutional kernels in SSRN and FDSSC increase with depth. Alternately updated cliques have a recurrent feedback structure and go deeper into the convolutional layers with a fixed number of parameters [25]. Therefore, we propose combining small convolutional kernels with this loop structure and design two alternately updated blocks to learn refined spectral and spatial features separately from HSIs.

As shown in Figure 3, there are two stages in the alternately updated spectral blocks. In the initialization stage (stage 1), the 3D convolutional layers use k kernels with sizes of $1 \times 1 \times m$ to learn deep spectral features. In stage 2, the 3D convolutional layers use k kernels with sizes of $1 \times 1 \times m$ to learn refined spectral features. A feature map with a size of $s \times s \times b$ and a number, n, was input to the alternately updated spectral block. This input is denoted as $X_0^{(1)}$, where the subscript 0 represents the feature map in the initial position of the alternately updated spectral block. The superscript (1) indicates the feature map is in the first stage of the alternately updated process. In stage 1, the input of every convolutional layer is the output of all the previous convolutional layers. Stage 1 can be formulated as follows:

$$X_l^{(1)} = f(\sum_{j<l} X_j^{(1)} * W_{jl}), \tag{2}$$

where $X_l^{(1)}$ is the output of the lth $(l \geq 1)$ convolutional layer in stage 1 of an alternately updated spectral block, $f(\geq)$ is a nonlinear activation function, $*$ is the convolutional operation using the padding method, and WW_{jl} is a parameter reused in stage 2.

In the looping stage (stage 2), each convolutional layer (except the input convolutional layer) is alternately updated to refine features. Stage 2 has a recurrent feedback structure, meaning that the feature map can be refined several times using the same weights. Therefore, any two convolutional layers in the alternately updated spectral block are connected bi-directionally. Stage 2 can then be formulated as follows:

$$X_l^r = f(\sum_{j<l} X_j^r * W_{jl} + \sum_{k>l} X_k^{(r-1)} * W_{kl}), \tag{3}$$

where $r \geq 2$ since the feature map is in stage 2 and can be updated multiple times by the recurrent feedback structure. Similarly, $l \geq 1$ since the input feature map is not updated.

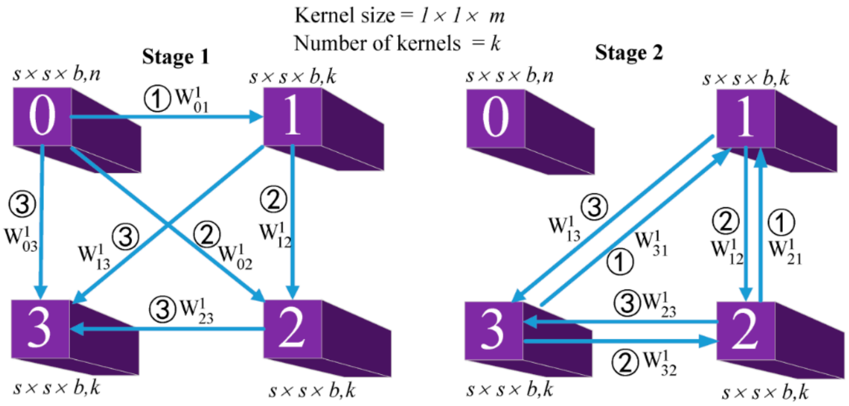

Figure 3. Two stages of alternately updated spectral blocks with three convolutional layers.

After learning refined deep spectral features, the input convolutional layer and the updated convolutional layer are concatenated in the alternately updated spectral block and transferred to the next block. Once spectral information from the HSI has been learned, the high dimensions of the feature map can be reduced by valid convolution and reshaping operations (see figure in Section 2.4.). The resulting input to the alternately updated spatial block is a feature map with number, n, and size $t \times t \times 1$.

As shown in Figure 4, there are two different convolutional kernels in the alternately updated spatial block. The 3D convolutional layers use $ka \times 1 \times 1$ and $k1 \times a \times 1$ convolutional kernels to learn deep refined spatial features with an alternately updated structure that is also used for the alternately updated spectral block. In the spatial block, two different convolutional kernels learn spatial features in parallel rather than in series. The convolutional relationship between the spatial block is the same as for the previous block.

These alternately updated blocks achieve spectral and spatial attention due to the presence of refined features obtained in the looping stages. Densely connected forward and feedback structures allow the spectral and spatial information to flow in convolutional layers within the blocks. These alternately updated blocks also include weight sharing. In stage 1, the weights increase linearly as the number of convolutional layers increases. However, in stage 2, the weights are fixed since they are shared. The partial weights from stage 1, such as W_{12}, W_{13}, and W_{23} (see Figure 2), are reused in stage 2. As features are cycled repeatedly in stage 2, the number of parameters remains unchanged.

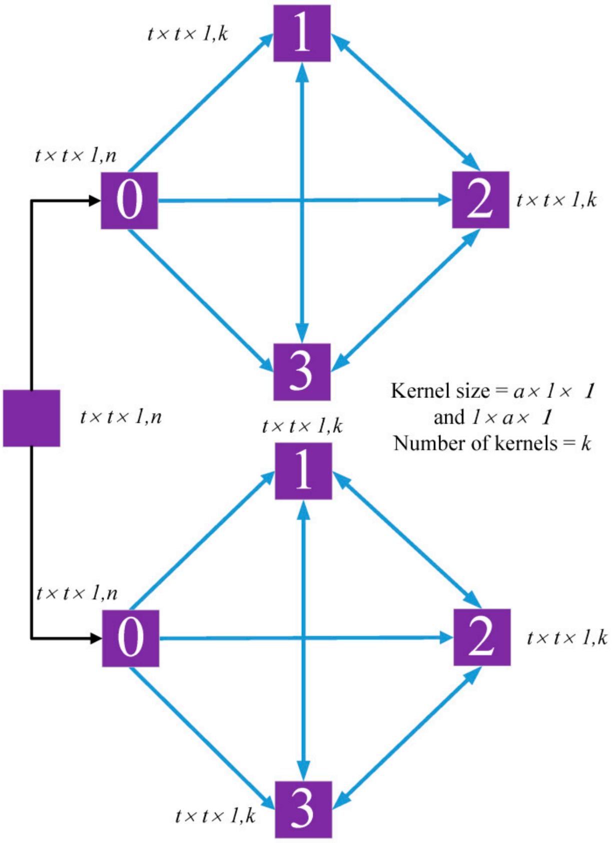

Figure 4. A representation of the alternately updated spatial block with two different convolutional kernels and three convolutional layers for each convolutional kernel.

2.3. Optimization by the Cross-Entropy Loss and Center Loss Functions

HSI classification is inherently a multi-classification task and cross-entropy loss with a softmax layer is a well-known objective function that is used for such problems. The softmax cross-entropy loss can be written in the following form:

$$\mathcal{L}_{softmax} = -\sum_{i=1}^{m} \log \frac{\exp(W_{y_i}^T x_i + b_{y_i})}{\sum\limits_{j=1}^{n} \exp(W_j^T x_i + b_j)}, \tag{4}$$

where m is the size of the mini-batch, n is the number of classes, x_i is the ith deep feature belonging to the y_ith class, W_j is the jth column of the weights W in the last fully connected layer, and b is the bias. The last layer of the CNN-based model is typically fully connected, as it is difficult to make the dimensions of the last layer equal to the number of categories without a fully connected layer. Intuitively, one would expect that learning more discriminatory features would improve the generalization performance. As such, we introduce an auxiliary loss function [24] to improve the discrimination of features acquired by the model. This function can be formulated as follows:

$$\mathcal{L}_{center} = \frac{1}{2}\sum_{i=1}^{m} \left\| x_i - c_{y_i} \right\|_2^2, \tag{5}$$

where c_{y_i} is the central feature in the y_ith class. The function decreases the quadratic sum of the distance from the center of the feature to the features of each sample in one batch, which decreases the intra-class distance. The center of feature c_{y_i} can then be updated through iterative training.

When two loss functions are used together for HSI classification, the softmax cross-entropy loss is considered to be responsible for increasing the inter-class distance. The center loss is then responsible for reducing the intra-class distance, thus increasing the discriminant degree and generalization abilities of learned features. Consequently, the objective function for the AUSSC can be written in the following form:

$$\begin{aligned}\mathcal{L} &= \mathcal{L}_{\text{softmax}} + \lambda\mathcal{L}_{center} \\ &= -\sum_{i=1}^{m} \log \frac{\exp(W_{y_i}^{T} x_i + b_{y_i})}{\sum_{j=1}^{n} \exp(W_j^{T} x_i + b_j)} \\ &\quad + \frac{\lambda}{2}\sum_{i=1}^{m} \left\| x_i - c_{y_i} \right\|_2^2\end{aligned} \tag{6}$$

where λ ∈ [0, 1) controls the proportion of center loss and the value of λ is determined experimentally, as discussed in the following section. In summary, the cross-entropy loss is the principal objective function and the inter-class distance is the principal component. The center loss is the auxiliary used to reduce the intra-class distance.

2.4. Alternatively Updated Spectral–Spatial Convolutional Network

A flowchart is included below to explain the steps in the AUSSC end-to-end network. Considering the cost and time requirements of the collection of HSI labeled samples, we propose a 3D CNN-based framework that maximizes the flow and circulation of spectral and spatial information. Figure 5 shows a $9 \times 9 \times L$ cube, which is used as input in our technique, where L is the number of HSI bands. Due to high computational costs, two convolutional layers were used in the alternately updated blocks and a single loop was used in stage 2.

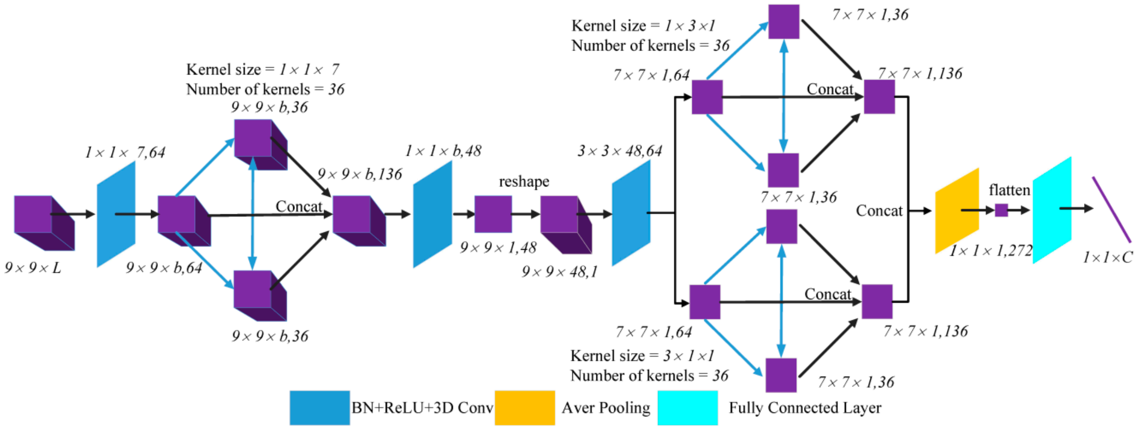

Figure 5. The AUSSC proposed for hyperspectral image (HSI) classification of labeled pixels with an input size of $9 \times 9 \times L$. The value L is the number of HSI bands and C is the number of classes.

L2 loss and batch normalization (BN) [27] were used to improve the normalization of our model. In a broad sense, L2 and other regularization parameter terms added to the loss function in machine learning are essentially weighted norms. The goal of normalization with L2 loss is to effectively reduce the size of the original parameter values in the model, with BN performing normalization operations on input neuron values. The normalization target regularizes its input value to a normal distribution with a mean value of zero and a variance of one. The blue layers and blue lines both refer to the BN, rectified linear units (ReLU), and the convolution operation. The first convolutional layer lacks both a BN and a ReLU.

The original HSI input, which has a size of $9 \times 9 \times L$, flows to the first convolutional layer with a kernel size of $(1, 1, 7)$ and a stride of $(1, 1, 2)$ to generate feature maps with a size of 6 $49 \times 9 \times b$The number of kernels in the convolutional layers of alternately updated spectral block was 36, the kernel size was $(1, 1, 7)$, and the convolutional padding method was the same. As a result, the output size for each layer remained 36 $9 \times 9 \times b$, which was unchanged in stage 1 and stage 2. After concatenating the input and updated feature maps, the output of the alternately updated spectral blocks had a size of 136 $9 \times 9 \times b$.

A valid convolutional layer with 48 channels and a kernel size of $1 \times 1 \times b$ was included between alternately updated spectral and spatial blocks. This reduced the dimensions of the output of alternately updated spectral blocks, resulting in 48 feature maps with a size of $9 \times 9 \times 1$. After reshaping the third dimension and the channel dimension, 48 channels with a size of $9 \times 9 \times 1$ were merged into a single $9 \times 9 \times 48$ channel. A valid convolutional layer with a kernel size of $3 \times 3 \times 48$ and 64 kernels transformed the feature map into 64 channels with a size of $7 \times 7 \times 1$.

Similar to the alternately updated spectral blocks, the alternately updated spatial block featured two convolutional kernels with sizes of $1 \times 3 \times 1$ and $3 \times 1 \times 1$In stage 1 and stage 2, the output of each layer was 36 kernels with a size of $7 \times 7 \times 1$. The results of two convolutional kernels were concatenated into 272 kernels with a size of $7 \times 7 \times 1$. Finally, the output passed through a 3D average pooling layer with a pooling size of $7 \times 7 \times 1$, which was converted into 272 feature maps with a size of $1 \times 1 \times 1$. After the flattening operation, a vector with a size of $1 \times 1 \times C$ was produced by the fully connected layer, where C is the number of classes. Trainable AUSSC parameters were optimized by iterative training using Equation (6) and used to compute the loss between the predicted and real values.

The following sections provide a summary of the advantages of this proposed AUSSC architecture. First, the use of three different small convolutional kernels reduced both the number of parameters and overfitting, thereby increasing the nonlinear representation ability of the model and the diversity of features. Compared with symmetric splitting into several identical small convolutional kernels, this asymmetric splitting can handle more and richer features. Second, refined deep features learned by both forward and feedback connections between convolutional layers are more robust and have more high-level spectral and spatial information. Additionally, SSRN and FDSSC learn deeper features by increasing the number of convolutional layers in the blocks. However, unlike these conventional models, AUSSC can go deeper with fixed parameters due to its loop structure and shared weights. Finally, an auxiliary loss function was used to reduce the intra-class distance and increase the distinction between features of different categories.

3. Experimental Data Sets and Framework Settings

3.1. Description of Experimental Data Sets

Three common HSI data sets were used to validate the proposed AUSSC, as follows: The Indiana Pines (IP; northwestern Indiana, USA), Kennedy Space Center (KSC; Merritt Island, FL, USA), and Salinas Scene (SS; Salinas Valley, CA, USA). These IP data were obtained by the NASA Airborne Visible Imaging Spectrometer (AVIRIS) sensor. The size of the IP data was 145×145, with 220 bands containing 16 kinds of ground cover. The KSC data were collected by the AVIRIS sensor in 1996 and had a size of 512×614, with 176 bands and 13 ground truth classes. The SS data were also collected by the AVIRIS sensor and had a size of 512×217, with 204 bands and 9 ground truth classes. Table 2 lists these classes and the corresponding false-color composite maps for three data sets.

However, with the development of state-of-art algorithms for hyperspectral image classification, these three data sets are easily classified. When the number of training samples was more than 800, SSRN and FDSSC achieved an accuracy higher than 98% for the three HSI data sets. The difference between the classification accuracies of these methods is less than 1%. Therefore, in addition to the three data sets discussed above, this study included the Houston (Houston, TX, USA) data set, which was distributed for the 2013 GRSS Data Fusion Contest [28]. The Houston data are more difficult as conventional algorithms (SSRN, FDSSC, etc.) have been unable to achieve classification above 90% with 200 labeled training samples. The size of the Houston data was 349 ×1905, with 144 bands containing 15 kinds of ground cover. Table 3 lists the classes and corresponding false-color composite maps for this data set.

Table 2. Color codes for the classes, class types, and sample numbers (SN) for the ground truths of the Indiana Pines (IP) data, Kennedy Space Center (KSC) data, and Salinas Scene (SS) data.

Color	IP Data		KSC Data		SS Data	
	Class	**SN**	**Class**	**SN**	**Class**	**SN**
🟥	Alfalfa	46	Scrub	347	Brocoli_green_weeds_1	2009
🟩	Corn-notill	1426	Willow swamp	243	Brocoli_green_weeds_2	3726
🟦	Corn-mintill	830	CP hammock	256	Fallow	1976
🟨	Corn	237	Slash pine	252	Fallow_rough_plow	1394
🟦	Grass-pasture	483	Oak/Broadleaf	161	Fallow_smooth	2678
🟪	Grass-trees	730	Hardwood	229	Stubble	3959
⬜	Grass-pasture-mowed	28	Swamp	105	Celery	3579
⬛	Hay-windrowed	478	Graminoid marsh	390	Grapes_untrained	11271
🟥	Oats	20	Spartina marsh	520	Soil_vinyard_develop	6203
🟫	Soybean-notill	972	Cattail marsh	404	Corn_senesced_green_weeds	3278
🟩	Soybean-mintill	2455	Salt marsh	419	Lettuce_romaine_4wk	1068
🟪	Soybean-clean	593	Mud flats	503	Lettuce_romaine_5wk	1927
🟦	Wheat	205	Water	927	Lettuce_romaine_6wk	916
🟦	Woods	1265			Lettuce_romaine_7wk	1070
🟧	Buildings-Grass-Trees	386			Vinyard_untrained	7268
🟨	Stone-Steel-Towers	93			Vinyard_vertical_trellis	1807
Total	10,249		5211		54,129	

Table 3. Color codes for the classes, class types, and sample numbers (SN) for the ground truth of the Houston data.

NO.	Color	Class	SN
1	🟥	Grass Healthy	1374
2	🟩	Grass Stressed	1454
3	🟦	Grass Synthetic	795
4	🟨	Tree	1264
5	🟦	Soil	1298
6	🟪	Water	339
7	⬜	Residential	1476
8	⬛	Commercial	1354
9	🟥	Road	1554
10	🟫	Highway	1424
11	🟩	Railway	1332
12	🟪	Parking Lot 1	1429
13	🟦	Parking Lot 2	632
14	🟦	Tennis Court	513
15	🟧	Running Track	798
		Total	17,036

Quantitative analysis was performed with the same limited training samples using all methods. Different sets of training samples were used to demonstrate the effectiveness of the AUSSC method under different conditions. A subset of 200 labeled samples were used for training and 100 labeled samples were used for validating. A series of 400, 600, 800, and 1000 training samples were then included to test the robustness and generalizability of the proposed AUSSC.

3.2. Framework Setting

The framework for all data sets was established as follows. From 10 random seeds, all data sets were randomly divided into the three following groups: A training set, a validation set, and a test set. The training sets were used to optimize model parameters. The validation sets were not directly used in the training process and were only included to verify whether the model was overfitting. The testing sets were used to test the performance of the model after the training was completed. The number of validation sets was half the number of training sets and the remainder of the sets were test sets. The batch size was set to 16 and the Adam [29] optimizer was used for stochastic optimization. The initialization of model weights was performed using the He normal distribution method [30] for all 3D convolutional layers and the Xavier normal distribution method for the fully connected layer [31]. We used a variable learning rate, which was gradually reduced during the optimization process. This was done because the learning rate must be smaller when closer to the valley. The number of training epochs was set to 400 and the initial learning rate was set to 0.0001 for IP, KSC, and SS data sets and 0.0003 for the Houston data set. The learning rate was halved when the validation loss did not decrease after 10 epochs.

In addition to these basic settings, four key factors were used to configure the AUSSC framework for HSI classification. Namely, (1) the number of convolutional layers and loops in one block of stage 2; (2) the number of convolutional kernels in alternately updated blocks; (3) the spatial size of input cubes; and (4) the coefficients of the center loss function. These four factors are discussed by the OA of IP, KSC, and SS below.

First, the number of convolutional layers and loops in each block of stage 2 determined the depth of the entire network, which consequently affected classification accuracy and runtime. As shown in Figure 6, appropriately increasing the number of convolutional layers and the number of loops improved classification. However, the network depth had a significant impact on training time and was almost linearly related to the training time. Therefore, we used two convolutional layers and only loop in each block to conserve training time.

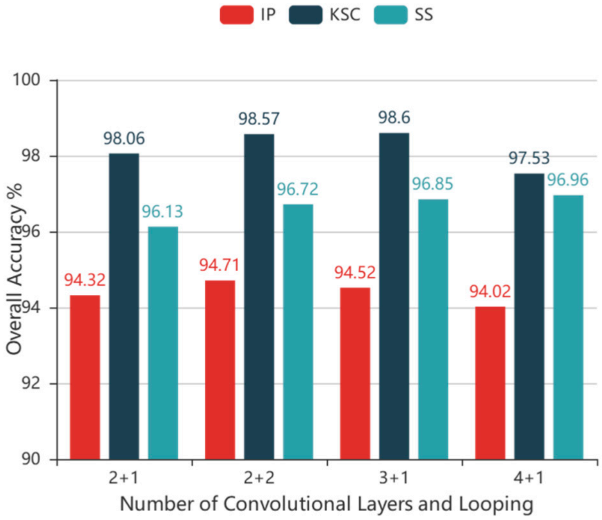

Figure 6. The overall accuracy of the AUSSC with different numbers of convolutional layers and loops in each block. The $a + b$ notation on the x-axis denotes the AUSSC with a convolutional layers and b looping iterations in stage 2 of each block. IP: Indiana Pines data set; KSC: Kennedy Space Center data set; SS: Salinas Scene data set.

Second, increasing the number of convolutional kernels often extracted more rich features. If enough convolutional kernels were provided, abstract high-order structures could be efficiently learned from the convolutional layer. As shown in Figure 7, the overall accuracy (OA) of the AUSSC was weakly positively related to the number of convolutional kernels, which had little effect on training time. Combining the performance of the AUSSC for the three data sets, the number of kernels in the first convolutional layer was set to 64 in each block and the number of kernels was set to 36 in two blocks.

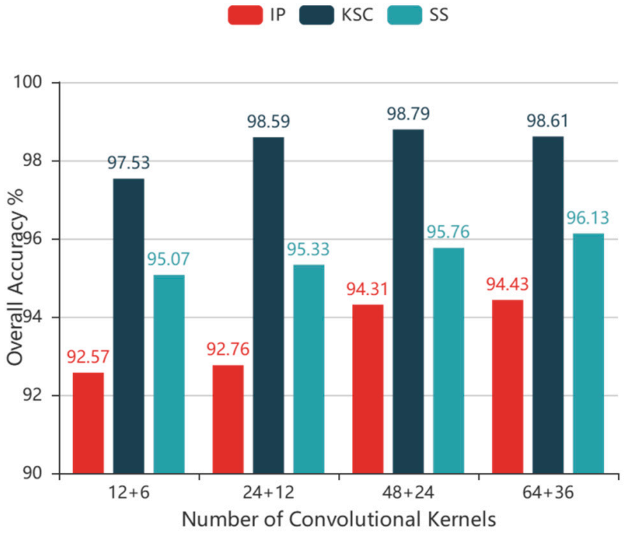

Figure 7. The overall accuracy of the AUSSC with different numbers of convolutional kernels in the first layer using two blocks. The $a + b$ notation on the x-axis denotes an AUCN with a kernels in the first convolutional layer and b kernels in the two blocks.

Third, a larger input space allowed more spatial information to be extracted. Input samples with spatial sizes of 5×5, 7×7, 9×9, and 11×11 were used in the three data sets. As shown in Figure 8, the OAs of the IP, KSC, and SS data sets increased with increasing input spatial size. However, for inputs with spatial sizes greater than or equal to 9×9, the increase in OA was less than 1%. Considering the cost of calculation, the 9×9 spatial size was selected for all data sets to test the performance of the AUSSC framework.

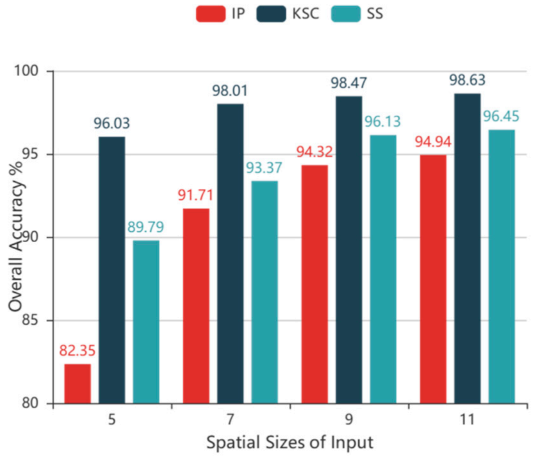

Figure 8. The overall accuracy of the AUSSC with different input spatial sizes.

Moreover, the coefficient of center loss also played an important role in our proposed AUSSC. The coefficient of L2 loss was set to 0.0001 and the possible values of the coefficients for center loss were set to 0, 0.1, 0.01, and 0.001. As shown in Figure 9, the center loss could not be used directly as an objective function. However, as an auxiliary objective function, the center loss can slightly increase the overall classification accuracy. When the coefficient of center loss was set to 0.001, the OA of the AUSSC using the IP and SS data sets increased slightly. However, the OA of the AUSSC using the KSC data set increased by nearly 1%. As such, the coefficient of center loss was set to 0.001.

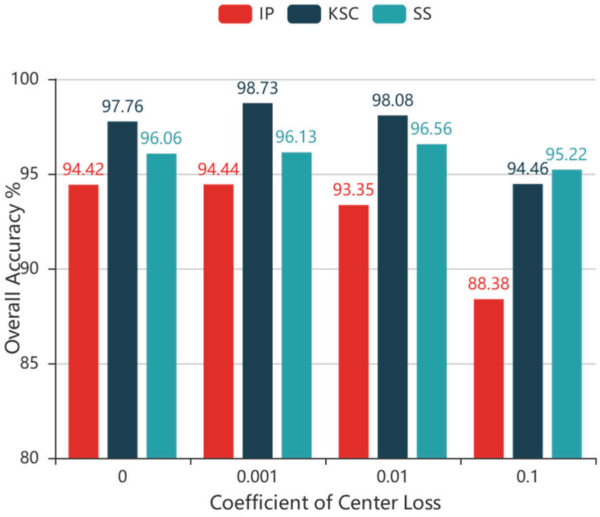

Figure 9. The overall accuracy of AUSSC for different center-loss coefficients.

4. Classification Results and Discussion

4.1. Experimental Results

In this section, we compare the proposed AUSSC framework with deep learning-based methods, including SAE-LR [14], CNN [18], SSRN [21], 3D-GAN [24], and FDSSC [22]. As SSRN, FDSSC, and the proposed AUSSC are all 3D CNN-based methods, the input spatial size was fixed at 9 × 9 to allow a fair comparison. Ten groups of 200 training samples were randomly selected from the IP, KSC, SS, and Houston data sets. The classification accuracy indices for the experiment included the OA, average accuracy (AA), and kappa coefficient (K). The results of these three metrics are displayed in the form of mean ± standard deviation. The original hyperspectral data were normalized to a zero mean and standard deviation of one. The dimensions of the image block were the same as those of the original hyperspectral data. Figures 10–13 show classification results obtained from the IP, KSC, SS, and Houston data sets using different algorithms.

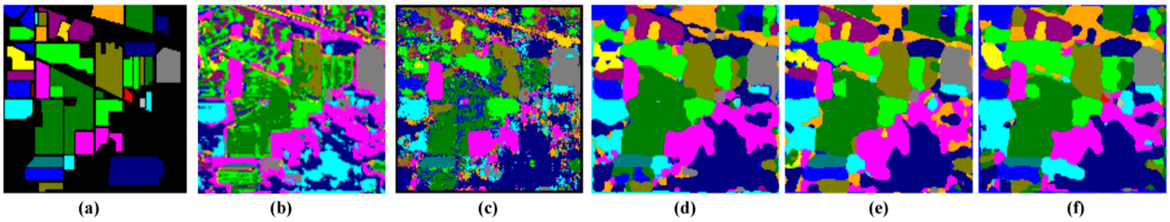

Figure 10. Classification results for the Indian Pines data set obtained using different methods. (**a**) Ground-truth map; (**b**) SAE-LR; (**c**) CNN; (**d**) SSRN; (**e**) FDSSC; and (**f**) AUSSC.

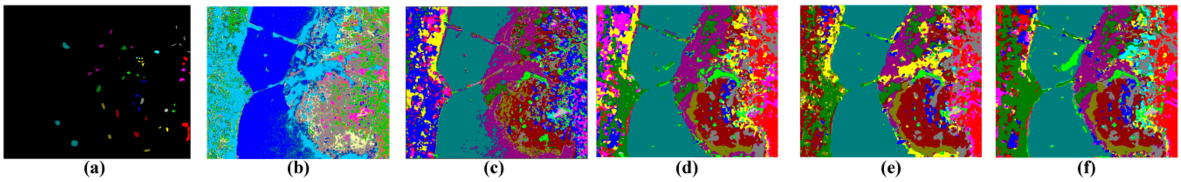

Figure 11. Classification results obtained from the KSC data set using different methods. (**a**) Ground-truth map; (**b**) SAE-LR; (**c**) CNN; (**d**) SSRN; (**e**) FDSSC; and (**f**) AUSSC.

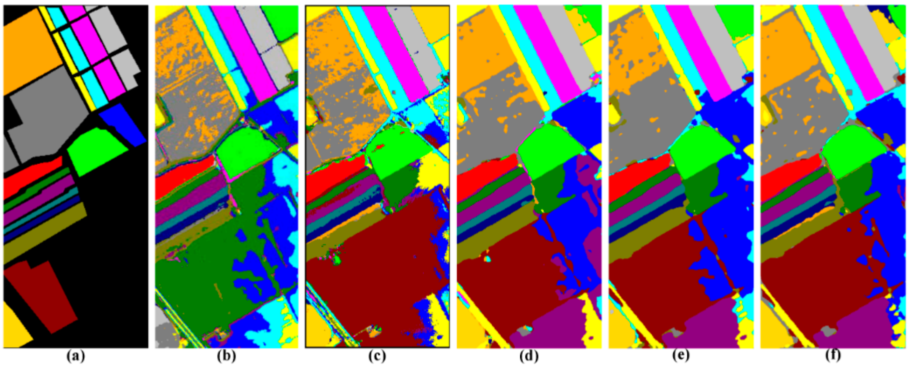

Figure 12. Classification results obtained from the SS data set using different methods. (**a**) Ground-truth map; (**b**) SAE-LR; (**c**) CNN; (**d**) SSRN; (**e**) FDSSC; and (**f**) AUSSC.

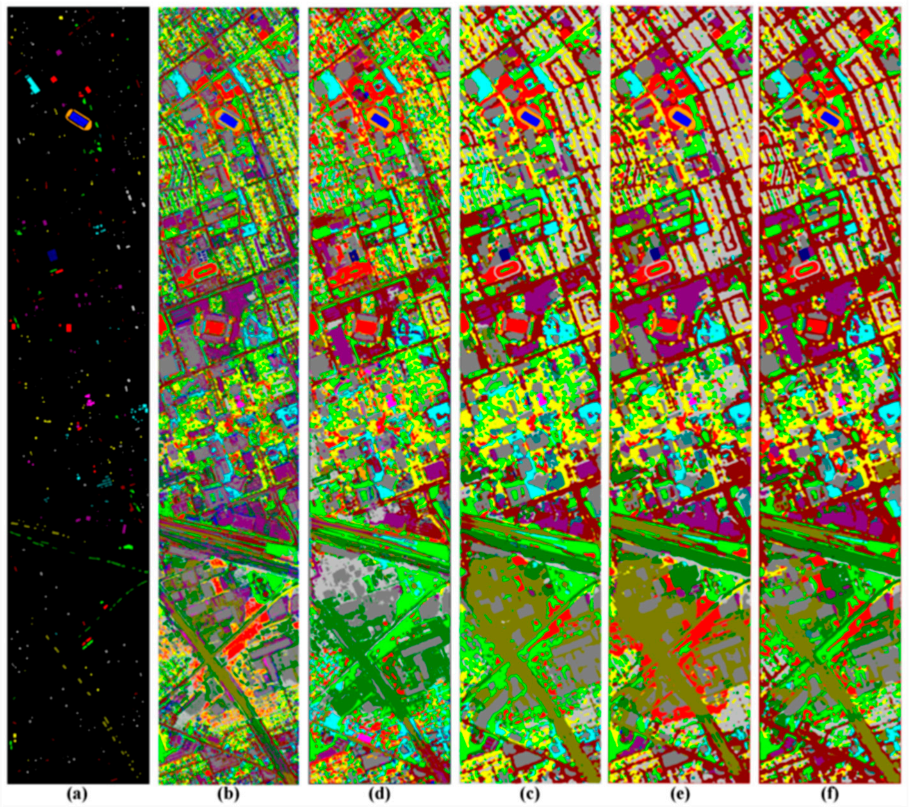

Figure 13. Classification results obtained from the Houston data set using different methods. (**a**) Ground-truth map; (**b**) SAE-LR; (**c**) CNN; (**d**) SSRN; (**e**) FDSSC; and (**f**) AUSSC.

Tables 4–7 display the results of the OA, AA, kappa coefficient, and accuracy of each category for the IP, KSC, SS, and Houston data sets and the best accuracy is shown in bold. These experimental results show that our proposed AUSSC method is superior to early deep learning methods (SAE-LR and CNN), novel 3D-GAN, and recent 3D CNN-based methods (SSRN and FDSSC).

Table 4. Overall accuracy (OA), average accuracy (AA), kappa coefficient (K), and accuracy for each HSI category in the Indiana Pines (IP) data set. Data are given as mean ± standard deviation.

Methods	SAE-LR	CNN	SSRN	3D-GAN	FDSSC	AUSSC
OA (%)	57.44 ± 0.56	59.84 ± 0.98	90.47 ± 2.24	90.69 ± 0.86	92.12 ± 1.05	**94.55 ± 1.09**
AA (%)	46.30 ± 1.17	51.60 ± 1.17	88.46 ± 3.29	83.14 ± 1.82	83.16 ± 5.08	**94.44 ± 1.11**
K ×100	51.04 ± 0.58	53.97 ± 1.03	89.12 ± 2.54	89.62 ± 0.26	91.02 ± 1.18	**93.77 ± 1.25**
1	22.89 ± 7.33	1.33 ± 1.09	90.00 ± 30.0	30.21 ± 1.03	70.00 ± 45.8	**98.33 ± 5.00**
2	45.46 ± 3.80	41.53 ± 3.04	87.35 ± 11.6	81.79 ± 0.26	90.38 ± 4.87	**94.60 ± 3.04**
3	26.69 ± 2.61	30.91 ± 8.28	87.18 ± 7.24	75.93 ± 1.26	86.48 ± 3.99	**90.06 ± 4.53**
4	33.80 ± 10.2	13.28 ± 4.16	**94.28 ± 6.63**	90.08 ± 1.23	92.34 ± 6.67	93.73 ± 5.98
5	46.82 ± 7.78	70.80 ± 1.59	95.89 ± 3.66	86.39 ± 2.12	96.65 ± 1.95	**98.59 ± 2.55**
6	79.64 ± 1.71	90.78 ± 0.78	94.09 ± 2.07	93.28 ± 0.23	95.11 ± 2.44	**97.84 ± 0.81**
7	41.45 ± 10.5	20.74 ± 8.95	71.98 ± 37.8	40.71 ± 1.05	40.00 ± 49.0	**86.57 ± 13.0**
8	96.57 ± 1.26	94.93 ± 4.04	94.13 ± 3.30	**98.11 ± 0.21**	92.51 ± 2.06	97.22 ± 2.11
9	38.82 ± 23.4	0.00 ± 0.00	50.0 ± 50.0	20.00 ± 1.96	10.00 ± 30.0	**91.61 ± 9.49**
10	50.47 ± 1.91	52.53 ± 1.24	86.47 ± 7.69	74.28 ± 0.89	84.87 ± 9.39	**92.40 ± 2.86**
11	70.89 ± 2.49	61.88 ± 4.33	91.88 ± 5.03	91.12 ± 0.25	**95.32 ± 2.12**	93.97 ± 3.29
12	28.41 ± 6.67	26.57 ± 2.96	88.93 ± 6.27	84.99 ± 1.46	92.45 ± 4.35	**94.52 ± 2.55**
13	22.57 ± 8.28	94.03 ± 1.33	97.15 ± 4.18	49.75 ± 2.45	**99.70 ± 0.90**	97.67 ± 2.99
14	78.23 ± 5.89	93.74 ± 1.40	96.03 ± 3.11	94.38 ± 0.26	96.39 ± 2.74	**96.65 ± 1.49**
15	39.57 ± 7.08	33.46 ± 3.23	92.00 ± 6.50	94.47 ± 0.79	90.36 ± 9.55	**94.63 ± 1.59**
16	18.59 ± 21.6	**99.12 ± 0.82**	97.98 ± 2.71	84.22 ± 1.16	98.01 ± 1.33	92.70 ± 5.28

Table 5. OA, AA, K, and accuracy for each HSI category in the Kennedy Space Center (KSC) data set.

Methods	SAE-LR	CNN	SSRN	3D-GAN	FDSSC	AUSSC
OA (%)	57.68 ± 1.74	69.87 ± 0.36	96.23 ± 1.40	96.89 ± 1.24	96.28 ± 1.26	**98.26 ± 0.70**
AA (%)	44.99 ± 4.59	69.59 ± 0.44	94.58 ± 1.53	94.14 ± 0.40	94.58 ± 1.70	**97.48 ± 1.01**
K ×100	52.45 ± 1.94	65.87 ± 0.37	95.80 ± 1.56	96.52 ± 0.26	95.86 ± 1.41	**98.0 ± 0.78**
1	81.87 ± 15.5	4.09 ± 2.79	97.87 ± 3.50	98.29 ± 0.42	97.99 ± 2.24	**99.02 ± 1.02**
2	52.83 ± 29.6	85.29 ± 1.91	94.29 ± 5.66	79.84 ± 1.45	93.01 ± 4.35	**96.35 ± 6.83**
3	35.82 ± 33.2	69.38 ± 1.94	85.31 ± 13.7	98.44 ± 0.14	86.65 ± 10.3	**98.98 ± 3.05**
4	0.00 ± 0.00	32.06 ± 1.78	84.11 ± 11.4	86.51 ± 1.12	76.02 ± 10.7	**87.54 ± 6.86**
5	21.22 ± 22.4	52.50 ± 2.09	82.92 ± 16.0	**98.7 ± 0.14**	86.61 ± 11.2	94.24 ± 8.50
6	0.37 ± 0.51	58.60 ± 1.40	96.51 ± 5.23	**100.00 ± 0.00**	96.23 ± 5.82	97.53 ± 4.94
7	23.56 ± 19.0	99.23 ± 1.54	93.14 ± 8.05	**97.14 ± 1.06**	97.32 ± 4.68	96.12 ± 8.69
8	66.24 ± 10.1	62.13 ± 2.46	97.83 ± 1.67	72.95 ± 2.10	97.28 ± 2.27	**98.58 ± 0.87**
9	51.76 ± 25.0	86.69 ± 0.31	99.84 ± 0.22	99.23 ± 0.09	99.84 ± 0.18	**99.92 ± 0.13**
10	20.00 ± 14.4	73.47 ± 2.67	99.02 ± 2.38	**100.00 ± 0.00**	**100.00 ± 0.00**	99.97 ± 0.08
11	89.64 ± 3.48	90.00 ± 0.00	99.50 ± 0.73	**100.00 ± 0.00**	98.70 ± 1.63	99.18 ± 1.29
12	51.24 ± 12.0	93.25 ± 0.43	99.25 ± 0.85	96.48 ± 1.23	**99.87 ± 0.22**	99.83 ± 0.25
13	90.44 ± 3.00	97.93 ± 0.22	99.91 ± 0.21	**100.00 ± 0.00**	**100.00 ± 0.00**	**100.00 ± 0.00**

Table 6. OA, AA, K, and accuracy for each HSI category in the Salinas Scene (SS) data set.

Methods	SAE-LR	CNN	SSRN	3D-GAN	FDSSC	AUSSC
OA (%)	71.51 ± 0.10	85.92 ± 2.79	94.02 ± 2.79	93.02 ± 1.54	95.50 ± 0.69	**96.13 ± 0.57**
AA (%)	78.03 ± 0.14	91.14 ± 2.02	97.39 ± 0.63	89.15 ± 0.39	**97.41 ± 0.32**	97.37 ± 0.53
K ×100	68.48 ± 0.11	84.42 ± 3.03	93.34 ± 3.10	92.07 ± 1.22	94.99 ± 0.77	**95.70 ± 0.64**
1	99.26 ± 0.26	**100.00 ± 0.00**	**100.00 ± 0.00**	98.12 ± 1.02	**100.00 ± 0.00**	**100 ± 0.00**
2	98.90 ± 0.22	**99.42 ± 0.77**	98.90 ± 2.00	94.11 ± 0.12	99.82 ± 0.42	99.82 ± 0.45
3	80.92 ± 0.84	64.65 ± 20.8	**98.38 ± 1.40**	76.46 ± 0.28	95.40 ± 2.60	95.04 ± 4.35
4	98.71 ± 0.11	98.74 ± 0.34	98.97 ± 0.94	**100.00 ± 0.47**	98.08 ± 1.55	97.99 ± 1.47
5	74.30 ± 0.19	97.69 ± 1.63	98.92 ± 1.46	88.25 ± 1.89	**99.42 ± 0.61**	98.69 ± 2.31
6	99.8 ± 0.03	99.97 ± 0.06	99.94 ± 0.16	99.34 ± 0.36	**99.99 ± 0.01**	99.98 ± 0.05
7	99.22 ± 0.09	99.74 ± 0.17	**99.97 ± 0.01**	99.90 ± 0.67	99.44 ± 0.79	99.56 ± 0.65
8	78.78 ± 3.10	56.58 ± 19.7	89.97 ± 9.86	89.44 ± 1.13	90.27 ± 4.42	**92.83 ± 2.96**
9	0.00 ± 0.00	99.99 ± 0.01	99.49 ± 0.66	**100.00 ± 0.00**	99.51 ± 0.43	99.41 ± 0.21
10	74.08 ± 0.84	86.06 ± 4.35	**98.71 ± 1.75**	98.13 ± 1.00	96.29 ± 3.27	98.25 ± 2.22
11	93.68 ± 0.28	85.14 ± 4.31	96.07 ± 2.01	**96.69 ± 2.12**	96.33 ± 1.51	93.90 ± 4.29
12	**99.99 ± 0.03**	92.66 ± 8.38	99.05 ± 0.65	99.06 ± 1.04	98.06 ± 1.72	98.40 ± 1.60
13	**99.18 ± 0.06**	98.16 ± 3.06	98.43 ± 1.88	77.92 ± 1.68	99.08 ± 1.22	98.59 ± 1.41
14	94.28 ± 0.49	97.46 ± 1.87	98.23 ± 1.64	78.21 ± 0.67	**98.35 ± 1.66**	96.13 ± 4.98
15	57.39 ± 0.96	85.57 ± 8.72	83.26 ± 11.8	70.88 ± 0.45	88.62 ± 5.21	**89.32 ± 4.69**
16	0.00 ± 0.00	96.43 ± 5.90	**100.00 ± 0.00**	90.0 ± 0.12	99.85 ± 0.30	99.94 ± 1.70

Table 7. OA, AA, K, and accuracy for each HSI category in the Houston data set.

Methods	SAE-LR	CNN	SSRN	FDSSC	AUSSC
OA (%)	76.18 ± 3.22	75.01 ± 0.75	88.89 ± 2.25	89.40 ± 1.26	**91.21 ± 1.57**
AA (%)	75.44 ± 0.14	76.45 ± 0.58	91.35 ± 1.71	91.39 ± 1.02	**93.30 ± 1.04**
K ×100	74.06 ± 3.82	72.98 ± 0.81	87.99 ± 2.43	88.55 ± 1.36	**90.50 ± 1.70**
1	95.96 ± 0.54	86.61 ± 2.38	91.44 ± 5.19	90.91 ± 5.66	**96.17 ± 2.18**
2	85.84 ± 0.17	**99.19 ± 0.12**	94.92 ± 3.06	93.02 ± 8.54	94.55 ± 3.26
3	95.35 ± 0.41	95.42 ± 0.81	99.50 ± 0.55	99.49 ± 1.03	**99.61 ± 0.60**
4	95.52 ± 0.17	83.06 ± 1.08	97.46 ± 2.82	**99.16 ± 0.39**	97.81 ± 2.37
5	94.28 ± 0.54	**100 ± 0.00**	96.55 ± 2.40	96.61 ± 4.08	97.84 ± 2.78
6	69.91 ± 1.26	87.58 ± 0.24	99.75 ± 0.74	**100 ± 0.00**	**100 ± 0.00**
7	70.46 ± 0.64	70.69 ± 3.08	83.18 ± 8.13	89.19 ± 3.07	**90.14 ± 7.56**
8	68.31 ± 2.05	62.17 ± 1.69	**96.18 ± 4.10**	95.81 ± 4.23	95.89 ± 4.09
9	65.76 ± 0.19	72.56 ± 4.66	78.42 ± 6.49	**83.22 ± 7.48**	78.91 ± 6.99
10	59.64 ± 0.19	49.79 ± 3.45	78.51 ± 8.40	79.95 ± 5.48	**87.78 ± 5.84**
11	76.71 ± 0.56	76.32 ± 3.62	82.14 ± 87.23	82.99 ± 7.12	**89.28 ± 7.02**
12	88.03 ± 0.79	36.12 ± 8.41	**88.03** ± 6.54	82.97 ± 6.57	81.04 ± 8.89
13	11.58 ± 0.42	32.51 ± 5.44	88.52 ± 8.26	82.59 ± 8.27	**96.11 ± 0.40**
14	58.75 ± 0.33	94.75 ± 1.36	97.42 ± 3.62	**97.26 ± 3.11**	96.60 ± 3.25
15	95.46 ± 0.57	**100 ± 0.00**	98.32 ± 0.88	97.68 ± 1.56	97.75 ± 1.46

As shown in Table 4, the values of OA, AA, and K, obtained using the AUSSC, were 2.43%, 11.28%, and 2.75% higher than those obtained using FDSSC, which exhibited the second-best performance for the IP data set. AUSSC also achieved the best classification accuracy in 10 categories of the IP data set. AUSSC achieved an accuracy similar to SSRN and 3D-GAN for Class 4 (Corn) and Class 8 (Hay-windrowed), respectively. FDSSC achieved significantly better results for Class 11 (Soybean-mintill) and Class 13 (Wheat). However, FDSSC (like other methods), achieved poor results for Class 9 (Oats), with an average accuracy of only 10%. In contrast, AUSSC achieved excellent results with an average accuracy of 92.61%. CNN achieved the best results for Class 16 (Stone-Steel-Towers) but produced 0% for Class 9 (Oats) and performed poorly in four other categories.

As shown in Table 5, the values of OA, AA, and K obtained using AUSSC were respectively 1.37%, 3.34%, and 1.48% higher than those produced by 3D-GAN, which exhibited the second-best performance for the KSC data set. AUSSC also achieved the best accuracy in 7 of 13 KSC categories, producing results similar to those of 3D-GAN for Class 10 (Cattail marsh) and Class 11 (Salt marsh). 3D-GAN achieved significantly better results for Class 5 (Oak), Class 6 (Hardwood), and Class 7 (Swamp). However, its accuracy for Class 2 (Willow swamp) and Class 8 (Graminoid marsh) was ~20% lower than that of our method.

As shown in Table 6, the values of OA and K obtained using AUSSC were 0.63% and 0.71% higher than those produced by FDSSC, which exhibited the second-best performance for the SS data set. AUSSC achieved similar or better results than FDSSC across all 16 categories in the SS data set. As shown in Table 7, the values of OA, AA, and K, obtained using AUSSC, were 1.81%, 1.91%, and 1.95% higher than those obtained by FDSSC, which exhibited the second-best performance for the Houston data set. CNN achieved excellent results for Class 2 (Grass Stressed), Class 5 (Soil), and Class 15 (Running Track). However, the accuracy of CNN in Category 10 (Highway), Class 12 (Parking Lot 1), and Class 13 (Parking Lot 2) was ~40% lower than that of our method.

These experimental results indicate AUSSC achieved the best performance in terms of OA and K for all four HSI data sets. Other methods, especially CNN, were superior to our methods in some categories, but performed poorly in others. These poorly performing categories dramatically reduced the OA, AA, and K.

With the exception of the 3D-GAN data, which were obtained from the literature, these classification results shown in Tables 4–7 were trained and tested using a desktop computer with 32 GB of memory equipped with an NVIDA GTX 1080Ti GPU. Table 8 shows the mean and standard deviation of the training time and testing time for 10 runs using CNN-based methods and the minimum time is shown in bold. As shown in the tables, the training times for deep 3D CNN-based methods were longer than those of other deep learning-based methods. The AUSSC required a longer training time than SSRN or FDSSC. For AUSSC applied to the IP data set, the number of floating-point operations per second (FLOPs) was 5362.386 K and the number of parameters was 761.064 K.

Table 8. Training and testing times for CNN-based methods across the four data sets.

Data set	Time	CNN	SSRN	FDSSC	AUSSC
IP	Training/sec	**9.25 ± 0.40**	73.9 ± 5.32	63.5 ± 3.72	439 ± 4.16
	Testing/sec	**0.71 ± 0.11**	6.84 ± 0.20	7.91 ± 0.11	11.1 ± 0.21
KSC	Training/sec	**8.10 ± 0.51**	72.0 ± 2.50	57.1 ± 4.95	420 ± 4.21
	Testing/sec	**0.46 ± 0.16**	2.18 ± 0.07	3.40 ± 0.06	4.83 ± 0.07
SS	Training/sec	**10.1 ± 0.59**	77.9 ± 3.00	63.5 ± 4.02	433 ± 3.99
	Testing/sec	**2.05 ± 0.11**	27.0 ± 0.86	43.1 ± 0.50	58.9 ± 0.47
Houston	Training/sec	**10.2 ± 0.55**	149 ± 4.34	83.5 ± 1.26	594 ± 4.80
	Testing/sec	**1.52 ± 0.05**	13.9 ± 0.49	10.9 ± 0.13	11.9 ± 0.18

To corroborate the robustness and generalizability of the proposed method, Figures 14 and 15 show the OA obtained using different methods for different training samples. When the number of training samples was higher than 400, our method performed similarly to SSRN and FDSSC. This is because the OA of SSRN and FDSSC reached more than 98%, creating a small gap between our method and these conventional techniques. This also demonstrates that the three datasets published more than 10 years ago are easily classified by state-of-the-art methods. The Houston data set, provided by the University of Houston for the 2013 IEEE GRSS Data Fusion Contest, is more challenging. As shown in Figure 15, it is more discriminant than the three datasets in comparing AUSSC with other methods. The resulting difference in OA between AUSSC, FDSSC, and SSRN was more than 1%.

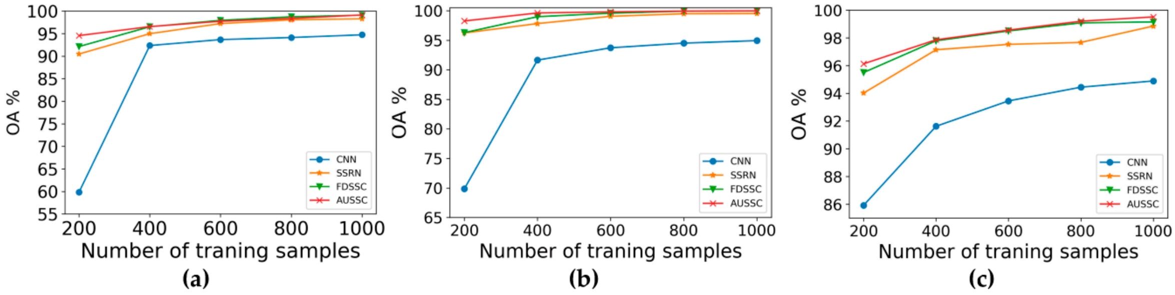

Figure 14. The overall accuracy (OA) of different methods for different numbers of training samples in three HSI data sets. (**a**) Indiana Pines (IP) data set; (**b**) Kennedy Space Center (KSC) data set; and (**c**) Salinas Scene (SS) data set.

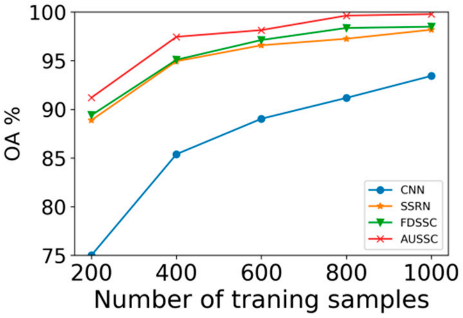

Figure 15. The overall accuracy (OA) of different methods for different numbers of training samples in the Houston data set.

4.2. Discussion

In this study, a highly limited number of training samples (200) was used to demonstrate that our proposed method can reduce data dependence. Insufficiently labeled data are unavoidable in remote sensing applications. Additionally, the collection and labeling of remote sensing data is complex and expensive. Thus, it is very difficult to build large-scale, high-quality labeled sets. The number of labeled samples used for training is the most important factor in deep-learning supervised methods, as data dependence is one of the most serious problems in deep learning. Compared with traditional machine-learning methods, deep learning relies heavily on large-scale training data, which are necessary to understand potential patterns. Semi-supervised 3D-GANs also require ~200 training samples; however, their classification accuracy is significantly lower.

The proposed method offers three principal benefits. First, it provides an end-to-end framework for HSI classification. SAE-LR, CNN, and 3D-GAN all require PCA to preprocess hyperspectral data. Second, deep CNN architectures and convolutional kernels were used to determine classification accuracy in 3D CNN-based methods [20]. These networks include only two convolutional layers with $3 \times 3 \times m$ convolutional kernels. SSRN and FDSSC use residual blocks, dense blocks, and two different convolutional kernels to learn deep spectral and spatial features. The biggest difference between AUSSC and the 3D CNN-based methods discussed above is its use of recurrent CNN architectures and three 1D convolutional kernels. Alternately updated blocks can not only learn deep spectral and spatial features but also refined spectral and spatial features. As a result, three 1D convolutional kernels can be combined to generate more abundant features. As a result, AUSSC achieved better classification accuracy than current state-of-the-art deep learning-based methods. Finally, unlike these other methods, only cross-entropy objective functions were used in the AUSSC. We also introduced center loss in the AUSSC as an auxiliary objective function to learn more discriminating features.

Although the proposed method provides better performance than conventional architectures (especially SSRN and FDSSC), it has a much higher computational requirement (see Table 8). There are three primary reasons for this. First, AUSSC uses more convolutional kernels in two blocks than

SSRN and FDSSC. Second, the use of the center loss function increases the computational cost. Finally, and most importantly, more training epochs are used in AUSSC than in SSRN and FDSSC. In fact, the training time for one epoch in AUSSC is only slightly longer than in FDSSC or SSRN. However, AUSSC requires far more training epochs. The regular updating of graphics cards and the use of high performance graphics cards, such as the NVIDIA GeForce RTX 2080Ti, could effectively alleviate this problem.

5. Conclusions

In this study, refined spectral and spatial features in HSIs were used as core concepts to design an end-to-end CNN-based framework for HSI classification. This alternately updated convolutional spectral–spatial network utilizes alternately updated spectral and spatial blocks and primarily includes small convolutional kernels in three different dimensions to learn HIS features, combining them into advanced features.

The learning of deep refined spectral and spatial features by alternately updated blocks makes our method superior to other deep learning-based methods, as this allows it to achieve a high classification accuracy. Furthermore, experimental results also demonstrated that the center loss function can slightly improve the classification accuracy of hyperspectral images. Results showed that when 200 training samples were used from different HSI data sets, the AUSSC achieved the highest classification accuracy among the deep learning-based methods for all three data sets. Additionally, using different training samples, the AUSSC was also found to be the best method in terms of OA for all HSI data sets. However, the AUSSC has a longer training time than other conventional algorithms. In a future study, network pruning will be used to reduce the heavy calculation of the deep model.

Author Contributions: All authors made significant contributions to this work. W.W. and S.D. conceived and designed the experiments; S.D. performed the experiments; W.W. and S.W. analyzed the data; and S.W. contributed analysis tools.

Funding: The financial support for this work was provided in part by the Natural Science Foundation of Shanghai under Grant 19ZR1435900, in part by the Bidding Project of the Lab of Green Plate-making and Standardization for Flexographic Printing under Grant ZBKT201710 and in part by the Shanghai Research Institute of Publishing and Media in 2018 under Grant SAYB1803.

Acknowledgments: The IP, KSC, and SS data used in this study were obtained from public domains and are available online at http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes. The authors would like to thank the Hyperspectral Image Analysis group and the NSF Funded Center for Airborne Laser Mapping (NCALM) at the University of Houston for providing the Houston data set used in this study and the IEEE GRSS Data Fusion Technical Committee for organizing the 2013 Data Fusion Contest.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analysis, or interpretation of data; in the writing of the manuscript; nor in the decision to publish the results.

References

1. Willett, R.M.; Duarte, M.F.; Davenport, M.A.; Baraniuk, R.G. Sparsity and structure in hyperspectral imaging: Sensing, reconstruction, and target detection. *IEEE Signal Process. Mag.* **2014**, *31*, 116–126. [CrossRef]
2. Caporaso, N.; Whitworth, M.B.; Grebby, S.; Fisk, I.D. Non-destructive analysis of sucrose, caffeine and trigonelline on single green coffee beans by hyperspectral imaging. *Food Res. Int.* **2018**, *106*, 193–203. [CrossRef] [PubMed]
3. Thomas, S.; Kuska, M.T.; Bohnenkamp, D.; Brugger, A.; Alisaac, E.; Wahabzada, M.; Behmann, J.; Mahlein, A.K. Benefits of hyperspectral imaging for plant disease detection and plant protection: A technical perspective. *J. Plant Dis. Prot.* **2018**, *125*, 5–20. [CrossRef]
4. Lu, X.Q.; Yuan, Y.; Zheng, X.T. Joint dictionary learning for multispectral change detection. *IEEE Trans. Cybern.* **2017**, *47*, 884–897. [CrossRef] [PubMed]
5. Lu, G.L.; Fei, B.W. Medical hyperspectral imaging: A review. *J. Biomed. Opt.* **2014**, *19*, 010901. [CrossRef] [PubMed]

6. Melgani, F.; Bruzzone, L. Classification of hyperspectral remote sensing images with support vector machines. *IEEE Trans. Geosci. Remote Sens.* **2004**, *42*, 1778–1790. [CrossRef]
7. Ghamisi, P.; Maggiori, E.; Li, S.; Souza, R.; Tarablaka, Y.; Moser, G.; Giorgi, A.D.; Fang, L.; Chen, Y.; Chi, M.; et al. New frontiers in spectral-spatial hyperspectral image classification: The latest advances based on mathematical morphology, markov random fields, segmentation, sparse representation, and deep learning. *IEEE Geosci. Remote Sens. Mag.* **2018**, *6*, 10–43. [CrossRef]
8. Fauvel, M.; Benediktsson, J.A.; Chanussot, J.; Sveinsson, J.R. Spectral and spatial classification of hyperspectral data using svms and morphological profiles. *IEEE Trans. Geosci. Remote Sens.* **2008**, *46*, 3804–3814. [CrossRef]
9. Ghamisi, P.; Benediktsson, J.A.; Ulfarsson, M.O. Spectral-spatial classification of hyperspectral images based on hidden markov random fields. *IEEE Trans. Geosci. Remote Sens.* **2014**, *52*, 2565–2574. [CrossRef]
10. Chen, Y.; Nasrabadi, N.M.; Tran, T.D. Hyperspectral image classification via kernel sparse representation. *IEEE Trans. Geosci. Remote Sens.* **2013**, *51*, 217–231. [CrossRef]
11. 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]
12. 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]
13. Chen, Y.S.; Zhao, X.; Jia, X.P. Spectral-spatial classification of hyperspectral data based on deep belief network. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2015**, *8*, 2381–2392. [CrossRef]
14. Makantasis, K.; Karantzalos, K.; Doulamis, A.; Doulamis, N. Deep supervised learning for hyperspectral data classification through convolutional neural networks. In Proceedings of the 2015 IEEE International Geoscience and Remote Sensing Symposium, New York, NY, USA, 26–31 July 2015; pp. 4959–4962.
15. 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]
16. Li, Y.; Zhang, H.; Shen, Q. Spectral-spatial classification of hyperspectral imagery with 3d convolutional neural network. *Remote Sens.* **2017**, *9*, 67. [CrossRef]
17. Zhong, Z.L.; Li, J.; Luo, Z.M.; Chapman, M. Spectral-spatial residual network for hyperspectral image classification: A 3-d deep learning framework. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 847–858. [CrossRef]
18. Wang, W.J.; Dou, S.G.; Jiang, Z.M.; Sun, L.J. A fast dense spectral-spatial convolution network framework for hyperspectral images classification. *Remote Sens.* **2018**, *10*, 1068. [CrossRef]
19. 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]
20. Zhu, L.; Chen, Y.S.; Ghamisi, P.; Benediktsson, J.A. Generative adversarial networks for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 5046–5063. [CrossRef]
21. Mnih, V.; Heess, N.; Graves, A.; Kavukcuoglu, K. Recurrent models of visual attention. In Proceedings of the 28th Annual Conference on Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; pp. 2204–2212.
22. Wang, F.; Jiang, M.; Qian, C.; Yang, S.; Li, C.; Zhang, H.; Wang, X.; Tang, X. Residual attention network for image classification. In Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR, Honolulu, HI, USA, 21–26 July 2017; pp. 6450–6458.
23. Stollenga, M.F.; Masci, J.; Gomez, F.; Schmidhuber, J. Deep networks with internal selective attention through feedback connections. In Proceedings of the 28th Annual Conference on Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; pp. 3545–3553.
24. Wen, Y.; Zhang, K.; Li, Z.; Qiao, Y. A discriminative feature learning approach for deep face recognition. In *Computer Vision–Eccv 2016: 14th European Conference, Amsterdam, the Netherlands, October 11–14, 2016, Proceedings, Part VII*; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 499–515.
25. Yang, Y.B.; Zhong, Z.S.; Shen, T.C.; Lin, Z.C. Convolutional neural networks with alternately updated clique. In Proceedings of the 2018 IEEE/Cvf Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 2413–2422.
26. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 1 July 2016; pp. 2818–2826.

27. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the 32nd International Conference on Machine Learning, Lile, France, 6–11 July 2015; pp. 448–456.
28. Debes, C.; Merentitis, A.; Heremans, R.; Hahn, J.; Frangiadakis, N.; van Kasteren, T.; Liao, W.Z.; Bellens, R.; Pizurica, A.; Gautama, S.; et al. Hyperspectral and lidar data fusion: Outcome of the 2013 grss data fusion contest. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 2405–2418. [CrossRef]
29. Kingma, D.P.; Ba, J.L. Adam: A method for stochastic optimization. In Proceedings of the International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015; pp. 1–13.
30. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the 15th IEEE International Conference on Computer Vision, Santiago, Chile, 11–18 December 2015; pp. 1026–1034.
31. Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the 13th International Conference on Artificial Intelligence and Statistics, Sardinia, Italy, 13–15 May 2010; pp. 249–256.