remote sensing

MDPI

Article

Convolutional Neural Networks Based Hyperspectral Image Classification Method with Adaptive Kernels

Chen Ding *, Ying Li, Yong Xia, Wei Wei, Lei Zhang and Yanning Zhang

Shaanxi Key Lab of Speech & Image Information Processing (SAIIP), School of Computer Science and Engineering, Northwestern Polytechnical University, Xi'an 710129, China; lybyp@nwpu.edu.cn (Y.L.); yxia@nwpu.edu.cn (Y.X.); weiweinwpu@nwpu.edu.cn (W.W.); zhanglei211@mail.nwpu.edu.cn (L.Z.); ynzhang@nwpu.edu.cn (Y.Z.)

* Correspondence: dingchen@mail.nwpu.edu.cn; Tel.: +86-159-0299-1949

Academic Editors: Qi Wang, Nicolas H. Younan, Carlos López-Martínez and Prasad S. Thenkabail
Received: 10 May 2017; Accepted: 14 June 2017; Published: 16 June 2017

Abstract: Hyperspectral image (HSI) classification aims at assigning each pixel a pre-defined class label, which underpins lots of vision related applications, such as remote sensing, mineral exploration and ground object identification, etc. Lots of classification methods thus have been proposed for better hyperspectral imagery interpretation. Witnessing the success of convolutional neural networks (CNNs) in the traditional images based classification tasks, plenty of efforts have been made to leverage CNNs to improve HSI classification. An advanced CNNs architecture uses the kernels generated from the clustering method, such as a K-means network uses K-means to generate the kernels. However, the above methods are often obtained heuristically (e.g., the number of kernels should be assigned manually), and how to data-adaptively determine the number of convolutional kernels (i.e., filters), and thus generate the kernels that better represent the data, are seldom studied in existing CNNs based HSI classification methods. In this study, we propose a new CNNs based HSI classification method where the convolutional kernels can be automatically learned from the data through clustering without knowing the cluster number. With those data-adaptive kernels, the proposed CNNs method achieves better classification results. Experimental results from the datasets demonstrate the effectiveness of the proposed method.

Keywords: hyperspectral image classification; automatic cluster number determination; adaptive convolutional kernels

1. Introduction

Different from traditional images (e.g., RGB image), hyperspectral image (HSI) contains a continuous spectrum at each pixel, which is beneficial for identifying different imaged land covers. With such abundant spectral information, hyperspectral image (HSI) classification that aims at assigning each pixel a pre-defined class label has facilitated various applications, such as mineral exploration, ground object identification, survey of agriculture and monitoring of geology, etc. Therefore, plenty of efforts have been made in HSI classification. According to the feature utilized, HSI classification methods can be roughly divided into hand-crafted feature based methods and the deep learning feature based methods. A detailed review can be seen from Section 2. For hand-crafted feature based methods, HSI is often represented by the features designed manually [1–7]. However, due to their shallow structure, the representation ability of such features is limited, especially for HSIs which often exhibit high nonlinearity aroused by the high-dimensionality and mixture of pixels. On the contrary, deep learning feature based methods can automatically extract features from training data with deep architectures. It has been proved that those deep features perform well in representing the complicated nonlinearity of data, which has promoted the development of deep learning feature based HSI classification methods in recent years [8–12].

Since the convolutional kernels should be updated through the network training, traditional deep learning based methods exhaust much training time. To address this problem, an advanced CNNs architecture has been proposed recently, which adopts the kernels pre-learned from clustering the training data without updating them in the training process any more. One typical method is the K-means Net proposed in [13], where each CNNs kernel is first learned from a specific cluster obtained by conducting the K-means algorithm on training data. Nevertheless, the cluster number K (i.e., the number of kernels in CNNs) of K-means Net should be assigned empirically, which limits the representational power of CNNs. Specifically, a different number K of kernels designed manually in the convolutional layer will change the structure of CNNs and thus influence the output of CNNs. In addition, the number K is expected to be adaptive to different images and tasks. Therefore, how to data-adaptively choose a proper number of kernels is crucial for representing data characteristics with CNNs. However, most of the existing CNNs based HSI classification methods fail to pay sufficient consideration to this problem.

In this study, we propose a MCFSFDP based CNNs framework for HSI classification. First, inspired by clustering by fast search and find of peaks (CFSFDP) [14], a novel clustering method, named modified clustering by fast search and find of peaks (MCFSFDP), is proposed to data-adaptively learn a specific number of kernels from training data. The convolution kernels can be automatically determined by the center of each cluster and the inter-cluster margin, which guarantees the pre-learned kernels to be suitable for the data structure. Then, the CNNs framework with those pre-learned convolutional kernels is employed to classify each pixel in the HSI. Extensive experimental results demonstrate that the proposed method outperforms several state-of-the-art CNNs based methods in classification accuracy.

In summary, the proposed CNNs framework has two key advantages: (1) a specific number of convolutional kernels can be data-adaptively learned from training data, which can well represent the data characteristics; and (2) the MCFSFDP based CNNs framework is effective for HSI classification.

2. Related Work

Based on the feature adopted in classification of HSI, the HSI classification method can be roughly divided into two categories, including the hand-crafted feature based methods and the deep learning feature based methods.

2.1. Hand-Crafted Feature Based Methods

Linear features extracted by principal component analysis (PCA) [15] and partial least squares (PLS) [16] are applied to classify the HSI data. The kernel methods are further developed to exploit the nonlinear feature of HSI [17]. To depict the spatial texture of image, the wavelet transform (WT) methods [18,19] have been widely used, which often show different scales and perform effectively for classification in the high spatial resolution remotely sensed (HSRRS) data. Considering the complicated spatial correlation, some Gaussian Markov Random Field (GMRF) [20,21] methods are proposed to model such correlation within a graph structure. In [22], a spatial feature index that measured the gray similarity distance in every direction is used to describe the shape feature in local area that is surrounding a pixel in HSI. An adaptive mean-shift (MS) analysis framework [2] is proposed for object extraction and classification of HSI over urban areas, which is able to obtain an object-oriented representations of HSI data. Li et al. [3] integrate the spectral and spatial information in a Bayesian framework, which utilizes a Multinomial Logistic Regression (MLR) algorithm to learn the posterior probability distributions from the spectral information. In addition, this method uses subspace projection to better characterize noise, highly mixed pixels and contextual information. In [4], a mathematical morphology (MM) based method is utilized to process the HSI data. In this approach, opening and closing morphological transforms are used to isolate bright (opening) and dark (closing) structures in images, where bright/dark means brighter/darker than the surrounding features in the images. To model different kinds of structural information, morphological attribute profiles (APs) are adopted to provide a multi-level characterization for an image created by the sequential application of morphological attribute filters [23]. Based on Gray Level Co-occurrence matrix (GLCM), Zortea et al. attempt to extract the contextual information of images by concatenating the spectral features used for classification [1]. To improve the classification result of HSI, the Edge-Aware Filtering (EAF) and Edge-Preserving Filtering (EPF) methods are proposed in [24,25]. Based on the EPF method, a spectral-spatial classification framework was proposed in [25], which can significantly enhance the classification accuracy. Kang et al. propose combining a recursion with image fusion to enhance the image classification accuracy [26]. Recently, the Bag-of-Words (BOW) model has shown a promising way to handle the remote sensing imagery classification problem. In the BOW model, images can be represented by the frequency of visual words that are constructed by quantizing local features with a clustering method, such as K-means and so on [27,28]. Due to the capacity of extracting the handcrafted local features, such as local structural points, color histogram and texture features [29,30], BOW based methods present good performance. Manifold regularized kernel logistic regression (KLR) are proposed to solve multi-view image classification [31]. To integrate different levels of features for saliency detection, Wang et al. [32] propose a multiple-instance learning based framework that fuses the low-level, mid-level, and high-level features into a unified model. While effective, the trepresentation capacity of the manual feature extraction based methods is limited.

2.2. Deep Learning Feature Based Methods

Recently, with the development of deep learning technology, lots of methods based on deep learning have been developed for image classification, such as deep brief network (DBN) and stacked auto-encoder (SAE). The DBN and SAE are unsupervised learning methods that are also used for spectral-spatial classification of hyperspectral data without using the label information [9,33]. The concept of deep learning is introduced into the hyperspectral data classification for the first time [9]. The Canonical Correlation Analysis Network is useful for multi-view image classification [34]. With the development of convolutional neural networks (CNNs) [35], which has been widely applied to the image processing and achieved spectacular effects, more and more deep CNNs frameworks have emerged, such as AlexNet [36], VGGNet [37], GoogLeNet [38] and ResNet [39], which can provide results comparable with human beings in image classification and recognition tasks. Those methods can automatically learn features from the training data, which can replace the manually-engineered features, and have shown significant effects on HSI classification [8–10]. For example, Li et al. [40] applied 3D-CNNs for spectral-spatial feature extraction and classification, where 3D kernels were used to extract the feature from HSI cube without any preprocessing or post-processing. In [41], the transfer learning method for HRRS scene classification is used for transferring features from successfully pre-learned CNNs. Different from the CNNs methods, the convolutional kernels are updated in the training process, and the kernels in PCA-Net [42] and K-means Net are pre-learned before the network training and don't need to be updated in the network training. In addition, the kernels come from data directly. PCA-Net [42] adopts the principle components of training data as multistage filter banks, while K-means Net learns the kernels by clustering the training data. In this study, we mainly focus on the K-means Net. Although K-means Net can be directly applied to the classification and reduces the training time by employing the pre-learned kernels, it is difficult to determine the number of kernels that is crucial for the performance. To address this issue, we attempt to adaptively generate a specific number of kernels from the training data of CNNs framework.

3. MCFSFDP Based CNNs

The traditional CNNs framework contains the convolutional layer, fully connected layer and a classification layer. The convolution layer is updated through the error feedback process, which is different from the pre-learned convolutional kernels based CNNs framework.

The proposed MCFSFDP based CNNs method includes three major modules: (1) data pre-processing module, which extracts patches from block samples; (2) MCFSFDP based kernel learning module, which learns the convolutional kernels from those extracted patches; and (3) classification modules which utilize the learned convolution kernels.

The flowchart of our MCFSFDP based CNNs method is shown in Figure 1.

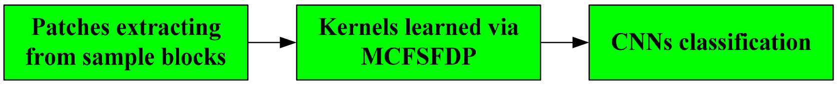

Patches extracting from sample blocks
Kernels learned via MCFSFDP
CNNs classification

Figure 1. The flowchart of the MCFSFDP based CNNs method.

3.1. Data Pre-Processing

In this study, we follow the standard data pre-processing principle in K-means Net [13]. Specifically, a HSI used in this classification task is denoted by $\boldsymbol{R}$. Though HSI is 3D data, it also can be seen as a collection of 2D images (i.e., images from different bands). Here, we denote the HSI as 2D form. First, we randomly select M pixels from $\boldsymbol{R}$, and then extract M corresponding blocks $\{B_i\}_{i=1}^{M}$ with a size of $m \times m$ as samples, where each block is centered at each selected pixel. These extracted M samples are roughly divided into three parts, namely, training samples, validation samples and testing samples. The property of center block pixel is described by all the pixels in the block. Then, $\{B_i\}_{i=1}^{M}$ are put into the network and the center pixel labels of block B_i are used as the ground truth for training.

In addition, we randomly extract N patches $\{P_j\}_{j=1}^{N}$ with a size of $n \times n$ from M_T training samples, M_T denotes the number of training samples, where $M_T < M$ and $n < m$. The extracted N patches $\{P_j\}_{j=1}^{N}$ are used for learning the convolutional kernels with a size of $n \times n$ via MCFSFDP. The producing process of the block (sample) and patch is shown in Figure 2.

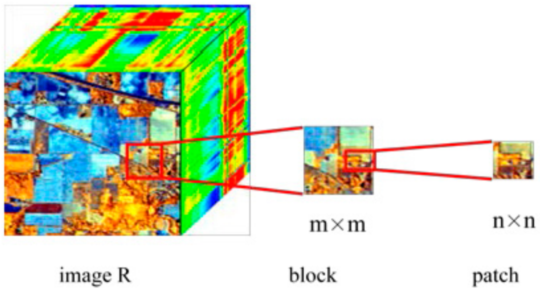


Figure 2. The block (sample) is extracted from image R and the patch is extracted from block, respectively.

3.2. MCFSFDP Based CNNs Kernels Learning

To obtain the kernels with those cropped patches, a suitable clustering method is necessary. Lots of clustering methods have been proposed, among which clustering by fast search and find of peaks (CFSFDP) [14], is a typical state-of-the-art method. The reason for partial success of CFSFDP on clustering is based on the idea that "cluster centers are characterized by a higher density than their neighbors and by a relatively large distance from points with higher densities" and the cluster centers can be determined through two thresholds of distance and density [14].

Though CFSFDP has shown its power for clustering, we find that when we apply it directly to generate the kernels for CNNs, the generated kernels are not always optimal for hyperspectral image classification tasks. This phenomena is observed from the experimental results (a similar conclusion also can be seen from the results in Section 4.3.1). In our opinion, we consider kernels (filters) as the standards for comparing the samples, which also show the evaluation standards for determining which cluster they belong to. Since the inter-cluster points are difficult to classify, we should also select several inter-cluster points with representations as the clusters (kernels). To address this problem, we propose a new clustering method based on CFSFDP, which only uses distance threshold to generate the kernel centers. The proposed method differs from the traditional CFSFDP in two aspects: (1) CFSFDP simultaneously uses the points with a large distance and high density to determine the cluster center, which easily excludes the outlier points into the generation of cluster centers; while the proposed MCFSFDP method only uses distance threshold to generate the cluster center, the cluster centers can be generated from either outlier points (with only large distance) or points of density; (2) the number of clusters via CFSFDP is determined 'semi-automatically', i.e., an extra frame needs to be introduced to help determine the number of clusters, while the number of clusters can be automatically determined through the proposed method. We give the details of the proposed method as follows.

The same as the CFSFDP algorithm in [14], we assume that the cluster centers are characterized by a higher density than their neighbors and by a relatively large distance from points with higher densities.

Following this idea, we firstly reshape each patch P_j into a column vector as a data point j with a size of $1 \times n^2$. For each point j, we compute two values: its local density ρ_j and its distance δ_j from the point with higher density, where, if the point j has the highest density, δ_j denotes the largest distance between j and other points.

Both of these values depend only on the Euclidean distances d_{jk} between any pair of data points j and k. The local density ρ_j of data j is defined as

$$\rho_j = \sum_k \chi(d_{jk} - d_c), \tag{1}$$

where $\chi(x) = 1$ if $x < 0$ and $\chi(x) = 0$ otherwise, and d_c is a cut-off distance. Basically, ρ_j is equal to the number of points that are closer than d_c to point j. δ_j is evaluated through computing the minimum distance between the point j and any other point with higher density in Equation (2):

$$\delta_j = \min_{k:\rho_k > \rho_j} (d_{jk}) \,. \tag{2}$$

For the point with the highest density, we usually take $\delta_j = \max_k(d_{jk})$. Note that δ_j is much larger than the typical nearest neighbor distance only for points that are local or global maxima in the density. Thus, the cluster centers are recognized as points for which the value of δ_j is anomalously large and the value of ρ_j is higher than a value density at the same time. To show the distance and density of each point intuitively, we give the decision graph of 10,000 patches with a size of 10 × 10 from the real Indian pines dataset in Figure 3.

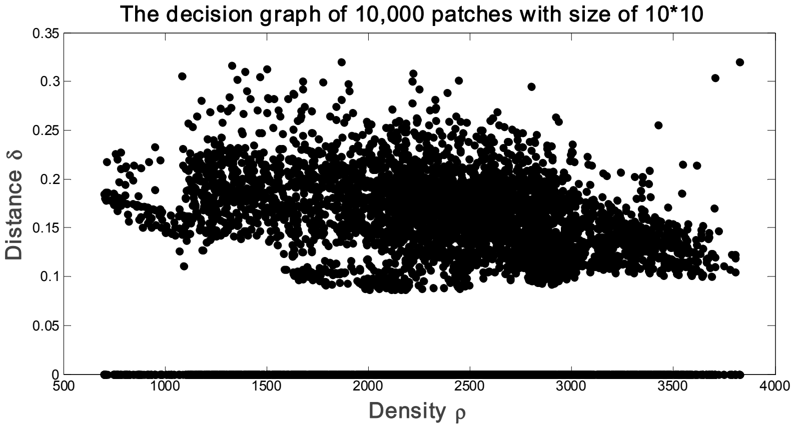


Figure 3. Decision graph of 10,000 patches with a size of 10 × 10 on the Indian pines dataset.

Different from choosing cluster centers in CFSFDP [14], we use the MCFSFDP algorithm to learn the kernels adaptively. Firstly, we choose the distance δ as the only threshold for choosing kernels from the decision graph in MCFSFDP.

To adapt the kernels and choose the number of kernels, we select the optimal distance threshold value δ_A as the following steps:

$$num_v = f(\delta_v), \tag{3}$$

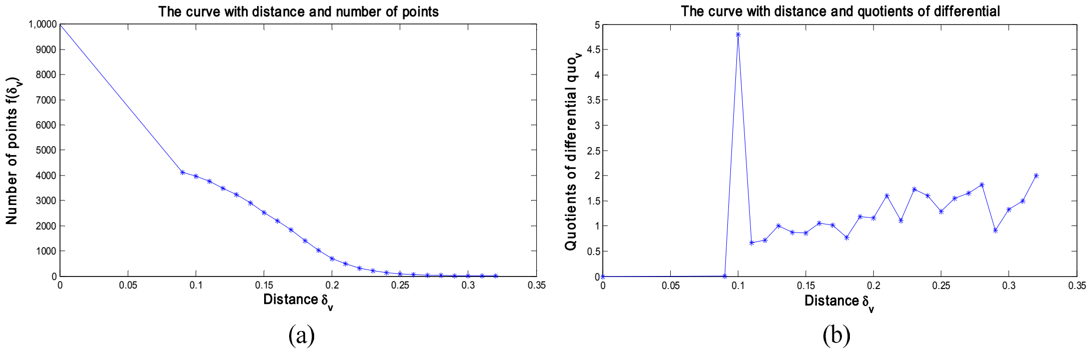
$$con_v = [f(\delta_{v+1}) - f(\delta_v)]/(\delta_{v+1} - \delta_v), \tag{4}$$

$$quo_v = |con_v / con_{v+1}|. \tag{5}$$

where, in Equation (3), δ_v denotes the value of distance that contains points and $f(\delta_v)$ gives the mapping relationship of the number of points whose distances are equal or larger than δ_v, as shown as Figure 4a. In Equation (4), where $\delta_{v+1} \geq \delta_v$, con_v denotes the differential of $f(\delta_v)$, which is an intermediate result between Equations (3) and (5). Equation (5) denotes the variation quantity of the number of points with δ_v, shown as Figure 4b.

δ_A denotes the adaptive distance threshold, and the points whose distances are larger than δ_A are chosen as CNN kernels. δ_A is a critical point that must satisfy the number num_v and num_{v+1} of points are stable (in other words, they have a similar quantity), at the same time, the value $|con_v/con_{v+1}|$ is larger than the value $|con_{v+1}/con_{v+2}|$. In this time, δ_v is selected as the adaptive distance threshold δ_A.

In other words, to determine the adaptive distance threshold δ_A intuitively, from Figure 4a, we can find the value region δ_v (0.25–0.30) from curve 1 when num_v begins to approach to 0; as can be seen from Figure 4b, con_v with the distance value δ_v in region (0.25–0.30) has a local maxima at $\delta_v = 0.28$. The distance δ_v (0.28) that belongs to the region (0.25–0.30) is confirmed as the adaptive threshold distance as δ_A. In conclusion, by observing Figure 4, the adaptive distance threshold δ_A is determined as 0.28 on the Indian Pines dataset.



(a) (b)

Figure 4. The curve for determining the adaptive distance with patches with a size of 10 × 10 the on Indian pines dataset. (**a**) shows the curve of point-number over distance δ_v; (**b**) gives the curve of quotients of differential over distance δ_v.

Finally, the points j with the distance value $\delta_j > \delta_A$ are adaptively chosen as the kernels and thus the number of kernels is also adaptively determined through the threshold δ_A. Those chosen points are then reshaped to patches with a size of $n \times n$ as the convolutional kernels in the CNNs framework. The CNNs with the pre-learned adaptive kernels are called MCFSFDP Net. The pre-learned kernels are denoted as w^k in the following sections.

3.3. Convolutional Neural Networks

With the pre-learned kernels w^k, a convolutional neural network such as [13] is designed for per-pixel level HSI classification. This CNNs structure consists of an input layer, a convolutional layer, a pooling layer, a fully connected layer and a soft-max layer, as shown in Figure 5.

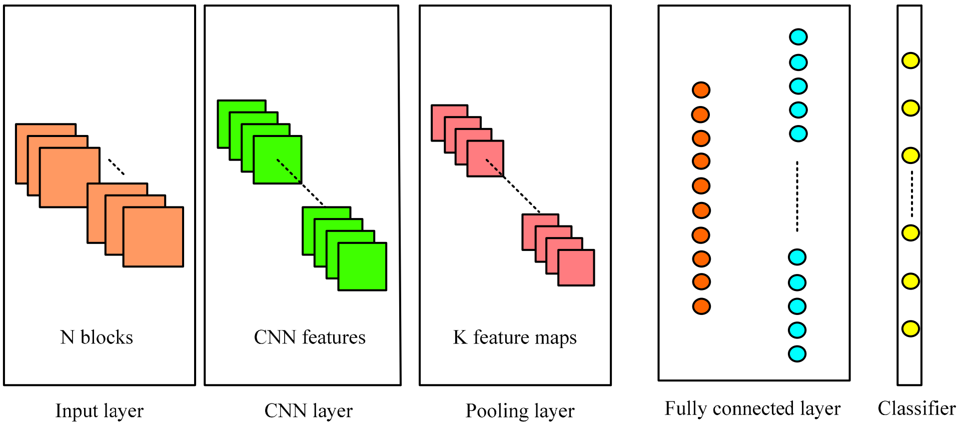


Figure 5. The structure of MCFSFDP based CNNs.

There are k kernels in the convolutional layer. Each feature map is calculated by taking the dot product between the k-th kernel w^k of size $n \times n$, $w \in R^{n\times n\times k}$, and local context area x of size $m \times m$ with c number of channels, $x \in R^{m\times m\times c}$. The feature map corresponding with the k-th filter $f \in R^{(m-n+1)\times(m-n+1)}$ is calculated as:

$$f_{ij}^k = \sigma(\sum_c \sum_{a=0}^{n-1} \sum_{b=0}^{n-1} w_{abc}^k x_{i+a,j+b}^c), \tag{6}$$

where σ is the rectified linear unit (ReLU). The kernels were pre-trained using the MCFSFDP algorithm.

The maximum pooling over a local non-overlapping spatial region is adopted to down-sample the convolutional layer. The pooling layer for the k-th filter, $g \in R^{(m-n+1)/p\times(m-n+1)/p}$, is calculated as:

$$g_{ij}^k = \max(f_{1+p(i-1),1+p(j-1)}^k, \cdots, f_{pi,1+p(j-1)}^k, \cdots, f_{1+p(i-1),pj}^k, \cdots, f_{1+pi,pj}^k). \tag{7}$$

The k feature maps are reshaped to the column vectors and all the column vectors are connected with a fully connected auto-encode unit. The autoencode unit is used to process the connected column vector and represented the feature of the column vector. The output results of the hide layer in the auto-encode unit were used to connect the classification layer.

The last CNNs step is a soft-max layer used for final classification.

4. Experiments and Analysis

Three datasets were utilized to validate the feasibility and effectiveness of the proposed CNNs based MCFSFDP method (named as MCFSFDP Net) in HSI classification. In the following sections, dataset and experimental settings are described firstly, and then the effectiveness and the superiority of the proposed method are tested.

4.1. Datasets

To find images with less categories and obvious discriminations between categories, we firstly select an image dataset with a size of 256 × 256. The image of this dataset has been manually labeled as three categories, including mountains, sky and roads. One hundred samples with a size of 25 × 25 from each category that were extracted from this image. We randomly choose 210 context area samples for training, 30 samples for validation and 60 other samples for testing. The details of selected image samples were given in Table 1.

Table 1. Ground truth classes and their respective sample numbers in Dataset 1.

Class	Samples		
	Training	Validation	Testing
Mountain	70	10	20
Sky	70	10	20
Road	70	10	20

In order to evaluate the proposed method on complex data, Dataset 2 includes the benchmark Indian Pines image, which is HSI data captured by the airborne visible imaging spectrometer (AVIRIS) sensor with a moderate spatial resolution of 20 m over the Indian Pines test site in northwestern Indiana in 1992. As shown in Figure 6, this image contains145 × 145 pixels and 224 spectral bands, whose wavelength ranges from 0.4 to 2.5 um. The number of bands of corrected data was reduced to 200 (extracted the 1–200 bands). In addition, 6476 image context area samples with a size of 19 × 19 were extracted. Among them, 3238, 647 and 2591 samples were used for training, validation and testing, respectively. The details of each category of image samples were given in Table 2.

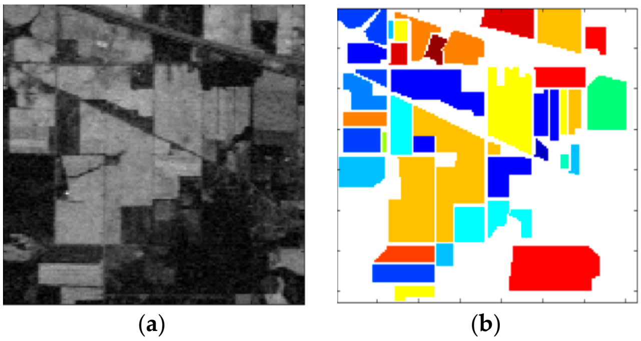


Figure 6. The Indian Pines on Dataset 2. (**a**) shows the composite image; (**b**) shows the groundtruth of Indian Pines dataset, where the white area denotes the unlabeled pixels.

Table 2. Groundtruth of classes and their respective sample numbers on Indian Pines scene.

Class		Samples			
Number	Classes	Total	Training	Validation	Testing
1	Alfalfa	46	23	4	19
2	Corn-notill	1288	636	132	520
3	Corn-mintill	63	29	7	27
4	Corn	35	17	3	15
5	Grass-pasture	180	90	14	76
6	Grass-trees	730	342	84	304
7	Grass-pasture-mowed	28	16	1	11
8	Hay-windrowed	94	45	8	41
9	Oats	20	10	2	8
10	Soybean-notill	807	406	71	330
11	Soybean-mintill	2067	1019	215	833
12	Soybean-clean	227	124	22	81
13	Wheat	204	107	28	69
14	Woods	560	307	44	209
15	Buildings-Grass-Trees-Drives	73	38	9	26
16	Stone-Steel-Towers	54	29	3	22
	Total	**6476**	**3238**	**647**	**2591**

The third Dataset 3 includes the benchmark Pavia University image, which is HSI data captured by a ROSIS sensor with a moderate spatial resolution of 1.3 m over the flight campaign over Pavia, northern Italy. As shown in Figure 7, this image contains 610 × 610 pixels and 103 spectral bands. The number of bands was reduced to 100 (extracted the 1–100 bands). Furthermore, 34,400 image context area samples with a size of 11 × 11 were extracted. Among them, 17,200, 3440 and 13,760 samples were used for training, validation and testing, respectively. The details of each category of samples were given in Table 3.

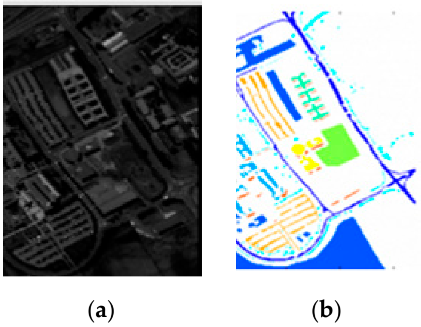


Figure 7. The Pavia University in Dataset 3. (**a**) shows the composite image; (**b**) shows the groundtruth of the Pavia University dataset, white area denotes the unlabeled pixels.

Table 3. Groundtruth of classes and their respective sample numbers in the Pavia University scene.

Class		**Samples**			
Number	**Classes**	**Total**	**Training**	**Validation**	**Testing**
1	Asphalt	5446	2718	580	2148
2	Meadows	12,695	6307	1320	5068
3	Gravel	1314	674	126	514
4	Trees	2709	1329	241	1139
5	Painted metal sheets	1345	688	153	504
6	Bare Soil	5029	2517	453	2059
7	Bitumen	1330	686	120	524
8	Self-Blocking Bricks	3630	1810	362	1458
9	Shadows	902	471	85	346
	Total	**34,400**	**17,200**	**3440**	**13,760**

4.2. Experimental Parameter Settings

Ten thousand patches were randomly extracted from the training samples for learning kernels. For each dataset, the sample (blocks) size and the number of patches should be maintained consistently in different pre-learned CNNs frameworks.

The CNNs framework that is shown in Figure 5 uses one convolutional layer, one pooling layer, one auto-encode layer and a classifier. In our algorithm, the pooling layer adopted the non-overlap rule, the number of neurons in the hide layer of auto encode was set to 100 and the maximum iterations for training the classifier was 400. The learning rate is 0.0001 and momentum is 1. The batch sizes on the three datasets are chosen as 10, 50 and 200, respectively.

The codes are running on the computer with Intel Xeon E5-2678 V3 2.50 GHz × 2 (Intel, Santa Clara, CA, USA), NVIDIA Tesla (NVIDIA, Santa Clara, CA, USA) K40c GPU × 2, 128 GB RAM, 120 GB SSD and Matlab 2016a (MathWorks, Natick, MA, USA). The gradient is computed via batch gradient descent, which is not computed by GPU.

The average test accuracy is calculated on 10 independent Monte Carlo runs.

4.3. Experimental Results

4.3.1. Effectiveness of the Kernels Learned by MCFSFDP

The aim of this experiment is to validate the effectiveness of the kernels learned by MCFSFDP. To this end, we compared those kernels with those learned as the cluster center obtained by CFSFDP algorithm. Those two kinds of kernels were then integrated into the same CNNs framework for HSI classification on Dataset 1. To obtain fair comparison results, both of the numbers of kernels in those two methods were fixed at 49. The kernel size was set to 14 × 14 and the pooling size was designed as 4 × 4. The average testing classification accuracy of those two methods was shown in Table 4.

Table 4. The testing accuracy compared with learned 49 kernels via CFSFDP and MCFSFDP-M on Dataset 1.

Methods	CFSFDP Net	MCFSFDP Net
Accuracy (%)	81.67 ± 0.5904	**95.00 ± 0.5887**

It reveals that the kernels learned by the MCFSFDP are more effective than the kernels learned by the CFSFDP.

4.3.2. Effectiveness of the Kernels Number Determined by MCFSFDP

To demonstrate the effectiveness of the kernels number determined by MCFSFDP, we compared MCFSFDP with its variants for classification in each dataset. Those variants shared the same CNNs architecture and the kernel learning scheme excepted choosing the kernels number manually. Dataset 1, Dataset 2 and Dataset 3 were used in the experiment. For each dataset, the kernel size and the pooling size can be found in Table 5.

Table 5. The chosen block size, kernel size and pooling size of each dataset.

Dataset	Dataset 1	Dataset 2	Dataset 3
Block Size	25 × 25	19 × 19	11 × 11
Kernel Size	10 × 10	6 × 6	2 × 2
Pooling Size	4 × 4	7 × 7	2 × 2

We report the testing classification accuracy of all these methods on each dataset in Tables 6–8, respectively. Each variant is denoted as MCFSFDP-M Net followed with a specific number which indicates the kernel number chosen manually. Similarly, the number that followed MCFSFDP Net represents the kernel number automatically determined by the proposed method.

Table 6. The testing accuracy of MCFSFDP-M Net compared with MCFSFDP Net on Dataset 1.

Methods	MCFSFDP-M Net-20	MCFSFDP-M Net-25	MCFSFDP-M Net-41	MCFSFDP-M Net-55	MCFSFDP Net-35
Accuracy (%)	93.33 ± 0.5887	95.00 ± 0.5904	95.00 ± 0.5904	95.00 ± 0.5904	**96.67 ± 0.5887**
Distance threshold	0.19	0.18	0.16	0.15	**0.17**
Number of kernels	20	25	41	55	**35**

Table 7. The testing accuracy of MCFSFDP-M Net compared with MCFSFDP Net on Dataset 2.

Methods	MCFSFDP-M Net-14	MCFSFDP-M Net-24	MCFSFDP-M Net-31	MCFSFDP-M Net-83	MCFSFDP-M Net-151	MCFSFDP Net-50
Accuracy (%)	95.29 ± 0.0870	96.51 ± 0.4146	97.03 ± 0.1940	97.07 ± 0.3434	96.82 ± 0.1457	**97.84 ± 0.2249**
Distance threshold	0.27	0.26	0.25	0.23	0.22	**0.24**
Number of kernels	14	24	31	83	151	**50**

Table 8. The test accuracy of MCFSFDP-M Net compared with MCFSFDP Net on Dataset 3.

Methods	MCFSFDP-M Net-19	MCFSFDP-M Net-42	MCFSFDP-M Net-152	MCFSFDP Net-78
Accuracy (%)	88.98 ± 0.2651	89.32 ± 0.1908	89.54 ± 0.1002	**90.58** ± **0.1477**
Distance threshold	0.08	0.07	0.05	**0.06**
Number of kernels	19	42	152	**78**

In Table 6, the proposed method determines the kernel number as 35. The manually chosen kernel number in other variants are 20, 25, 41 and 55, respectively. The accuracy, distance threshold and the number of kernels for each method are shown in different rows. It can be seen that the proposed method shows the best classification accuracy. Similar phenomenon arises in Tables 7 and 8. Therefore, we can conclude that the proposed method is able to seek a good kernel number for different datasets.

4.3.3. Performance Evaluation of MCFSFDP Net

In this part, the proposed method was compared with three state-of-the-art pre-learned kernels based CNNs methods, including K-means Net [13], PCA-Net [42] and Random Net. For fair comparison, the same CNNs architecture was adopted by all comparison methods. The number of kernels for K-means Net, PCA-Net and Random Net was set to 50, while the proposed method determines the number of kernels automatically. For each dataset, the kernel size and the pooling size can be found in Table 9.

Table 9. The testing accuracy of different CNNs methods compared with MCFSFDP Net on Dataset 1.

Methods	K-Means Net-50	PCA Net-50	Random Net-50	MCFSFDP Net-35
Accuracy (%)	93.33 ± 0.5887	90.00 ± 1.8175	95.00 ± 1.8175	**96.67** ± **0.5887**

It reveals that the proposed algorithm can produce more accuracy for pixel classification than those three types of pre-learned kernels based CNNs methods on this dataset as shown in Table 9. Moreover, the proposed MCFSFDP Net with 35 kernels that has less computational complexity than comparison methods with 50 kernels in the training process.

The average testing classification accuracy of our proposed algorithm, K-means Net, PCA-Net and Random Net on Dataset 2 was given in Table 10. The results obviously show that the proposed MCFSFDP Net obtains better accuracy than those three types of pre-learned kernels based CNNs methods, which is consistent with the results obtained from Dataset 1.

Table 10. The testing accuracy of different CNNs methods compared with MCFSFDP Net on Dataset 2.

Methods	K-Means Net-50	PCA Net-50	Random Net-50	MCFSFDP Net-50
Accuracy (%)	95.02 ± 0.3343	97.30 ± 1.1916	97.12 ± 0.6195	**97.84** ± **0.2249**

The average classification accuracy of our proposed method compared with another three kernels pre-learned based CNNs on the Pavia University image was presented in Table 11. The results show that our proposed CNNs method is more accurate than those three types of pre-learned kernels based CNNs methods. Even if the proposed method needs more kernels number to perform the better classification result.

Table 11. The testing accuracy of different CNNs methods compared with MCFSFDP Net on Dataset 3.

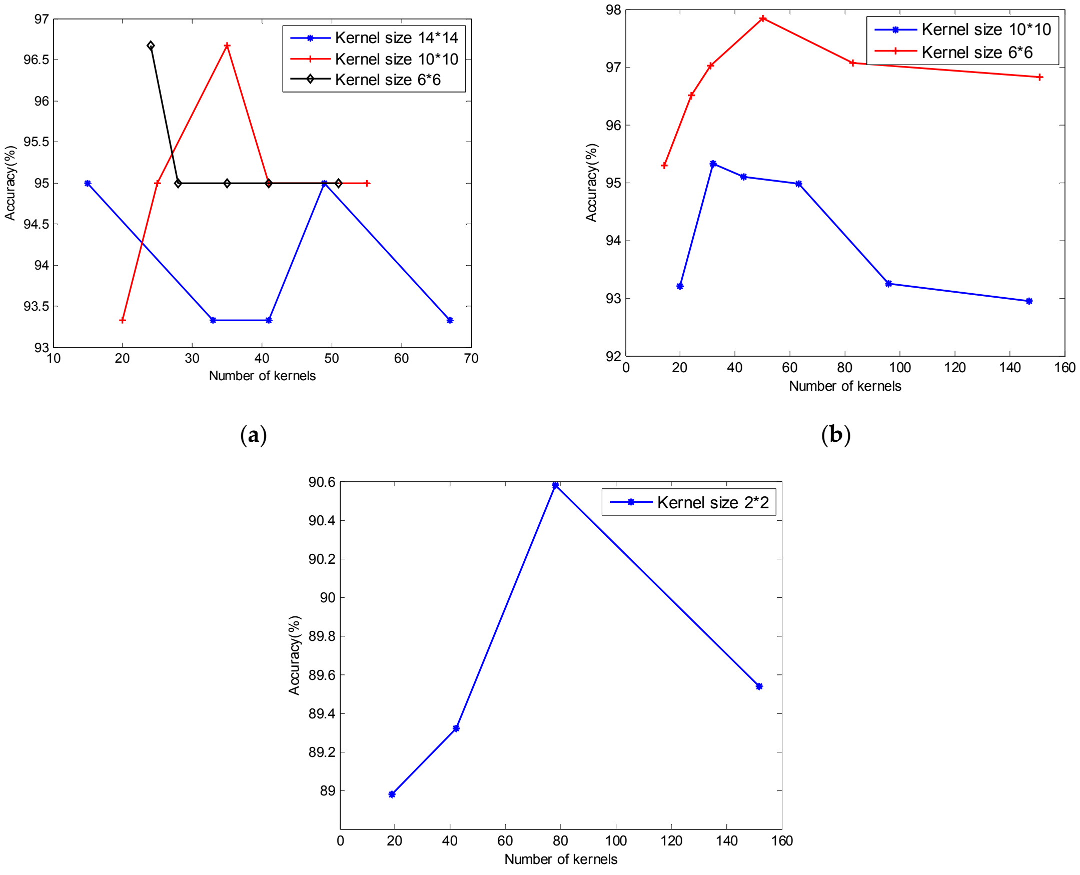
Methods	K-Means Net-50	PCA Net-50	Random Net-50	MCFSFDP Net-78
Accuracy (%)	89.77 ± 0.3399	90.14 ± 0.2652	90.47 ± 0.5113	**90.58** ± **0.1477**

5. Discussion

5.1. Effect ofthe Number of Kernels

In the MCFSFDP-M Net, the number of kernels influences the pixel-level classification. Figure 8 shows the classification accuracy achieved with different numbers A_k that were manually selected via MCFSFDP on Dataset 1, Dataset 2 and Dataset 3.

Figure 8a shows the classification results with the variation of kernel numbers A_k on each kernel size $n \times n$ on Dataset 1. The accuracy of MCFSFDP-M Net computation cannot be enhanced when the kernel number A_k was increased. Figure 8b shows the highest accuracy on Dataset 2. While the kernel number is manually chosen via MCFSFDP, the accuracy can get a high point in the number range of the kernels, as the adaptive kernels learned through the MCFSFDP method. It demonstrates again that the accuracy cannot be enhanced with the increased kernel number on Dataset 3, as shown in Figure 8c.



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

(**c**)

Figure 8. The classification accuracy influence with the number of kernels. (**a**) the classification accuracy with the increased number of kernels with different kernel size on Dataset 1; (**b**) the classification accuracy with the increased number of kernels with different kernel size on Dataset 2; (**c**) the classification accuracy with the increased number of kernels on Dataset 3.

5.2. Effect of the Kernel Size

In our proposed MCFSFSP based CNN method, the kernel size has a major impact on the pixel classification performance. Table 12 gives the average classification accuracy obtained by using different kernel size. It shows that the highest classification accuracy was achieved when kernel size was set to 10 × 10 and 6 × 6 on Dataset 1 and 6 × 6 on Dataset 2.

Table 12. The average classification accuracy obtained by using different kernel size.

Dataset	Dataset 1			Dataset 2	
Pooling Size	4 × 4	4 × 4	4 × 4	5 × 5	7 × 7
Kernel Size	14 × 14	10 × 10	6 × 6	10 × 10	6 × 6
Number of Kernels	15	35	24	32	50
Distance Value	0.22	0.17	0.17	0.28	0.24
Accuracy (%)	95	**96.67**	**96.67**	95.33	**97.84**

6. Conclusions

In this paper, we propose a novel CNNs classification framework for HSIs, which can data-adaptively learn a specific number of kernels from the training data. In particular, this model adopts the MCFSFDP algorithm to cluster the training data, and then the convolutional kernels can be determined automatically by the cluster center and inter-cluster margin. With those pre-learned kernels, a CNNs framework is developed for classifications. We have compared the proposed CNNs framework against three state-of-the-art deep learning methods with pre-trained kernels on three datasets. The experimental results demonstrate the superiority of the proposed CNNs framework in classification accuracy. Moreover, we validate that the proposed method is able to seek a good kernel number for a specific dataset. These adaptively learned kernels can help us understand the complexity of data and adjust the CNNs architecture for good feature extraction.

In terms of future research, we will exploit a multi-layer architecture via MCSFDP based CNNs to enhance the classification accuracy with less samples.

Acknowledgments: This work was supported by the Key Project of the National Natural Science Foundation of China (Grant No. 61231016), the National Natural Science Foundations of China (Grant No. 61471297, Grant No. 61671385 and Grant No. 61301192) and the China 863 Program (Grant No. 2015AA016402).

Author Contributions: All of the authors made significant contributions to this work. Chen Ding and Yanning Zhang devised the approach and analyzed the data; Yong Xia, Wei Wei, Lei Zhang and Ying Li helped design the remote sensing experiments and provided advice for the preparation and revision of the work; Chen Ding performed the experiments.

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

References

1. Zortea, M.; Martino, M.D.; Serpico, S. A SVM ensemble approach for spectral-contextual classification of optical high spatial resolution imagery. In Proceedings of the 2007 IEEE International Geoscience and Remote Sensing Symposium, Barcelona, Spain, 23–28 July 2007; pp. 1489–1492.
2. Huang, X.; Zhang, L. An Adaptive Mean-Shift Analysis Approach for Object Extraction and Classification From Urban Hyperspectral Imagery. *IEEE Trans. Geosci. Remote Sens.* **2008**, *46*, 4173–4185. [CrossRef]
3. Li, J.; Bioucas-Dias, J.M.; Plaza, A. Spectral-Spatial Hyperspectral Image Segmentation Using Subspace Multinomial Logistic Regression and Markov Random Fields. *IEEE Trans. Geosci. Remote Sens.* **2012**, *50*, 809–823. [CrossRef]
4. Benediktsson, J.A.; Palmason, J.A.; Sveinsson, J.R. Classification of hyperspectral data from urban areas based on extended morphological profiles. *IEEE Trans. Geosci. Remote Sens.* **2005**, *43*, 480–491. [CrossRef]
5. Wei, W.; Zhang, Y.; Tian, C. Latent subclass learning-based unsupervised ensemble feature extraction method for hyperspectral image classification. *Remote Sens. Lett.* **2015**, *6*, 257–266. [CrossRef]
6. Zhang, L.; Wei, W.; Tian, C.; Li, F.; Zhang, Y. Exploring Structured Sparsity by a Reweighted Laplace Prior for Hyperspectral Compressive Sensing. *IEEE Trans. Image Process.* **2016**, *25*, 4974–4988. [CrossRef]

7. Zhang, L.; Wei, W.; Zhang, Y.; Shen, C.; van den Hengel, A.; Shi, Q. Dictionary learning for promoting structured sparsity in hyperspectral compressive sensing. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 7223–7235. [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.; 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]
10. Zhao, W.; Du, S. Spectral-Spatial Feature Extraction for Hyperspectral Image Classification: A Dimension Reduction and Deep Learning Approach. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 4544–4554. [CrossRef]
11. Wang, Q.; Lin, J.; Yuan, Y. Salient Band Selection for Hyperspectral Image Classification via Manifold Ranking. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, *27*, 1279. [CrossRef] [PubMed]
12. Wang, Q.; Yuan, Y.; Yan, P. Visual Saliency by Selective Contrast. *IEEE Trans. Circuits Syst. Video Technol.* **2013**, *23*, 1150–1155. [CrossRef]
13. Längkvist, M.; Kiselev, A.; Alirezaie, M.; Loutfi, A. Classification and Segmentation of Satellite Orthoimagery Using Convolutional Neural Networks. *Remote Sens.* **2016**, *8*, 329. [CrossRef]
14. Rodriguez, A.; Laio, A. Clustering by fast search and find of density peaks. *Science* **2014**, *344*, 1492–1496. [CrossRef] [PubMed]
15. Timmerman, M.E. Principal Component Analysis (2nd ed.) by I.T. Jolliffe. *J. Am. Stat. Assoc.* **2003**, *98*, 1082–1083. [CrossRef]
16. Rosipal, R.; Krämer, N. Overview and recent advances in partial least squares. In *Subspace, Latent Structure and Feature Selection, Proceedings of the Statistical and Optimization Perspectives Workshop (SLSFS 2005), Bohinj, Slovenia, 23–25 February 2005*; Springer: Berlin/Heidelber, Germany, 2006; pp. 34–51.
17. Camps-Valls, G.; Bruzzone, L. *Kernel Methods for Remote Sensing Data Analysis*; John Wiley & Sons: River Street Hoboken, NJ, USA, 2009.
18. Myint, S.W. Wavelets for Urban Spatial Feature Discrimination: Comparisons with Fractal, Spatial Autocorrelation, and Spatial Co-occurrence Approaches. *Photogramm. Eng. Remote Sens.* **2004**, *70*, 803–812. [CrossRef]
19. Zhu, C.; Yang, X. Study of remote sensing image texture analysis and classification using wavelet. *Int. J. Remote Sens.* **1998**, *19*, 3197–3203. [CrossRef]
20. Dong, Y.; Forester, B.C.; Milne, A.K. Segmentation of radar imagery using the Gaussian Markov random field model. *Int. J. Remote Sens.* **1999**, *20*, 1617–1639. [CrossRef]
21. Dong, Y.; Forster, B.C.; Milne, A.K. Comparison of radar image segmentation by Gaussian-and Gamma-Markov random field models. *Int. J. Remote Sens.* **2003**, *24*, 711–722. [CrossRef]
22. Zhang, L.; Huang, X.; Huang, B.; Li, P. A pixel shape index coupled with spectral information for classification of high spatial resolution remotely sensed imagery. *IEEE Trans. Geosci. Remote Sens.* **2006**, *44*, 2950–2961. [CrossRef]
23. Dalla Mura, M.; Benediktsson, J.A.; Waske, B.; Bruzzone, L. Morphological Attribute Profiles for the Analysis of Very High Resolution Images. *IEEE Trans. Geosci. Remote Sens.* **2010**, *48*, 3747–3762. [CrossRef]
24. He, K.; Sun, J.; Tang, X. Guided image filtering. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *35*, 1397.
25. Kang, X.; Li, S.; Benediktsson, J.A. Spectral-Spatial Hyperspectral Image Classification With Edge-Preserving Filtering. *IEEE Trans. Geosci. Remote Sens.* **2014**, *52*, 2666–2677.
26. Kang, X.; Li, S.; Benediktsson, J.A. Feature Extraction of Hyperspectral Images With Image Fusion and Recursive Filtering. *IEEE Trans. Geosci. Remote Sens.* **2014**, *52*, 3742–3752.
27. Sivic, J.; Zisserman, A. Video Google: A Text Retrieval Approach to Object Matching in Videos. In Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV), Nice, France, 13–16 October 2003; pp. 1470–1477.
28. Lazebnik, S.; Schmid, C.; Ponce, J. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, New York, NY, USA, 17–22 June 2006; pp. 2169–2178.
29. Xia, G.S.; Delon, J.; Gousseau, Y. Accurate Junction Detection and Characterization in Natural Images. *Int. J. Comput. Vis.* **2014**, *106*, 31–56.
30. Xia, G.S.; Delon, J.; Gousseau, Y. Shape-based Invariant Texture Indexing. *Int. J. Comput. Vis.* **2010**, *88*, 382–403.

31. Liu, W.; Liu, H.; Tao, D.; Wang, Y.; Lu, K. Manifold regularized kernel logistic regression for web image annotation. *Neurocomputing* **2016**, *172*, 3–8.
32. Wang, Q.; Yuan, Y.; Yan, P.; Li, X. Saliency Detection by Multiple-Instance Learning. *IEEE Trans. Cybern.* **2013**, *43*, 660–672. [PubMed]
33. 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.
34. Yang, X.; Liu, W.; Tao, D.; Cheng, J. Canonical Correlation Analysis Networks for Two-view Image Recognition. *Inf. Sci.* **2017**, *385–386*, 338–352.
35. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324.
36. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25, Proceedings of the Neural Information Processing Systems, Stateline, NV, USA, 3–8 December 2012*; Neural Information Processing Systems Foundation, Inc.: La Jolla, CA, USA, 2012; pp. 1097–2013.
37. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv*, **2014**, arXiv:1409.1556.
38. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 8–10 June 2015; pp. 1–9.
39. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.
40. Li, Y.; Zhang, H.; Shen, Q. Spectral-Spatial Classification of Hyperspectral Imagery with 3D Convolutional Neural Network. *Remote Sens.* **2017**, *9*, 67. [CrossRef]
41. Hu, F.; Xia, G.S.; Hu, J.; Zhang, L. Transferring Deep Convolutional Neural Networks for the Scene Classification of High-Resolution Remote Sensing Imagery. *Remote Sens.* **2015**, *7*, 14680–14707. [CrossRef]
42. Chan, T.H.; Jia, K.; Gao, S.; Lu, J.; Zeng, Z.; Ma, Y. PCANet: A Simple Deep Learning Baseline for Image Classification? *IEEE Trans. Image Process.* **2015**, *24*, 5017–5032. [CrossRef] [PubMed]