applied sciences

MDPI

*Article*

# Learning Spatial–Spectral-Dimensional-Transformation-Based Features for Hyperspectral Image Classification

**Jun Wu [1,2], Xinyi Sun [1], Lei Qu [1,*], Xilan Tian [2] and Guangyu Yang [2]**

[1] Ministry of Education, Key Laboratory of Intelligent Computing and Signal Processing, Information Materials and Intelligent Sensing Laboratory of Anhui Province, School of Electronics and Information Engineering, Anhui University, Hefei 230601, China; junwu@ahu.edu.cn (J.W.)
[2] The 38th Research Institute, China Electronics Technology Group Corporation, 199 Xiangzhang Avenue, Hefei 230088, China
* Correspondence: qulei@ahu.edu.com

**Abstract:** Recently, deep learning tools have made significant progress in hyperspectral image (HSI) classification. Most of existing methods implement a patch-based classification manner which may cause training test information leakage or waste labeled information for non-central pixels of image patches. Therefore, it is challenging to achieve remarkable classification performance via the traditional convolutional neural networks (CNN) in the absence of label information. Moreover, due to the limitation of convolutional kernel sizes and convolution operations, the spectral information of HSI cannot be fully utilized with a traditional CNN framework. In this paper, we implement pixel-based classification by a special data division strategy and propose a novel spatial–spectral dimensional transformation (SSDT) to obtain spectral features containing more spectral information. Then, we construct a fully convolutional network (FCN) with two branches based on 3D-FCN and 2D-FCN to achieve broader spatial and spectral information interaction. Finally, the fused features are utilized to realize accurate pixel-based classification. We verify our proposed method on three classic publicly available datasets; the overall classification accuracy and average accuracy reach 82.27%/87.85%, 83.81%/81.55%, and 85.97%/83.89%. Compared with the latest proposed method SS3FCN in the no-information-leakage scenario, the overall classification accuracy of our proposed method is improved by 1.72%, 4.95% and 0.2%, and the average accuracy is improved by 0.95%, 3.92% and 2.67% on the three databases, respectively. Experimental results demonstrate the effectiveness of the proposed SSDT and the proposed CNN framework.

**Keywords:** hyperspectral images classification; feature extraction; spatial–spectral dimension transformation; semantic segmentation

check for updates

**Citation:** Wu, J.; Sun, X.; Qu, L.; Tian, X.; Yang, G. Learning Spatial–Spectral-Dimensional-Transformation-Based Features for Hyperspectral Image Classification. *Appl. Sci.* **2023**, *13*, 8451. https://doi.org/10.3390/app13148451

Academic Editor: Zhengjun Liu

Received: 9 June 2023
Revised: 12 July 2023
Accepted: 18 July 2023
Published: 21 July 2023

**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

## 1. Introduction

With the advent of the first aerial spectral imaging system [1], hyperspectral remote sensing images have attracted widespread attention for unique spectral information. Hundreds of contiguous and narrow spectral bands are contained in each hyperspectral image (HSI), which enable more accurate discrimination than the panchromatic and multispectral remote sensing images by taking advantage of the uniformity and wide wavelength range of hyperspectral data [2,3]. The applied technologies of hyperspectral information have been widely employed in military reconnaissance [4], exploration of natural resources, and natural disaster assessment [5].

HSI classification aims at automatically identifying the category of the target in HSI by mining discriminative features from the spatial and spectral information of them [6]. With the development of deep learning technology, the feature extraction method for HSI classification has gradually developed from the traditional hand-designed features to the automatic features based on the deep learning framework [7]. At first, the stacked autoencoder [8] and entitled deep belief network [9] were employed for HSI classification.

However, the classification accuracy of those networks was unsatisfactory when the input data was a one-dimensional vector. Then, the 1D-CNN was employed to extract the viable classification feature from the redundant spectral information. But using only one-dimensional spectral information limits the effectiveness of features such as spectral noise and other factors causing different objects to have the same spectrum or the same object to have different spectra. To solve this problem, 2D and 3D-CNNs [10] were applied to extract the spatial feature or the spectral–spatial feature simultaneously. And recently, the attention mechanism [11–13] was embedded into the classification model to select the beneficial features from redundant spectral information.

Although deep-learning-based HSI classification methods have achieved significant improvements, there are still two challenges. Firstly, the existing HSI classification methods are generally based on central pixel classification [14]. The spatial and spectral information of all pixels in the block is assigned to the central pixel, ignoring the spatial disparity and limiting the receptive field. And only a mere portion of the limited labeled pixels are utilized in the case without training test information leakage. Therefore, we turn our attention to semantic segmentation [15], which can classify each pixel of HSI. This approach not only improves the utilization of label information, but also greatly enhances the efficiency of prediction map generation. Secondly, affected by the sliding convolution mechanism of CNN and the limited size of the convolutional kernel, most of the existing CNN-based methods can only extract the adjacent spectral information and they cannot utilize the correlation information between multiple bands. However, it has been proven that the beneficial information for classification is not always continuous [1,16]. More extensive spectral information interaction between multiple bands is needed for HSI classification.

In this paper, we propose a pixel-based HSI classification method to improve the utilization of label information while achieving a more robust classification. Meanwhile, we propose a novel spatial–spectral dimensional transformation (SSDT) strategy to complete the all-channel interaction of spectral information and extend the range of information interaction of CNN-based methods. Then, we construct a fully convolutional network with two branches based on 3D-FCN and 2D-FCN which extracts joint spatial–spectral features and the novel spectral correlation features separately. Finally, we fuse two features into the classifier to obtain the classification results. The main contributions of our proposed method are summarized as follows:

(1) The pixel-to-pixel prediction strategy is utilized in the proposed method to enhance the utilization of the limited labeled sample. Meanwhile, a more reasonable data partition strategy is employed to ensure the accuracy of the sample labels, which can also prevent information leakage between the training and testing data.
(2) We propose the SSDT strategy and define the concept of the spectral self-correlation matrix by which the correlation between each spectral band within a small region is calculated one by one. Therefore, the correlation feature between each spectrum is fully utilized without being limited by the size of the convolution kernel.
(3) An FCN with two branches is constructed to implement the extraction and fusion of the proposed spectral correlation feature and classic spectral–spatial information by which the fused feature can realize the complementary advantages of the two parts. Extensive experiments show that the proposed SSDT-net achieves competitive classification performance on several HSI datasets.

## 2. Related Work

### 2.1. CNN-Based HSI Classification

Recently, the 1D, 2D, and 3D CNNs have been utilized to extract various kinds of HSI features via multiple non-linear transformation architectures [17]. Li [18] used two 2-D CNN networks to extract spatial, local spectral, and global spectral information at the same time. In addition, the SE structure [11] was applied to enhance the feature extraction capability of the two streams. Zou proposed an architecture with two FCNs for HSI classification based on 3-D and 1-D FCNs [19]. The 3-D branch extracted the spatial–

spectral information, and the 1-D branch extracted the spectral information as a supplement. However, we only extracted local spectral features and ignoreed the non-adjacent spectral features.

A complete convolutional neural network generally consists of multiple convolutional layers superimposed on each other. For each convolutional layer, the calculation process can be formulated as follows:

$$f_{ij}^{l}(x) = \sigma\left(\left(w^{l} * x\right)_{ij} + b_{l}\right), \tag{1}$$

where $w^{l}$ is the lth kernel function with the learned weights, $b_{l}$ is the bias, and $\sigma$ is the activation function.

As the depth of the network increases, the training process suffers from the problem of vanishing gradients. Therefore, the activation function is always conducted between the CNN layers to solve this problem. The Rectified Linear Unit (ReLU) is the most commonly used activation function, which is defined as

$$\sigma(x) = \max(0, x). \tag{2}$$

Another essential layer is batch normalization (BN) which is implemented to increase the convergence speed of the network and can also replace the Dropout to alleviate overfitting. The BN operation is defined as

$$\widetilde{X} = \frac{x - \mu_{B}}{\sqrt{\sigma_{B}^{2} + \epsilon}} \tag{3}$$

where $\mu_{B}$ and $\sigma_{B}$ stand for the mean and variance of x.

At the end of the classification network, a classifier is often used to convert the feature map into the corresponding probability distribution. The classifier in this paper is the Softmax function, as follows:

$$\text{Softmax}(x_{i}) = \frac{e^{x_{i}}}{\sum_{0}^{j} e^{j}}, \; i = 0, 1, 2, \ldots j - 1. \tag{4}$$

Our basic spectral full-channel interactive (SFCI) feature extraction branch comprises a convolution layer, batch normalization (BN), and a ReLU layer. The convolution layer in SFCI is conducted to extract spectral information at a long distance to make full use of the spectral information. The ReLU and BN layers are utilized to avoid overfitting.

### *2.2. Pixel-Based Classification for HSI*

In general, the performance of supervised learning methods requires much labeled training data to guarantee. However, due to the difficulty of data acquisition and labeling, the publicly available HSI data is very limited. It is a challenge to obtain accurate and robust classification methods by making reasonable use of limited public data.

Most of the existing HSI classification methods are patch-based methods that use the label of the center pixel to indicate the category of the image patch [14] as shown in Figure 1a. There are two drawbacks to these methods. Firstly, if there is no overlap in image block partitioning, only the pixel label values at the center of the image block are involved in model training, and a large amount of label information in the neighborhood is wasted. Meanwhile, the spatial information of the pixels inside the image block is ignored. These are very unfavorable to the classification results. Secondly, if there is overlap in image block partitioning, serious information leakage may occur, as shown in Figure 2. There is a part of sharing region between adjacent blocks; if these image blocks are divided into training data and test data separately, then the training data information leaks into the test data, resulting in inflated test results.

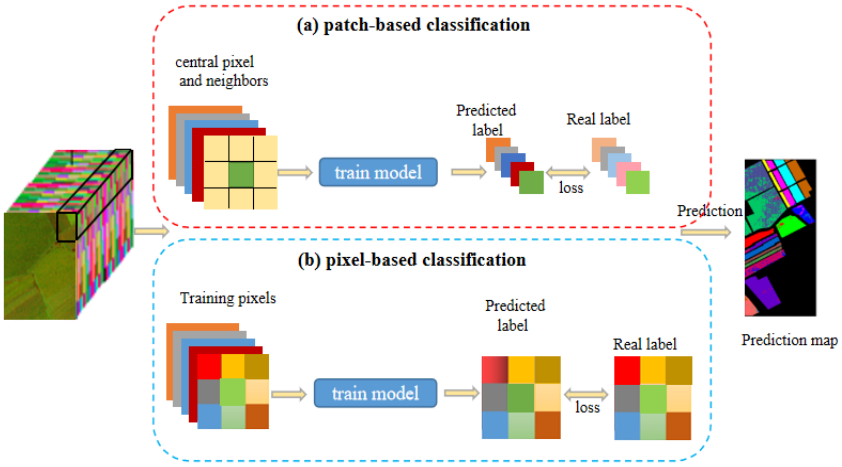



**Figure 1.** The pipelines of patch-based classification and pixel-based classification for HSI.

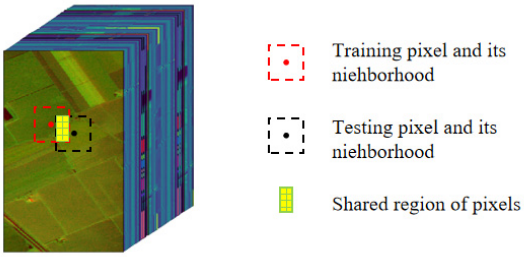



**Figure 2.** An example of the information leakage problem.

The pixel-based classification methods are inspired by semantic segmentation [15], as shown in Figure 1b. The label information of all the pixels in the image block is applied to the training and inference of the model, which effectively improves the utilization of the limited label information. Compared to patch-based classification, pixel-based classification aims at predicting the category of all pixels in the input patches, and we can easily obtain the test result map by the inference process [6]. Additionally, with a suitable data partitioning strategy, pixel-based classification can obtain a more robust and reliable classification performance.

### *2.3. Exploration of Spectral Information*

The most important characteristic of hyperspectral images is that they contain hundreds of spectral bands of information, which provide special information for target classification and identification that other kinds of images do not have. But there is a great deal of information redundancy in these highly correlated spectral bands, and some spectral bands may not carry discriminant information for classification. The methodology of exploring spectral information has progressed along with the development of HSI classification.

Extracting critical spectral bands is a primary strategy to reduce redundant information and enhance the discriminability of the spectral features. Principal component analysis (PAC) [20] is first employed to extract the main components of the spectrum. Then, the processed HSI is utilized for the subsequent classification task. In the CNN-based methods, the attention mechanism can be used to achieve the spectral selection task [21–23]. These methods demonstrate that simply selecting a dozen well-defined spectral bands is sufficient to obtain considerable results [24,25]. But directly discarding the spectral band is not conducive to the full use of spectral information [2]. Another approach is to extract the spectral information step by step using 1D convolutional kernels in the spectral dimension or 3D convolutional kernels in the spatial–spectral dimension. This approach does not miss any spectral information, but it is time-consuming and vulnerable to redundant

information. In addition, due to the limitation of the size of the convolution kernel, such methods can only extract the adjacent spectral information.

## 3. Proposed Method

The related research has proven that the reflection curves of the species are affected by the surrounding environment, light, and other factors [16,26]; in addition, different properties on different spectral bands may be exhibited [18,27,28]. Inspired by the success of covariance matrix features in the field of face classification and recognition, we assume that this correlation of spatial information among spectral bands may have considerable classification discrimination. Therefore, we try to construct a spectral self-correlation matrix to describe the correlation information and facilitate the information interaction between spectral bands.

The framework of the proposed method is shown in Figure 3. The pipeline of the framework contains three steps: data partitioning, data pre-processing, and classification network construction.

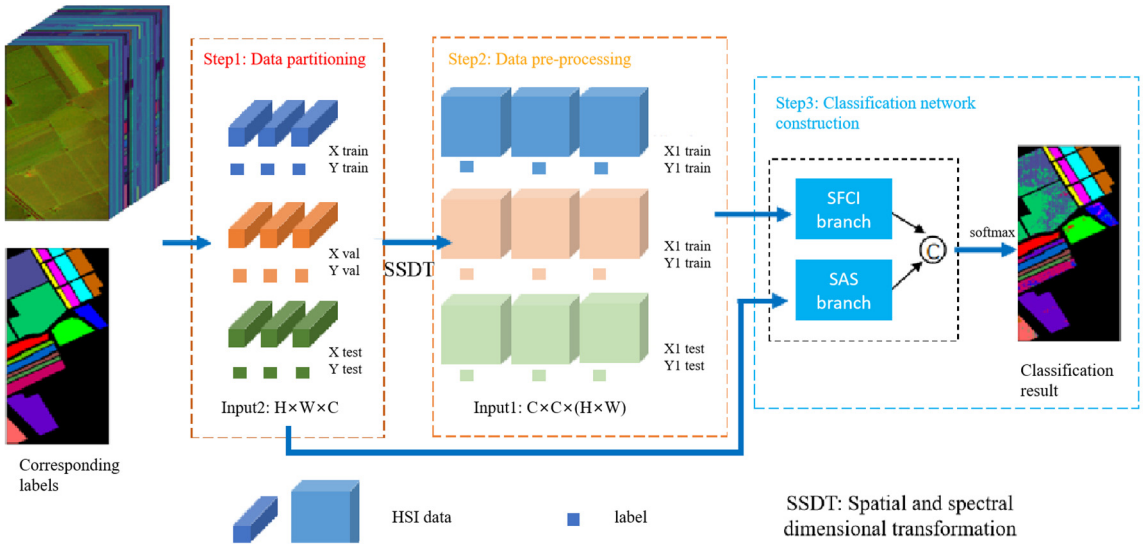



**Figure 3.** The framework of the proposed method.

### *3.1. Data Partitioning*

The purpose of data partitioning is to extract training, validation, and test data from the available HSI dataset. Most existing methods use sliding blocks by center pixel to obtain learnable data suitable for the CNN framework. However, such approaches result in overlap between the image blocks, which may lead to information leakages as exampled in Section 2.2. Recently, Nalepa et al. [29] proposed a data partitioning strategy without block overlapping. But their strategy leads to an unbalanced number of different categories in the training and test dataset because the partitioning process is random. Therefore, a reasonable partitioning strategy is needed to make the partitioned data independent of each other and to have a balanced number of categories.

To solve the problem of information leakage and unbalanced partitioning, we continue to utilize the information leakage-free strategy that we proposed in literature [19].

Firstly, we divide the whole HSI image into several blocks of the same size.1 Then, these blocks are divided into three categories: (1) the blocks consisting of background pixels entirely; (2) the blocks consisting of the same labeled pixels; (3) the blocks consisting of the different labeled pixels. We discard the first type of blocks and regard the second type of blocks as Testing set-1. The third type of blocks is divided into K groups according to the order of positions, and then the K groups are further divided into the training set, the testing set, and the validation set. For example, in the PU dataset, we divide the third type of blocks into ten groups in order, and then the order of blocks in the Mth group is [M,10 + M, ..., 10n + M],

where n denotes the number of blocks in the group. Then, we randomly select one group as the training set, one of the remaining nine groups as the validation set, and the remaining eight groups as Testing set-2. Testing set-1 (the second type of blocks) is combined with Testing set-2 as the whole testing set. Figure 4 shows that the number, class, and location of the selected datasets are not the same for different K groups at the same spatial location. The white patches represent the data that are fed into the network. This method ensures balanced data partitioning.

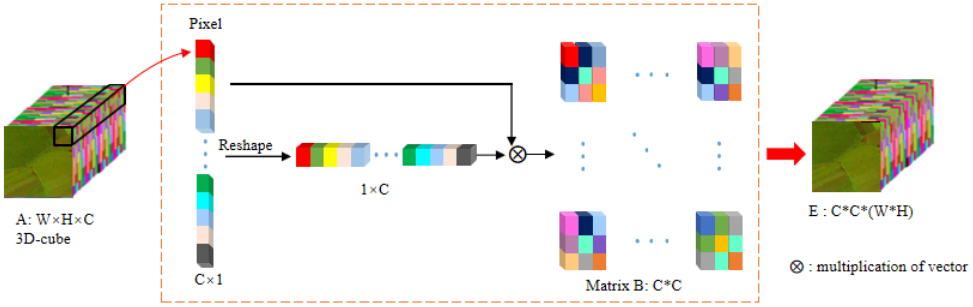



**Figure 4.** The details of the spatial and spectral dimensional transformation (SSDT).

Moreover, since the number of HSI samples is limited, we obtain more samples by applying sliding windows on these larger HSI image blocks. And we also utilize data augmentation [30–32] to increase the number of samples in the training and validation sets, such as flipping, panning, etc.

### *3.2. Data Pre-Processing*

The purpose of data pre-processing is to prepare the input data for the subsequent classification network. We conducted two branches in the proposed classification network to extract two different types of features of HSI, so we need to prepare two types of data for them in this step.

The first feature extraction branch is a 2D-FCN which can be utilized to extract our proposed spectral correlation feature. The HSI always includes three dimensions, X, Y, and Z, where X and Y are the spatial dimensions, and Z is the spectral dimension. To extract the correlation information between different spectral bands, we try to compute the self-correlation matrix of the spectral vectors at each pixel. Each element of the self-correlation matrix describes the correlation of two spectral bands at that pixel. We compute the spectral self-correlation matrix corresponding to all pixels within an image patch and stacked them together to form new image data. When we perform a convolution operation on this matrix, we can correlate information between any two spectra even if the convolution kernel is small.

The detail of SSDT is illustrated in Figure 4 given original input data A $\epsilon R^{W\times H\times C}$, where $W \times H$ refers to the spatial size and C is the number of spectral bands. For each pixel in the spatial dimensions, we can obtain a one-dimensional spectral vector $v_{(x,y)} \epsilon R^{1\times C}$ and reshape it into $v'_{(x,y)} \epsilon R^{C\times 1}$. Then, multiplication of the vector is executed between $v'_{(x,y)}$ and $v_{(x,y)}$, and we obtain a spectral self-correlation matrix (SCM) B $\epsilon R^{C\times C}$:

$$B = v'_{(x,y)} \times v_{(x,y)}, \tag{5}$$

where $\times$ is the multiplication of the vector, and the value of the matrix indicates a weighting with other spectral bands. Obviously, the matrix is symmetrically distributed, with the maximum value on the diagonal. the above operation is repeated, and we can convert the HSI data from spatial–spectral dimensions to spectral correlation spatial dimensions. The data size is converted from $W \times H \times C$ to $C \times C \times (W \times H)$. Meanwhile, as the hyperspectral data change, we also reshape the corresponding label vector into $1 \times 1 \times (W \times H)$.

The second feature extraction branch is a 3D-FCN, which can extract the spatial and spectral features simultaneously. A 3D convolutional kernel can cover the spatial and spectral dimensions at the same time, so we do not need to perform any additional operations on the input data of this branch; we can just use the data after the data partitioning. Therefore, the data size for the second branch is W × H × C, and the corresponding label size is 1 × 1 × C.

### *3.3. Classification of Network Construction*

Our proposed network is based on a fully convolutional segmentation network [15] with two branches, as shown in Figure 5. FCN transforms the fully connected layers in traditional CNNs into individual convolutional layers. Compared with the traditional method of image segmentation with CNNs, FCN has two significant advantages. Firstly, it can accept input images of arbitrary size without requiring all training and test images to have the same size. Secondly, it is more efficient because the problem of repeated storage and computation of convolution due to the use of pixel blocks is avoided.

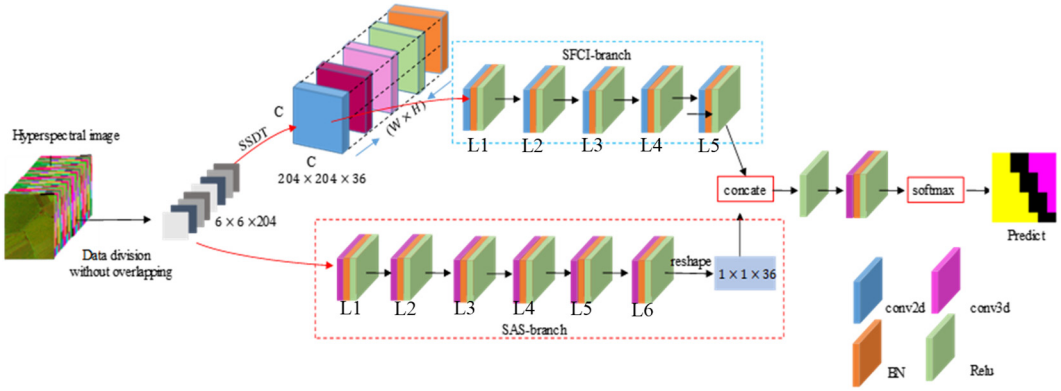



**Figure 5.** The architecture of the proposed SSDT-net.

In order to extract features in a better way, speed up the convergence speed of the network, and reduce the problem of gradient descent, our basic unit includes the convolutional layer, the batch normalization (BN) layer, and the ReLU layer. The convolutional and ReLU layers are used to hierarchically extract high-level features and improve the non-linear representation power of the network. The middle BN layer has an important role in the training procedure. Firstly, it accelerates the convergence of the network. In CNNs, there is the phenomenon of internal covariate bias. With the BN layers, the distribution of the data in each layer converges and the training is easier to converge. Secondly, it prevents gradient explosion and gradient disappearance. Thirdly, it prevents overfitting. In the training of the network, BN makes all the samples in a minibatch correlated so that the network does not generate a definite result from a particular training sample.

In Figure 5, we utilize 2D and 3D convolutions for extracting spectral, and spatial-spectral information, respectively. Then, the features extracted from the two branches are cascaded together and fed into the classification network to obtain the prediction results. The size of each layer of convolution kernel as well as its output size in Figure 5 are shown in Tables 1 and 2.

**Table 1.** The parameter details of the SFCI branch.

| Layer | Salinas Valley | | Pavia University | | Houston University | |
|---|---|---|---|---|---|---|
| | **Kernel Size** | **Output Size** | **Kernel Size** | **Output Size** | **Kernel Size** | **Output Size** |
| input | - | (204 × 204 × 36) | - | (103 × 103 × 64) | - | (144 × 144 × 36) |
| L1 | (10 × 10 × 1) | (39 × 39 × 36 × 64) | (6 × 6 × 1) | (33 × 33 × 64 × 32) | (10 × 10 × 1) | (27 × 27 × 36 × 64) |
| L2 | (3 × 3 × 1) | (20 × 20 × 36 × 64) | (3 × 3 × 1) | (17 × 17 × 64 × 64) | (3 × 3 × 1) | (14 × 14 × 36 × 64) |

**Table 1.** *Cont.*

| Layer | Salinas Valley | | Pavia University | | Houston University | |
|---|---|---|---|---|---|---|
| | **Kernel Size** | **Output Size** | **Kernel Size** | **Output Size** | **Kernel Size** | **Output Size** |
| L3 | $(3 \times 3 \times 1)$ | $(10 \times 10 \times 36 \times 128)$ | $(3 \times 3 \times 1)$ | $(9 \times 9 \times 64 \times 128)$ | $(3 \times 3 \times 1)$ | $(6 \times 6 \times 36 \times 64)$ |
| L4 | $(5 \times 5 \times 1)$ | $(5 \times 5 \times 36 \times 256)$ | $(5 \times 5 \times 1)$ | $(3 \times 3 \times 64 \times 256)$ | $(3 \times 3 \times 1)$ | $(4 \times 4 \times 36 \times 64)$ |
| L5 | $(3 \times 3 \times 1)$ | $(1 \times 1 \times 36 \times 256)$ | $(3 \times 3 \times 1)$ | $(1 \times 1 \times 64 \times 256)$ | $(4 \times 4 \times 1)$ | $(1 \times 1 \times 36 \times 64)$ |

**Table 2.** The parameter details of the SAS branch.

| Layer | Salinas Valley | | Pavia University | | Houston University | |
|---|---|---|---|---|---|---|
| | **Kernel Size** | **Output Size** | **Kernel Size** | **Output Size** | **Kernel Size** | **Output Size** |
| input | - | $(6 \times 6 \times 204)$ | - | $(8 \times 8 \times 103)$ | - | $(6 \times 6 \times 144)$ |
| L1 | $(1 \times 1 \times 10)$ | $(6 \times 6 \times 36 \times 64)$ | $(1 \times 1 \times 6)$ | $(8 \times 8 \times 33 \times 64)$ | $(10 \times 10 \times 1)$ | $(6 \times 6 \times 27 \times 64)$ |
| L2 | $(3 \times 3 \times 3)$ | $(6 \times 6 \times 20 \times 64)$ | $(3 \times 3 \times 3)$ | $(8 \times 8 \times 17 \times 64)$ | $(3 \times 3 \times 3)$ | $(6 \times 6 \times 14 \times 64)$ |
| L3 | $(3 \times 3 \times 3)$ | $(6 \times 6 \times 10 \times 128)$ | $(3 \times 3 \times 3)$ | $(8 \times 8 \times 9 \times 128)$ | $(3 \times 3 \times 3)$ | $(6 \times 6 \times 7 \times 128)$ |
| L4 | $(3 \times 3 \times 3)$ | $(6 \times 6 \times 5 \times 256)$ | $(3 \times 3 \times 3)$ | $(8 \times 8 \times 5 \times 256)$ | $(3 \times 3 \times 3)$ | $(6 \times 6 \times 4 \times 256)$ |
| L5 | $(3 \times 3 \times 3)$ | $(6 \times 6 \times 3 \times 256)$ | $(3 \times 3 \times 3)$ | $(8 \times 8 \times 3 \times 256)$ | $(3 \times 3 \times 3)$ | $(6 \times 6 \times 2 \times 256)$ |
| L6 | $(1 \times 1 \times 3)$ | $(6 \times 6 \times 1 \times 128)$ | $(1 \times 1 \times 3)$ | $(8 \times 8 \times 1 \times 256)$ | $(1 \times 1 \times 2)$ | $(6 \times 6 \times 1 \times 128)$ |
| reshape | - | $(1 \times 1 \times 36 \times 256)$ | | $(1 \times 1 \times 64 \times 512)$ | - | $(1 \times 1 \times 36 \times 256)$ |

The first branch is the spectral full-channel interaction (SFCI) branch which is used to extract the proposed spectral correlation feature. The existing research shows that the spatial features can provide complementary information to the spectral features, and the joint extraction of spectral–spatial information can significantly improve classification accuracy. Therefore, we add the spatial feature extraction (SAS) branch to extract spectral-spatial features jointly. The specific network parameters are described in Section 4.2. Finally, we fuse these two kinds of features adaptively and feed them into the classifier.

Before the training, we employ the SSDT strategy to transform raw data into C×C× (H × W) as input1 (204 × 204 × 36 for Salinas Valley). Considering the size of the input data block, we set the size of the convolution kernel to 3 × 3 or 5 × 5, etc. Since the dimensional transformation is performed in the SFCI- branch, the data of the SAS branch needs to be converted before the fusion. The size is reshaped into $1 \times 1 \times (\text{patche_size} \times \text{patche_size})$. Similarly, in the dataset processing step, the dimensions of labels are also stretched to ensure that the labels can correspond to the training, validation, and test data.

All experiments in this paper are based on focal loss [33] for model training and model selection, which is defined as

$$\text{Focal loss} = \begin{cases} -\alpha(1-\mathrm{y}\prime)^{\gamma}\log \mathrm{y}\prime, & \mathrm{y}=1 \\ -(1-\alpha)\mathrm{y}\prime^{\gamma}\log(1-\mathrm{y}\prime), & \mathrm{y}=0 \end{cases}. \tag{6}$$

The decision to use focal loss was based on reducing the impact of fewer categories on the final loss. Another advantage is it can reduce the impact of simple samples and complex samples on the classification results. It is caused by the problem that different pixels belong to the same class, but it has a different reflection curve.

After completing all configurations, we input the new data and the original data into the SFCI branch and the SAS branch, respectively. The training set is used to update the parameters for many epochs, while the validation set is used to verify the performance of the models and select the best-trained model. In the testing phase, we input the test data to the best-trained model to predict the classification accuracy.

## 4. Experimental Results and Analysis

### *4.1. Datasets Description*

To verify the accuracy of our method, three typical benchmark datasets are used in this paper.

(1) AVIRIS dataset: Salinas Valley. The image size is 512 × 217, and it contains 224 bands and 16 classes, of which 56,975 pixels are background pixels, and 54,129 pixels can be used for classification. The false-color image and ground-truth of Salinas Valley are shown in Figure 6a,b.

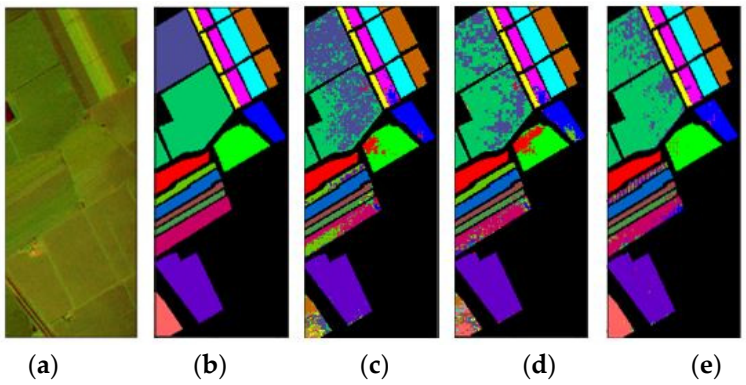



**Figure 6.** Prediction maps of different models for the Salinas Valley dataset. (**a**) False color image; (**b**) Ground truth; (**c**) 1D-CNN; (**d**) SS3FCN; (**e**) SSDT-net (OA = 83.13%, AA = 87.97%).

(2) Reflective Optics System Imaging Spectrometer (ROSIS) dataset: Pavia University. The size of the image is 610 × 340, and it contains 103 bands and nine classes. The false-color image and ground-truth of Pavia University are shown in Figure 7a,b.

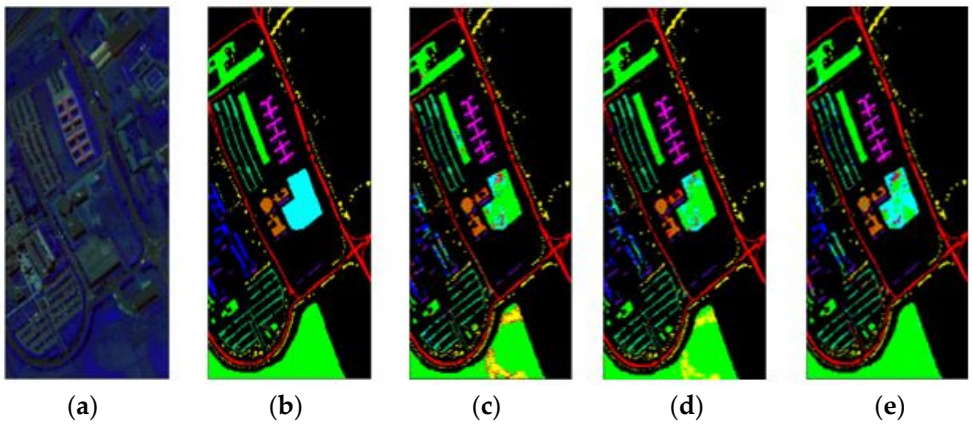



**Figure 7.** Prediction maps of different models for Pavia University dataset. (**a**) False color image; (**b**) Ground truth; (**c**) 1D-CNN; (**d**) SS3FCN; (**e**) SSDT-net (OA = 83.13%, AA = 81.55%).

(3) ITERS-CASI dataset: Houston University. It is over the University of Houston and the neighboring urban area. The data size is 349 × 1905 and contains 144 bands in the spectral range from 364 nm to 1046 nm. It has 15 classes. The false-color image and ground-truth of Houston University are shown in Figure 8a,b.

In the data division step, we continue to adopt the data partitioning rule we proposed in [19] to avoid potential information leakage [29]. We divide the HSI and labels into the same size of patches. Specifically, the patch size of Salina Valley is $6 \times 6 \times 204$, that of the Pavia University dataset is $8 \times 8 \times 103$, and that of the Houston University set is $6 \times 6 \times 144$.

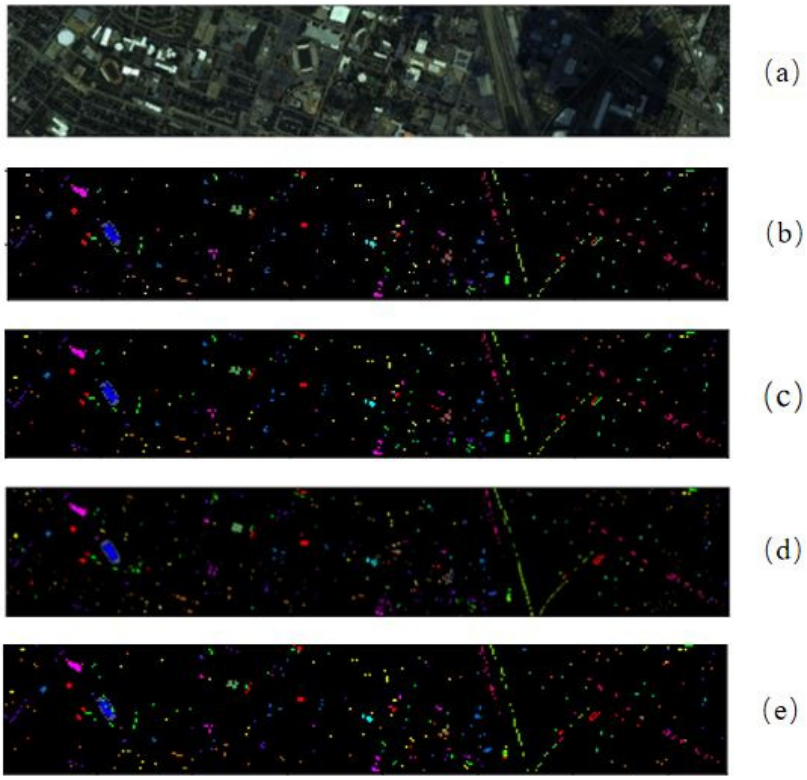



**Figure 8.** Prediction maps of different models for Houston University dataset. (**a**) False-color image; (**b**) Ground truth; (**c**) 1D-CNN; (**d**) SS3FCN; (**e**) SSDT-net (OA = 85.43%, AA = 83.55%).

### *4.2. Parameter Settings*

The HSI in three datasets is sampled into blocks. The size of blocks in the Salina Valley dataset is $6 \times 6 \times 204$, that of blocks in the Pavia University dataset is $8 \times 8 \times 103$, and that of blocks in the Houston University dataset is $6 \times 6 \times 144$. Then, the blocks in each dataset are divided into the training set, the validation set, and the test set with equal spacing and non-overlapping. In Salinas, the training set, the validation set, and the test set, respectively, account for 3.76%, 3.76%, and 92.56%. At Pavia University, the ratios are 6.64%, 6.64%, and 86.72%. At Houston University, the ratios are 18.6%, 18.6%, and 62.8%.

The above data of different proportions are input into the network for training and testing. To ensure that the number of pixels in the training set, the validation set, and the testing set is consistent with those of other methods, in Salinas, the experiment is repeated nine times with different training samples. The experiment number is ten times and six times that of the Salina for Pavia University and Houston University datasets, respectively. The computer configuration for this experiment is as follows: an i7-8700 K processor, 32 GB of RAM, and a GTX1080 GPU. The kernel size and output size of each dataset are shown in Tables 1 and 2. Taking the Salinas Valley data as an example, input1 passes through the SFCI branch to obtain 256 feature maps and 256 feature maps of the same size as the SAS branch. Then, these features are concatenated together and predicted by a softmax layer.

### *4.3. Classification Results*

The overall accuracy (OA), the average accuracy (AA), and the Kappa coefficient (K) are employed to quantitatively evaluate the classification performance. Higher values indicate better classification accuracy. To demonstrate the superiority of the techniques in this paper, we compare them with the state-of-the-art methods, including VHIS [29], 3DCNN [34], SS3FCN [19], PCA/PCA-ON [35], and GAN/PCA-ON [35].

(1) Salinas Valley Dataset: The experimental results of the Salina dataset are collected in Table 3, and the best performance indicators are bolded. Compared to other methods, as seen in Table 3, our method was confirmed to achieve the best result with OA, AA,

and Kappa of 82.24%, 87.85%, and 79.84%, respectively. Our method is consistent with that of SS3FCN [19], which is based on pixel-based classification rather than patch-based classification. Based on the SAS branch, we added the SFCI branch, and the OA, AA, and Kappa increased from 78.06%, 82.71%, and 75.31% to 82.27%, 87.85%, and 79.84%. This also confirms the role of dual branch for classification. In addition, we also clearly list the classification accuracy values of each class in Table 4, most of which are higher than those of other methods.

**Table 3.** The experimental results (%) of the Salinas Valley dataset.

| NO. | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | C9 |
|---|---|---|---|---|---|---|---|---|---|
| SSDT-net | **96.93** | **100** | **81.98** | **100** | **99.19** | **100** | **100** | **100** | **100** |
| SS3FCN [19] | 92.36 | 92.58 | 66.35 | 98.13 | 95.63 | 99.30 | 99.43 | 69.27 | 99.67 |
| VHIS [29] | 85.91 | 73.88 | 33.72 | 65.92 | 46.42 | 79.63 | 73.59 | 72.16 | 71.87 |
| PCA/PCA-ON [35] | 95.88 | 90.06 | 47.1 | 79.71 | 66.76 | 79.84 | 79.62 | 78.14 | 94.24 |
| GAN/PCA-ON [35] | 96.02 | 86.1 | 50.76 | 79.73 | 68.45 | 79.73 | 79.67 | 78.02 | 94.03 |
| 3DCNN [34] | 96.49 | 75.15 | 39.89 | 61.61 | 52 | 79.21 | 76.81 | 74.84 | 78.14 |
| **C10** | **C11** | **C12** | **C13** | **C14** | **C15** | **C16** | **OA** | **AA** | **Kappa** |
| 78.76 | 79.49 | **99.51** | **100** | **97.37** | 6 | **88.00** | **82.27 ± 1.5** | **87.85 ± 1.97** | **79.84 ± 1.56** |
| 84.07 | 85.31 | 97.98 | 98.45 | 87.32 | 52.31 | 59.97 | 81.32 | 86.13 | / |
| 73.11 | 72.51 | 71.06 | 75.80 | 72.04 | 45.03 | 22.54 | 64.20 | 64.70 | / |
| **87.65** | 73.19 | 90.75 | 98.35 | 87.57 | 57.01 | 46.11 | 76.67 | 78.25 | / |
| 85.72 | 70.95 | 89.08 | 96.93 | 85.54 | 54.66 | 43.87 | 75.87 | 77.45 | / |
| 85.69 | 71.56 | 76.49 | 80.86 | 62.15 | **61.8** | 33 | 69.72 | 96.09 | / |

**Table 4.** The experimental results (%) on the Pavia University dataset.

| Class | 3DCNN [34] | VHIS [29] | SS3FCN-VHIS | SS3FCN [19] | SSDT-net |
|---|---|---|---|---|---|
| C1 | 90.66 | 93.40 | 85.19 | 97.48 | **98.02** |
| C2 | 81.85 | 86.20 | 90.40 | 90.86 | **92.01** |
| C3 | 41.92 | 47.58 | 14.83 | **58.75** | 56.23 |
| C4 | 93.02 | 86.89 | 92.98 | 84.81 | **98.02** |
| C5 | 53.79 | 59.81 | 99.37 | 94.82 | **99.58** |
| C6 | 25.20 | 27.14 | 17.63 | 23.59 | **49.93** |
| C7 | 0 | 0 | 46.35 | 61.61 | **72.66** |
| C8 | 70.18 | 78.46 | **89.59** | 88.84 | 87.00 |
| C9 | 79.03 | 79.27 | **99.06** | 88.68 | 78.77 |
| OA | 70.07 | 73.26 | 76.23 | 79.89 | **83.81 ± 0.96** |
| AA | 60.18 | 62.08 | 70.60 | 76.60 | **81.55 ± 1.9** |
| Kappa | / | / | / | / | 73.67 ± 1.6 |

Figure 6 visualizes the false-color image of the original HSI and their corresponding ground-truth map, as well as the classification results of the different models. It can be seen from Figure 6c–e that only using spectral features for classification produces many noise points. The combination of 3D and 1D-CNN, such as SS3FCN, makes the classification effect improved. But some specific categories did not perform well in the classification, such as fallow_plow and corn weeds. Based on 3D-CNN, our method used full channel information interaction, which shows a better classification effect.

(2) Pavia University dataset: Table 4 demonstrates the experimental results of several competing methods on the Pavia University dataset, and the best performance indicators are bolded. The OA, AA, and Kappa of our proposed method are 83.81%, 81.55%, and 73.67%, respectively. Compared to the state-of-the-art comparison algorithm SS3FCN, we obtained an about 4% classification performance improvement in both OA and AA metrics. Specifically for each class of objectives in the database, we obtained optimal performance on six of the nine classes of objectives. Furthermore, we can find that the accuracy indicators

in Class 7 of 3DCNN and VHIS are both 0. In our experiment, we selected 9.1% training samples for Class 7. The correct proportion of training samples also leads to effective clarification of our proposed method.

Figure 7 visualizes the false-color image of the original HSI, their corresponding ground-truth map, and the classification results of the different models. Not only do the SSDT-net values improve in several metrics, but also the superiority of the algorithm can also be seen in Figure 7. In Figure 7b,e, it can be seen that the prediction map of SSDT-net is the most similar to the ground truth, especially for meadows. In some lower-right-corner areas, 1D-CNN and SS3FCN misjudge the category of pixels for meadows.

(3) Houston University Dataset: Compared to the other two datasets, the size of the Houston University dataset is larger (349 × 1905). But the number of labeled samples is also limited. Table 5 presents the overall classification performance and class-specific performance by different methods of classification. The SSDT-net obtains the best result, achieving 85.97%, 83.89%, and 84.76% for the three overall classification metrics. The SAS branch can even fully classify specific classes. With the same training and testing data, our method displays higher accuracy by 6.04%, 5.89%, and 6.51% than SS3FCN in terms of OA, AA, and Kappa. Figure 8 visualizes the false-color image of the original HSI, their corresponding ground-truth map, and the classification results of the different models. Our proposed method also obtained the best visual consistency with the ground truth.

**Table 5.** The result (%) of the Houston University dataset.

| NO. | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | C9 |
|---|---|---|---|---|---|---|---|---|---|
| SSDT-net | 91.37 | 87.20 | 98.97 | 93.08 | **99.28** | 91.36 | **87.08** | **66.37** | 76.59 |
| SS3FCN [19] | **94.67** | 85.45 | 98.60 | 92.90 | 96.26 | 75.97 | 86.25 | 66.37 | **78.09** |
| SS3FCN-VHIS | 90.18 | **87.78** | 98.39 | 93.87 | 94.60 | 89.63 | 77.21 | 58.71 | 73.48 |
| **C10** | **C11** | **C12** | **C13** | **C14** | **C15** | **OA** | **AA** | **K × 100** | |
| 72.46 | **80.02** | 68.51 | 54.53 | **97.14** | **95.77** | **85.97 ± 2.21** | **83.89 ± 2.53** | **84.76 ± 2.37** | |
| **81.78** | 76.29 | **68.52** | **68.74** | 94.37 | 91.04 | 83.30 | 83.69 | / | |
| 65.01 | 65.15 | 41.39 | 41.85 | 91.01 | 97.26 | 76.28 | 77.71 | / | |

(4) Overall comparison: In Table 6, we show the overall performance between our method with the other techniques that do not interact with full pixel spectral information. It includes the SAS branch, SS3FCN, 3D-CNN, VHIS, PCA/PAC-ON augmentation, and GAN/PCA-ON augmentation. Compared to single branches, dual branches obtained better performance. We suspect that the main reason is that the double branch implements the extraction and fusion of the proposed spectral correlation feature and classic spectral-spatial information. It also demonstrates that the abundant spectral information should be given more exploration in HSI classification.

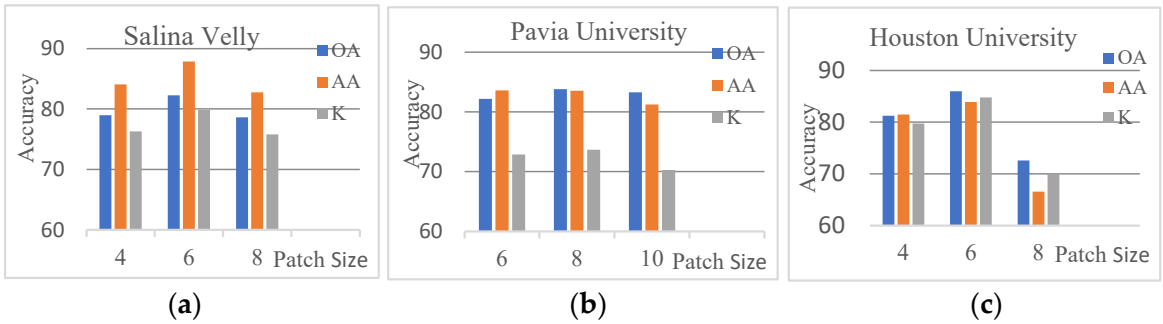
**Table 6.** The overall performance between different methods (%).

| | | SSDT | SAS Branch | SS3FCN | VHIS | SS3FCN-VHIS | PCA/PCA-ON | GAN/PCA-ON | 3DCNN |
|---|---|---|---|---|---|---|---|---|---|
| SalinaV | OA | **82.27** | 81.19 | 81.32 | 64.2 | 74.60 | 76.67 | 75.87 | 69.72 |
| | AA | **87.85** | 86.77 | 86.17 | 64.7 | 76.00 | 78.25 | 77.45 | 69.09 |
| PaviaU | OA | **83.81** | 82.2 | 79.89 | 73.26 | 76.23 | 73.84 | 71.46 | 70.07 |
| | AA | **83.55** | 80.55 | 76.6 | 62.08 | 70.6 | 62.71 | 62.83 | 60.18 |
| HoustonU | OA | **85.91** | 81.04 | 83.3 | / | 76.28 | / | / | / |
| | AA | **83.89** | 81.86 | 83.69 | / | 77.71 | / | / | / |

## 5. Discussions

### *5.1. The Impact of the Size of Input Patches*

The size of the training data is an essential factor affecting the classification performance of HSI. To verify the optimality of the input data size, we fed different size data blocks into the network separately for training; the obtained performance metrics are shown in Figure 9. It should be noted that for obtaining the probability scores corresponding to different classes, the number of kernels in the last layer should be the number of labeled classes for each dataset. In Salina Valley, the best results are achieved when the sample size is $6 \times 6$. When the size is $4 \times 4$, the OA, AA, and Kappa drop by 2.28%, 3.8%, and 3.57%, respectively. Similarly, when the size is $8 \times 8$, the three metrics decrease to 3.64%, 5.1%, and 4.06%. After several experiments on the other two datasets, the optimal training sample sizes were finally determined to be $8 \times 8$ and $6 \times 6$ (as shown in Figure 9b,c).




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

**Figure 9.** Influence of the patch sizes on OA, AA, and Kappa(K). (**a**) Salina Velly; (**b**) Pavia University; (**c**) Houston University.

### *5.2. Ablation Experiment*

To verify the effectiveness of each branch for classification, Figure 10 shows the experimental results of the single branch and the dual branch. These results illustrate the limitations of single-branch and the superiority of dual-branch fusion networks. For overall accuracy, the SAS branch reached 81.19%, 82.20%, and 83.29% on SV, PU, and HU datasets, respectively. This branch used 3D convolution to extract spatial and spectral information simultaneously. The SFCI branch realized the channel information interaction. However, because of the lack of spatial information, we selected to fuse the information of the two branches. The superiority of the two-branch network can be seen in Figure 10a, where the OA rises by 1.05%, 1.81%, and 2.68%. Similarly, for average accuracy, SSDT-net achieved better classification results.




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

**Figure 10.** The accuracy of different modules of the three datasets (SV: Salina Valley, PU: Pavia University, HU: Houston University) (**a**) Overall accuracy; (**b**) Average accuracy.

### 5.3. The Impact of the Number of Training Pixels

In addition to the above factors, the number of training pixels directly affects the performance of the entire model. In this section, we investigate the scenarios for different proportions of training samples. As we expected, the accuracy improved with an increase in the number of training samples. The result of the Salinas Valley dataset is shown in Table 7. With the rise of the proportions of training samples from 3.07% to 6.77%, the OA/AA increased from 78.02%/80.23% to 82.28%/90.74%. Similar results are obtained for the other datasets.

**Table 7.** The OA/AA with different proportions of training samples on the Salina Valley dataset.

| | **M = 5** | **M = 7** | **M = 9** | **M = 11** |
|---|---|---|---|---|
| Train pixels | **3662** | 3016 | 2839 | 1665 |
| Ratio (%) | **6.77** | 4.83 | 3.76 | 3.07 |
| OA (%) | **82.28 ± 1.66** | 81.14 ± 1.21 | 82.24 ± 1.5 | 78.02 ± 1.01 |
| AA (%) | **90.74 ± 1.32** | 88.93 ± 1.21 | 87.85 ± 1.97 | 80.23 ± 2.3 |

## 6. Conclusions

In this paper, we presented a method based on the spatial–spectral dimensional transformation for pixel-based HSI classification. The proposed method realized a spatial and spectral dimensional transformation by constructing the spectral self-correlation matrix. In this way, we could extract abundant spectral information of HSI, which is advantageous for classification, but the multiple spectral bands are farther away from each other. We refer to this combination of spectral information as the interaction of full-channel spectral information. We compared the proposed method with several classical and recently proposed classification methods on three databases and verified the availability of the proposed method.

However, there are some shortcomings in our proposed method. Firstly, due to the large number of spectral bands, the transformation of a one-dimensional spectral vector into a matrix with a huge number of parameters is more demanding on the hardware capabilities of computers. Secondly, although we computed the correlation between each spectrum through the spectral self-correlation matrix and utilized this correlation to improve the classification performance, we still did not explicitly show the correlation between this correlation and the target characteristics. Thirdly, the cascading of two branching features is too simple and does not fully utilize the complementary properties between the two features. In the future, we will make improvements to the above three issues. Firstly, we will try to construct the self-correlation matrix using the more efficient spectral information rather than utilizing all spectral information. Secondly, we will attempt to model the explicit correlation between target properties and spectral properties. Thirdly, we will promote the efficient combination of multiple types of features utilizing feature optimization.

**Author Contributions:** Conceptualization, J.W. and L.Q.; methodology, X.S.; software, J.W. and X.S.; validation, X.T. and G.Y.; writing—original draft preparation, J.W. and X.S.; writing—review and editing, J.W. and L.Q.; supervision, X.T. and G.Y.; funding acquisition, J.W., X.S. and L.Q. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Natural Science Foundation of China (61871411, 62271003, 62201008), the University Synergy Innovation Program of Anhui Province (GXXT-2021-001), and the Natural Science Foundation of Education Department of Anhui Province (KJ2021A0017), and the Scientific Research Projects for Graduate Students of Anhui Universities (YJS20210017).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** All databases used in this paper are publicly available.

**Acknowledgments:** The authors also acknowledge the high-performance Computing Platform of Anhui University and the high-performance GPU computing platform of the School of Electronic Information Engineering for providing the computing resources.

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

## References

1. Sun, W.; Du, Q. Hyperspectral band selection: A review. *IEEE Geosci. Remote Sens. Mag.* **2019**, *7*, 118–139. [CrossRef]
2. Landgrebe, D. Hyperspectral image data analysis. *IEEE Signal. Process. Mag.* **2002**, *19*, 17–28. [CrossRef]
3. Qian, Y.; Ye, M.; Zhou, J. Hyperspectral image classification based on structured sparse logistic regression and three-dimensional wavelet texture features. *IEEE Trans. Geosci. Remote Sens.* **2012**, *51*, 2276–2291. [CrossRef]
4. Du, B.; Zhang, Y.; Zhang, L. Beyond the sparsity-based target detector: A hybrid sparsity and statistics-based detector for hyperspectral images. *IEEE Trans. Image Process.* **2016**, *25*, 5345–5357. [CrossRef]
5. Zhang, B.; Wu, D.; Zhang, L. Application of hyperspectral remote sensing for environment monitoring in mining areas. *Environ. Earth Sci.* **2012**, *65*, 649–658. [CrossRef]
6. Yuan, L.; Hu, S.; Zhang, A.; Chai, S.; Wang, X. A Classification Method for Hyperspectral Imagery Based on Deep Learning. *Artif. Intell. Robot. Res.* **2017**, *06*, 31–39. (In Chinese)
7. Audebert, N.; Le Saux, B.; Lefèvre, S. Deep learning for classification of hyperspectral data: A comparative review. *IEEE Geosci. Remote Sens. Mag.* **2019**, *7*, 159–173. [CrossRef]
8. 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]
9. Chen, Y.; Zhao, X.; Jia, X. Spectral–spatial classification of hyperspectral data based on deep belief network. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2015**, *8*, 2381–2392. [CrossRef]
10. Li, Y.; Zhang, H.; Shen, Q. Spectral–spatial classification of hyperspectral imagery with 3D convolutional neural network. *Remote Sens.* **2017**, *9*, 67. [CrossRef]
11. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7132–7141.
12. Wang, L.; Peng, J.; Sun, W. Spatial–spectral squeeze-and-excitation residual network for hyperspectral image classification. *Remote Sens.* **2019**, *11*, 884. [CrossRef]
13. Roy, S.K.; Chatterjee, S.; Bhattacharyya, S. Lightweight spectral–spatial squeeze-and-excitation residual bag-of-features learning for hyperspectral classification. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 5277–5290. [CrossRef]
14. Xu, Z.; Yu, H.; Zheng, K. A Novel Classification Framework for Hyperspectral Image Classification Based on Multiscale Spectral-Spatial Convolutional Network. In Proceedings of the 11th Workshop on Hyperspectral Imaging and Signal Processing: Evolution in Remote Sensing (WHISPERS), Amsterdam, The Netherlands, 14–16 January 2021; pp. 1–5.
15. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 8–10 June 2015; pp. 3431–3440.
16. Patra, S.; Modi, P.; Bruzzone, L. Hyperspectral band selection based on rough set. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 5495–5503. [CrossRef]
17. Krizhevsky, A.; Sutskever, I.; Geoffrey, E.H. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
18. Li, X.; Ding, M.; Pižurica, A. Deep feature fusion via two-stream convolutional neural network for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2019**, *58*, 2615–2629. [CrossRef]
19. Zou, L.; Zhu, X.; Wu, C. Spectral–spatial exploration for hyperspectral image classification via the fusion of fully convolutional networks. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 659–674. [CrossRef]
20. Rodarmel, C.; Shan, J. Principal component analysis for hyperspectral image classification. *Surv. Land Inf. Sci.* **2002**, *62*, 115–122.
21. Miao, X.; Gong, P.; Swope, S. Detection of yellow starthistle through band selection and feature extraction from hyperspectral imagery. *Photogramm. Eng. Remote Sens.* **2007**, *73*, 1005–1015.
22. Chang, C.I.; Du, Q.; Sun, T.L. A joint band prioritization and band-decorrelation approach to band selection for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **1999**, *37*, 2631–2641. [CrossRef]
23. Bajcsy, P.; Groves, P. Methodology for hyperspectral band selection. *Photogramm. Eng. Remote Sens.* **2004**, *70*, 793–802. [CrossRef]
24. De Backer, S.; Kempeneers, P.; Debruyn, W. A band selection technique for spectral classification. *IEEE Geosci. Remote Sens. Lett.* **2005**, *2*, 319–323. [CrossRef]
25. Lorenzo, P.R.; Tulczyjew, L.; Marcinkiewicz, M. Hyperspectral band selection using attention-based convolutional neural networks. *IEEE Access* **2020**, *8*, 42384–42403. [CrossRef]
26. Wang, Q.; Li, Q.; Li, X. Hyperspectral band selection via adaptive subspace partition strategy. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2019**, *12*, 4940–4950. [CrossRef]
27. Li, S.; Qi, H. Sparse representation based band selection for hyperspectral images. In Proceedings of the IEEE International Conference on Image Processing, Brussels, Belgium, 11–14 September 2011; pp. 2693–2696.

28. Nalepa, J.; Myller, M.; Kawulok, M. Training-and test-time data augmentation for hyperspectral image segmentation. *IEEE Geosci. Remote Sens. Lett.* **2019**, *17*, 292–296. [CrossRef]
29. Nalepa, J.; Myller, M.; Kawulok, M. Validating hyperspectral image segmentation. *IEEE Geosci. Remote Sens. Lett.* **2019**, *16*, 1264–1268. [CrossRef]
30. Zhang, M.; Li, W.; Du, Q. Diverse region-based CNN for hyperspectral image classification. *IEEE Trans. Image Process.* **2018**, *27*, 2623–2634. [CrossRef] [PubMed]
31. Mei, S.; Ji, J.; Hou, J. Learning sensor-specific spatial-spectral features of hyperspectral images via convolutional neural networks. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 4520–4533. [CrossRef]
32. Li, W.; Wu, G.; Zhang, F. Hyperspectral image classification using deep pixel-pair features. *IEEE Trans. Geosci. Remote Sens.* **2016**, *55*, 844–853. [CrossRef]
33. Lin, T.Y.; Goyal, P.; Girshick, R. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
34. Gao, Q.; Lim, S.; Jia, X. Hyperspectral image classification using convolutional neural networks and multiple feature learning. *Remote Sens.* **2018**, *10*, 299. [CrossRef]
35. Nalepa, J.; Myller, M.; Kawulok, M. Hyperspectral data augmentation. *arXiv* **2019**, arXiv:1903.05580.


**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.