remote sensing

MDPI

Article

Automatic Kernel Size Determination for Deep Neural Networks Based Hyperspectral Image Classification

Chen Ding *, Ying Li, Yong Xia, 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.); zhanglei211@mail.nwpu.edu.cn (L.Z.); ynzhang@nwpu.edu.cn (Y.Z.)

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

Received: 6 December 2017; Accepted: 6 March 2018; Published: 8 March 2018

Abstract: Considering kernels in Convolutional Neural Networks (CNNs) as detectors for local patterns, K-means neural network proposes to cluster local patches extracted from training images and then fixate those kernels as the representative patches in each cluster without further training. Thus the amount of labeled samples necessitated for training can be greatly reduced. One key property of those kernels is their spatial size which determines their capacity in detecting local patterns and is expected to be task-specific. However, most of literatures determine the spatial size of those kernels in a heuristic way. To address this problem, we propose to automatically determine the kernel size in order to better adapt the K-means neural network for hyperspectral imagery classification. Specifically, a novel kernel-size determination scheme is developed by measuring the clustering performance of local patches with different sizes. With the kernel of determined size, more discriminative local patterns can be detected in the hyperspectral imagery, with which the classification performance of K-means neural network can be obviously improved. Experimental results on two datasets demonstrate the effectiveness of the proposed method.

Keywords: hyperspectral imagery; Convolutional Neural Networks; K-means; pre-learned kernels; automatic kernel size determination

1. Introduction

With the growth of remote sensing technology, hyperspectral imagery (HSI) which can provides both spatial information and abundant spectral information [1,2], has been widely employed in various applications such as mineral exploration, ground object identification, survey of agriculture, monitoring of geology, etc. In these applications, pixel level classification is a commonly used technology, which is crucial for both the low-level HSI processing and the high-level understanding of HSI.

Plenty of methods have been proposed for HSI classification. According to the feature utilized for representing pixels in a HSI, they often can be roughly divided into two categories, namely handcrafted feature based methods and deep learning feature based methods. In previous years, handcrafted feature based methods for HSI classification had gained much promising progress. Nevertheless, for the handcrafted feature based methods [3–9], various domain knowledge is required in order to extract the appropriate features for the following classification step. More importantly, the handcrafted features are often exhibit shallow structure, and thus are insufficient to represent the many complicated structures in the challenging HSI classification problems. Recently, deep learning feature based methods are extensively investigated, which can learn features from low level to high level with a deep hierarchical structure. Compared with those handcrafted features, the learned deep features often show better nonlinear representation ability for the original images. Therefore, numerous deep learning feature based methods have been developed for HSI classification [10–12].

Deep belief network (DBN) [13] and stacked auto-encoder (SAE) [14] are two widely used unsupervised deep learning methods. Given the learned deep features, an appropriate classifier can be further trained to process the spectral-spatial classification of hyperspectral data [11,15]. Different from SAE and DBN, convolutional neural networks (CNNs) turn to learn deep features from extensive labeled data, which have shown their advantage for traditional image classification problems. To date, many deep CNNs frameworks have been developed for RGB image classification, e.g., AlexNet, VGGNet, GoogLeNet and ResNet [16–21]. Some of them have been employed for HSI classification [22,23] and gained promising results. Nevertheless, the methods based on traditional CNNs need intricate networks structure and extensive time for networks training [16–21].

To mitigate this problem, some newly proposed CNNs methods commence at employing the pre-learned convolutional kernels to reduce the computational cost as well as training examples necessitated for updating the convolutional kernels. These pre-learned convolutional kernels based methods include PCA-Net [24], MCFSFDP-Net [25], Random Net and K-means Net [26]. For example, in [27], an approach for per-pixel classification of satellite data using CNNs is proposed. It firstly pre-learns CNNs kernels with an unsupervised clustering algorithm, e.g., K-means algorithm. Given those pre-learned kernels, only the classifier is trained with the back propagation scheme for per-pixel classification, object recognition and segmentation [26,27]. In [24], the principal components analysis (PCA) algorithm is combined with the support vector machine (SVM) classifier to build a network for original image classification. Nevertheless, the parameter of dimension reduction in PCA algorithm needs to be empirically determined. In [25], an modified clustering by fast search and find of density peaks (MCFSFDP) method is proposed to data-adaptively determine the number of pre-learned kernels. In both [25,26], the pre-learned convolutional kernels of CNNs framework are generated via clustering patches, which are randomly extracted from the original image. However, the important parameter, kernel size, still has to be determined empirically. How to data-adaptively determine the size of convolutional kernels in the pre-learned convolutional kernels based CNNs framework is seldom studied in recent researches. In recent years, some methods based on large scale computation for adaptively determining the CNNs architecture are emerged. In [28], a shape driven kernel adaptation in convolutional neural network method is proposed to explore how the shape information can be explicitly deployed into the popular CNN architecture to disentangle irrelevant non-rigid appearance variations in recognition tasks. Actually, this method adopts different adaptation functions to replace the single function in the CNN architecture. In addition, acquiring optimal parameter of networks is also a difficult task. Although in [29], to solve the problem of kernel size designing in CNNs framework, an evolutionary algorithms (EA) based method is proposed. However, this method needs much calculated quantity and hyper-parameters.

Since the convolutional kernels in both K-means Net and MCFSFDP-Net are learned through the clustering algorithm, the classification results are relying on the quality of the clustering. Better clustering results (i.e., better pre-learned kernels in the pre-learned CNNs framework) result in better features. The evaluation indicators of the clustering results can be divided into two categories, those for the samples without labels and those for the samples with given labels [30–33]. In the first category, it includes Compactness (CP), Separation (SP), Davies-Bouldin Index (DBI) and Dunn Validity Index (DVI). CP means the average distance between the points in one class and the center point in the same class, which only considers the effect of inner-class. SP is the average distance between two different center points. In addition, SP also only considers inner-class elements. The DBI is not suitable to evaluate the results of samples with the circular distribution, according to use the Euclidean distance. DVI represents the effectiveness for discrete points, however, it has poor effect for the samples with circular distribution. Generally speaking, this category of methods for evaluating the samples with given labels is just for the supervised clustering. However, most of the clustering problems are unsupervised. In the other category for non-labeled data, the indicators, such as Cluster Accuracy (CA), Rand Index (RI) and Normalized Mutual Information (NMI), are used for evaluating clustering results.

In [26], K-means Net utilizes K-means algorithm to learn kernels, which is an unsupervised clustering method and based on distance of the sample points. For this reason, the evaluation indicator for measuring the clustering results should be relevant to either inter-class or inner-class distance. Due to different kernel sizes, the same clustering sample with different size owns different location when it projects into 2-D plane, and the number of samples in each class is always different, which belongs to the problem of samples with non-uniform distribution. In this evaluation task, the traditional evaluation indicator based on either inter-class or inner-class distance is not suitable. To better deal with this evaluation problem, a more practical evaluation indicator should be designed to replace the recent unsuitable determination methods. What' more, the new practical evaluation indicator needs to consider the important factor of number of samples after clustering process in each class.

In this paper, to enhance the HSI classification results of K-means Net, we propose a new size-adaptive kernels based K-means Net. Specifically, a new clustering evaluation indicator for the groups of pre-learned kernels with different sizes is proposed to evaluate the clustering results and determines the adaptive kernel size. Using the proposed method, the adaptive kernel size can be easily determined to well represent the data characteristics. Experimental results on two datasets demonstrate that with the automatically determined kernel size, the proposed method outperforms several state-of-the-art CNNs methods.

In summary, the proposed CNNs framework has two key contributions: (1) a specific size of convolutional kernels can be determined by a new clustering evaluation indicator; (2) the K-means based CNNs framework with adaptive kernel size is effective for HSI classification.

2. The Proposed Method

The K-means based CNNs method with adaptive kernel size includes four major steps: (1) data pre-processing which extracts groups of patches with different patch sizes from block samples (the block samples are extracted from the original HSI for training); (2) K-means for clustering the convolutional kernels with groups of different sizes; (3) the evaluation of clustering results for determining the adaptive kernel size; and (4) HSI classification using the pre-learned kernels with adaptive kernel size in K-means based CNNs. The flowchart of the proposed method is shown as Figure 1.

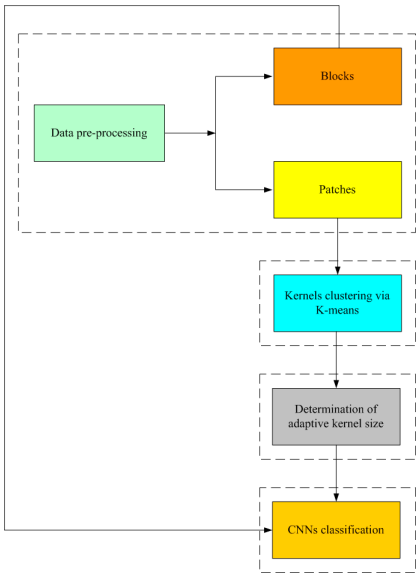


Figure 1. The flow chart of our proposed method.

2.1. Data Pre-Processing

For simplicity, we denote the HSI employed for classification as R in this paper.

First, we randomly select M pixels from R and then extract the corresponding blocks $\{B_i\}_{i=1}^{M}$ with size of $m \times m$ centered at each selected pixel as samples. Each pixel contains the information from all spectral bands, here, we omit the bands in the process of describing the size. These extracted M samples are roughly divided into three sets, namely training samples set (M_T), validation samples set (M_V) and testing samples set (M_P). The label of the central block pixel is represented through the property of the whole block. In other words, the property of the central pixel is described via the statistical property of pixels value which includes the central pixel and its surrounding pixels in each whole block. Then, $\{B_i\}_{i=1}^{M_T}$ are fed into the network and the central pixels labels of the input blocks $\{B_i\}_{i=1}^{M_T}$ are used as the ground truth for training a network. In this paper, through comparing different block sizes, we select the block with size of 27×27 (i.e., $m = 27$) to obtain the best classification results [23].

Moreover, we randomly extract N patches $\{P_j\}_{j=1}^{N}$ with a size of $n \times n$ from M_T training samples, where M_T denotes the number of training samples, with $M_T < M$ and $n < m$. The extracted N patches $\{P_j\}_{j=1}^{N}$ are used for learning the convolutional kernels with size of $n \times n$ via K-means clustering. In this paper, we choose some groups of patches with different sizes of 22×22, 20×20, $\ldots$, 6×6, respectively, for the results after convolutional process can be divided with no reminder in pooling process, the pooling size is designed as 2×2, in addition, N is designed as 10,000.

The process of data extraction is shown as Figure 2.

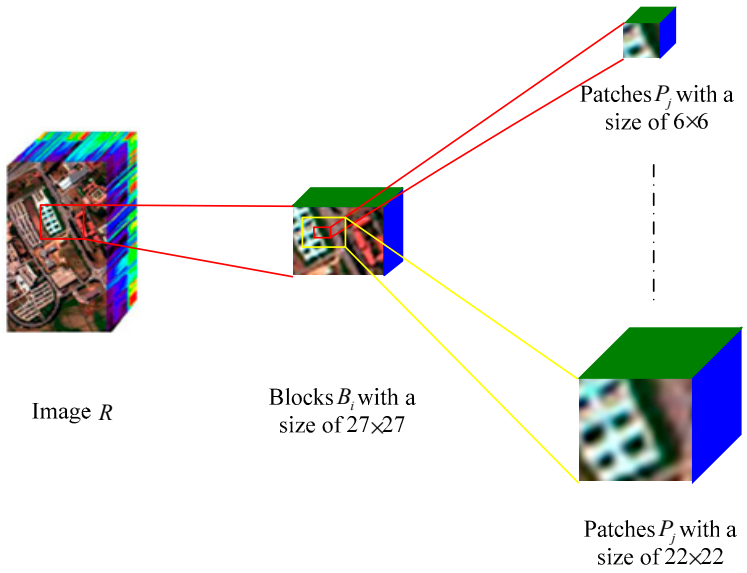


Figure 2. The block (sample) is extracted from image R and the different groups of samples with different sizes are extracted from block B_i, respectively.

2.2. Clustering the Convolutional Kernels via K-Means

The method of pre-learned convolutional kernels is based on K-means algorithm. To verify the adaptive size, we set the class number K as 50 by experience.

We firstly reshape each patch P_j into a column vector as a vector P_j with a size of $n^2 \times 1$. All the vectors denote as $P \in \{P_1, \ldots, P_j, \ldots, P_{10000}\}$. The steps of K-means are shown as follows:

Step 1: we randomly choose 50 vectors from P as the initial cluster centers, i.e., $\mu_1, \ldots, \mu_f, \ldots, \mu_{50}$, $f = 1, \ldots, 50$.

Step 2: for each vector P_j, if the vector has same label with cluster center μ_f, the vector should exhibit the nearest distance to the cluster center μ_f than that to the other cluster centers.

$$label_{P_j} = \underset{1 \le f \le 50}{\operatorname{argmin}} \|P_j - \mu_f\|_2 \tag{1}$$

where the distance is Euclidean distance and $label_{P_j}$ denotes the label of the vector P_j.

Step 3: for all c_f vectors P_j which have the same label of μ_f in class f. We calculate the new mean value μ'_f as the new cluster center;

$$\mu'_f = \frac{1}{\left|c_f\right|} \sum_{j \in c_f} P_j \tag{2}$$

c_f denotes the number of vectors in the class f.

Step 4: stop condition: repeat step 2 and step 3 Z times, here, $Z = 400$. After the computing process, the final mean values, i.e., $\mu_1, \ldots, \mu_f, \ldots, \mu_{50}$, represent the final cluster centers. For z groups of patch sizes, the cluster centers denote as $\mu_1^z, \ldots, \mu_f^z, \ldots, \mu_{50}^z$, here, $z = 1, 2, \ldots, 9$, since the chosen 9 groups of patch sizes. The cluster centers then are reshaped as the convolutional kernels and are ultimately adopted by the K-means Net.

Since the patches with different sizes extracted from the same sample, they often show different degrees of representation ability. Moreover, the patches with different sizes have different clustering results and different distributions in the 2-D plane. The detailed discussion will be introduced in Section 3.

2.3. Determination Method of Adaptive Kernel Size

Given the clustering results using different groups of patch (kernel) sizes, we determine the adaptive kernel size as the following steps:

Step 1: We compute the inner-class distance D_{inner}.

We compute the inner-class distance matrix $D_{inner(K_f)}$ of each class f. f denotes the class number. We adopt Euclidean distance for distance measure.

A variable value N_f is introduced to represent the number of patches in each class.

The matrix $D_{inner(K_f)}$ is given as Equation (3):

$$D_{inner(K_f)}(1, K_f) = \left\| P_{K_f} - u_f^z \right\|_2 \tag{3}$$

where f is 1, 2, ..., 50, P_{K_f} denotes the K_f-th vector in class f, $K_f = 1, 2, \ldots, N_f$, N_f denotes the number of patches in class f, u_f^z denotes the cluster center of the class f.

In each class, $D'_{inner(f)} = \sum_{K_f=1}^{N_f} D_{inner(K_f)}(1, K_f)$.

For the variable value N_f, we rank the number of samples in each class from small to large.

The weight w^f as the quotient between the number of patches in the class f and the number N of all patches, which is shown as Equation (4):

$$w^f = N_f / N \tag{4}$$

e^f denotes the weight of w^f, which represents the rank of the number of patches in each class. If a class has the max number of patches, the corresponding e^f is set to 50/50. Oppositely, if a class has the minimum number of points, whose e^f is set as 1/50 (50 denotes the number of classes). Through ranking the number of patches in the other classes from large to small, the other e^f corresponding to the patches number ranks of classes are set to 49/50, 48/50, ..., 3/50, 2/50, respectively.

The inner distance $D_{inner(f)}$ in class f is shown as the Equation (5):

$$D_{inner(f)} = w^f \cdot e^f \cdot D'_{inner(f)} / N_f \tag{5}$$

For all classes, the final demonstration of inner-class distance value D_{inner} is described as Equation (6):

$$D_{inner} = (\sum_{f=1}^{50} D_{inner(f)})/50 \tag{6}$$

Step 2: We compute the inter-class distance D_{inter}.

We compute the distance matrix which is composed of the distances among the cluster centers. In this paper, the distance matrix with a size of 50×50 is described as D_M. Each element in D_M is calculated as $D_M(r,t) = \|\mu_r^z - \mu_t^z\|_2$, where $r = 1, \ldots, 50$, $t = 1, \ldots, 50$, μ_r^z and μ_t^z are considered as the class centers of class r and class t, respectively.

We normalize the matrix D_M as Equation (7):

$$D_M = D_M/\max(D_M) \tag{7}$$

where $\max(D_M)$ denotes the max element in matrix D_M.

Finally, the inter-class distance D_{inter} is shown as:

$$D_{inter} = \frac{1}{50}\sum_{r=1}^{50}\sum_{t=1}^{50} D_M(r,t) \tag{8}$$

where r and t denote the line and column of the distance matrix D_M, respectively. D_{inter} is a scalar.

Step 3: The evaluation indicator of clustering results with different kernel sizes is shown as follows:

With the different kernel sizes $n = [22, 20, \ldots, 6]$, the evaluation indicator $EI(n)$ is:

$$EI(n) = \frac{D_{inter}}{D_{inner}} \tag{9}$$

where n denotes the kernel size and $EI(n)$ denotes the evaluation indicator value of the clustering result with a size of $n \times n$.

Then, through ranking $EI(n)$, the optimal kernel size is $n \times n$, which leads to the largest value $EI(n)$.

The flow chart of determining the adaptive size of the convolutioanal kernels is shown in Figure 3.

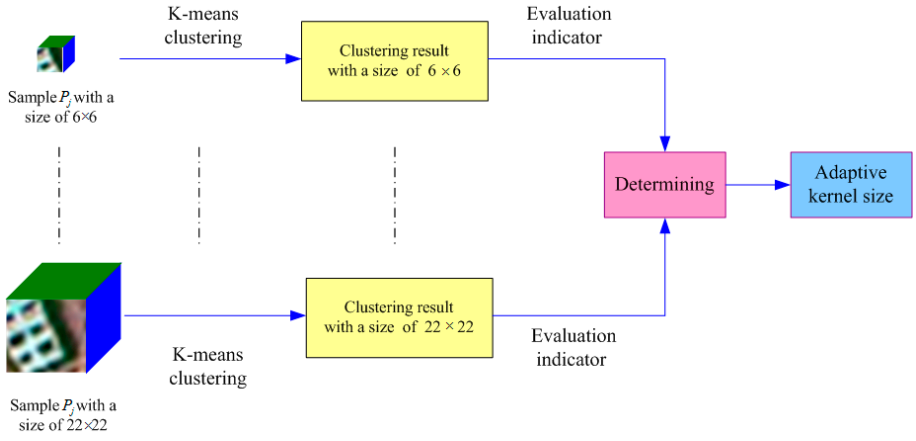


Figure 3. The flow chart of determining the adaptive size of the convolutional kernels.

2.4. Convolutional Neural Networks Classification

Through reshaping the clustering centers $\mu_1, \ldots, \mu_f, \ldots, \mu_{50}$ with the adaptive kernel size into patches $C^1, \ldots, C^f, \ldots, C^{50}$, these patches can be directly used for the convolutional kernels in the CNNs framework.

With the pre-learned kernels C^f, a convolutional neural networks as described in [27] is developed 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 shown as Figure 4.

There are 50 kernels in the convolutional layer. Each feature map is calculated by taking the dot product between the f-th kernel C^f with a size of $n \times n \times h$, $C \in R^{n \times n \times h \times f}$ and local context area x of size $m \times m \times h$ with h number of channels, $x \in R^{m \times m \times h}$. The feature map corresponding to the f-th filter $O \in R^{(m-n+1) \times (m-n+1)}$ is calculated as:

$$O_{ij}^f = \sigma(\sum_c \sum_{a=0}^{n-1} \sum_{b=0}^{n-1} C_{abc}^f x_{i+a,j+b}^h) \tag{10}$$

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

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

$$g_{ij}^f = \max(O_{1+p(i-1),1+p(j-1)}^f, \ldots, O_{pi,1+p(j-1)}^f, \ldots, O_{1+p(i-1),pj}^f, \ldots, O_{pi,pj}^f) \tag{11}$$

The f feature maps are reshaped into column vectors. Then, all the column vectors are connected into a fully connected layer, auto-encode unit is used to process the connected column vector and it represents the feature of column vector. The output results of hidden layer in the auto-encode unit were used to connect the classification layer.

The last soft-max layer is used for output final classification result.

The structure of the K-means based CNNs with adaptive kernel size is shown as Figure 4.

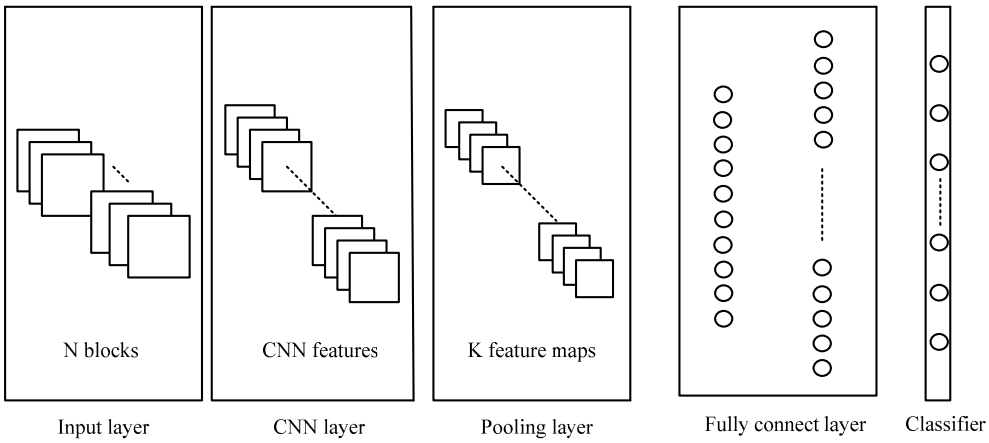


Figure 4. The structure of the K-means based CNNs.

3. Experiments and Analysis

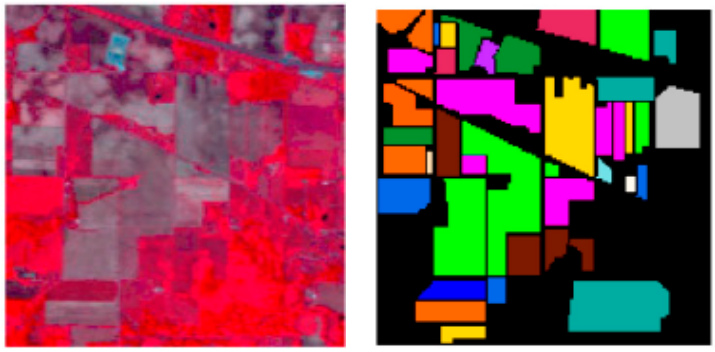
To demonstrate the effectiveness of the proposed method, two HSI datasets are adopted in the following experiments. These two datasets are used to validate the feasibility and effectiveness of the proposed CNNs based K-means with adaptive kernel size in classification. In the following sections, we firstly introduce the datasets. Then, the detailed experimental setting are provided. Finally, two experiments are conducted to show the HSI classification results of the proposed method.

3.1. Datasets

Two public image datasets are utilized in our experiments.

Dataset 1: In order to evaluate the proposed method on the complex dataset, the first dataset is the benchmark Indian Pines image, shown as Figure 5a. It is gathered by AVIRIS sensor over the Indian Pines test site in North-western Indiana. The ground reference is shown in Figure 5b. This

image contains 145 × 145 pixels and 224 spectral bands, the wavelength ranges from 0.4 to 2.5 um. The number of bands is reduced to 200. Eleven interesting classes of this image were classified. 5108 image context area samples with a size of 27 × 27 were extracted. We choose 11 categories of the total 16 categories for the experiment. Each category of image samples is given in Table 1. This dataset is used to analyze the distributions of the extracted patches in different groups with different sizes for certifying the extracted patches for learning kernels with different distributions. It is also used to test the feasibility and effectiveness of the proposed approach for classification.



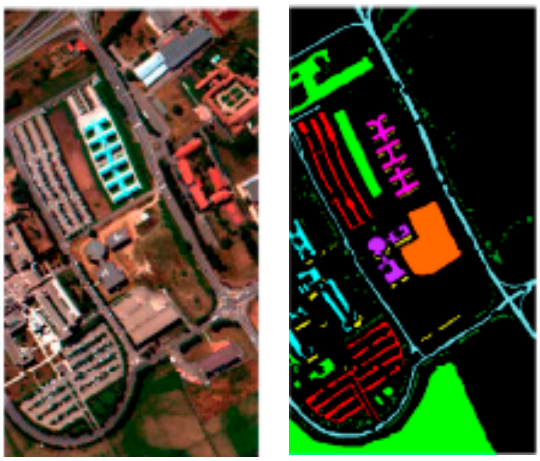
(**a**) (**b**)

Figure 5. The Indian Pines on Dataset 1. (**a**) shows the composite image; (**b**) shows the groundtruth of Indian Pines dataset, where white area denotes the unlabeled pixels.

Table 1. Ground truth of classes and their respective sample size on Indian Pines scene.

Class		Samples			
Number	**Classes**	**Total**	**Training**	**Validation**	**Testing**
1	Alfalfa	46	25	2	19
2	Corn-notill	1245	624	131	490
3	Grass-pasture	67	30	6	31
4	Grass-trees	701	332	73	296
5	Grass-pasture-mowed	28	15	3	10
6	Soybean-notill	807	396	70	341
7	Soybean-mintill	1986	1004	201	781
8	Soybean-clean	7	3	0	4
9	Woods	211	119	23	69
10	Buildings-Grass-Trees-Drives	7	5	0	2
11	Stone-Steel-Towers	3	1	1	1
	Total	5108	2554	510	2044

Dataset 2: The second dataset is the benchmark Pavia University image. It is acquired by ROSIS sensor during a flight campaign over Pavia, northern Italy. As shown in Figure 6, this image contains 610 × 610 pixels and 103 spectral bands. The number of bands is reduced to 100 by selecting the top 100 bands from 103 bands, and the whole image was used. The total numbers of samples are split into training, validation and testing samples with ratios 0.5, 0.1 and 0.4. Furthermore, 31,571 image context area samples with a size of 27 × 27 were extracted. Among them, 15,785, 3157 and 12,629 samples are used for training, validation and testing, respectively. The details of each category of samples were given in Table 2. This dataset is used to test the feasibility and effectiveness of the proposed approach for classification.



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

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

Table 2. Ground truth of classes and their respective sample size in the Pavia University scene.

Class		Samples			
Number	**Classes**	**Total**	**Training**	**Validation**	**Testing**
1	Asphalt	4728	2349	501	1878
2	Meadows	11,188	5593	1136	4459
3	Gravel	1048	509	142	397
4	Trees	2513	1273	238	1002
5	Painted metal sheets	1345	645	119	581
6	Bare Soil	5029	2530	466	2033
7	Bitumen	1330	651	132	547
8	Self-Blocking Bricks	3535	1820	329	1386
9	Shadows	855	415	94	346
	Total	31,571	15,785	3157	12,629

3.2. Experimental Parameter Settings

In the experiment, the samples (blocks) are randomly extracted from the HSI dataset, and then some groups of patches are extracted from the training samples for learning the pre-learned kernels. Each group of kernels should contain a constant kernel size. The pre-learned kernel number (patches number) should be fixed as 50. And the pre-learned kernels would be used in the pre-learned CNNs framework.

In the experiment, as shown in Figure 4, the CNNs framework used one convolutional layer, one pooling layer, one auto-encode layer and a classifier. In our algorithm, the pooling layer adopts the overlapping rule with size of 2×2, the number of neurons in the hidden layer of auto-encoding is set to 1000 and the max iteration for training the classifier was 400.The learning rate is 0.0001 and the momentum is 1. The batch size is set as 200. The testing accuracy is the average value of 10 trials.

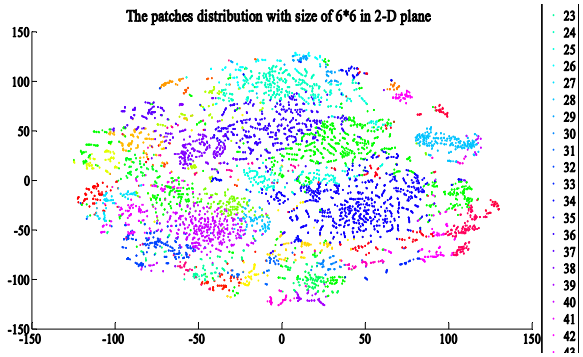
The code is running on the computer with Intel Xeon E5-2678 V3 2.50 GHz $\times$ 2 (Intel, Santa Clara, CA, USA), NVIDIA Tesla (NVIDIA, Santa Clara, CA, USA) K40c GPU $\times$ 2, 128 GB RAM, 120 GB SSD and Matlab 2016a (MathWorks, Natick, MA, USA). The gradient is computed via batch gradient descent,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

3.3. Experimental Results

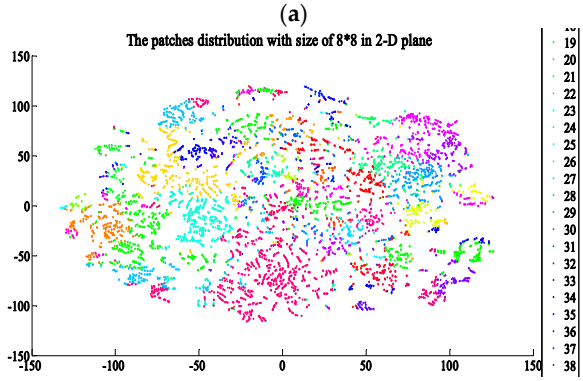
3.3.1. Different Performances in 2-D Plane of the Patches with Different Sizes

The aim of this experiment is to show the performance of the patches of non-uniform distribution with different sizes in 2-D plane. In the experiment, the patches with different sizes are extracted from the HSI in dataset 1, which are reshaped into vectors, and then the vectors are projected onto the 2-D plane through the tSNE_VISURE_2dDATA tools. The chosen patch sizes are 22×22, 20×20, ..., 6×6. The 2-D plane represents both distribution of each patch and the result of patches distribution after K-means clustering with 50 classes.

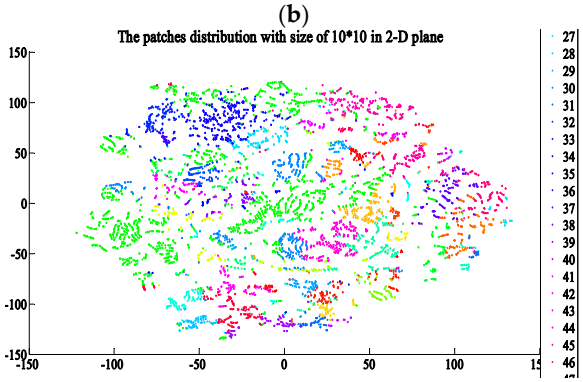
Figure 7 shows performances of the patches of projected into the 2-D plane with different sizes are different. This is because that the patches with different sizes show different qualities. The patches with different sizes projected onto 2-D plane after K-means clustering are also different, which make it difficult to evaluate the clustering results with the different patch size. For this reason, the evaluation indicator of clustering results should be defined anew.



(**a**)

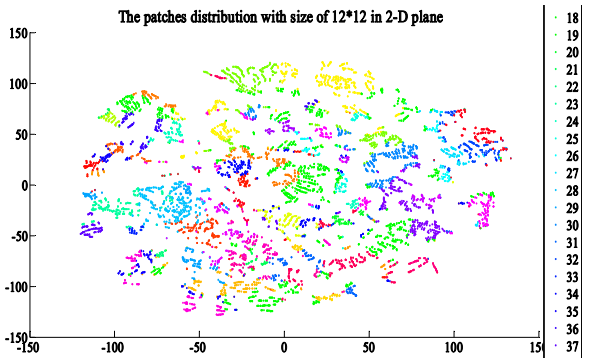


(**b**)

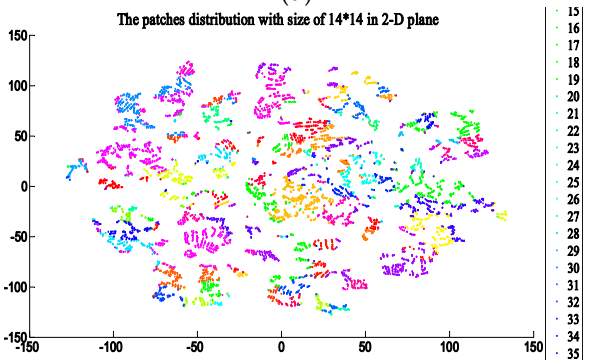


(**c**)

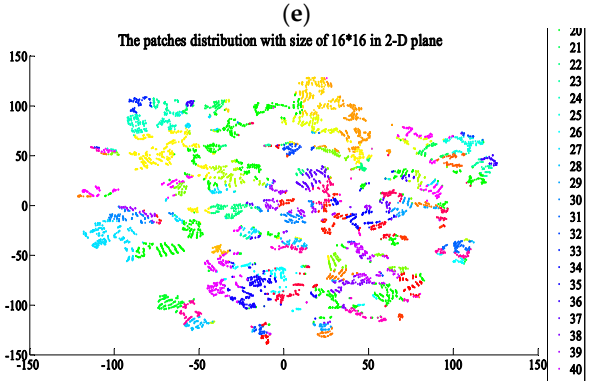
Figure 7. *Cont.*



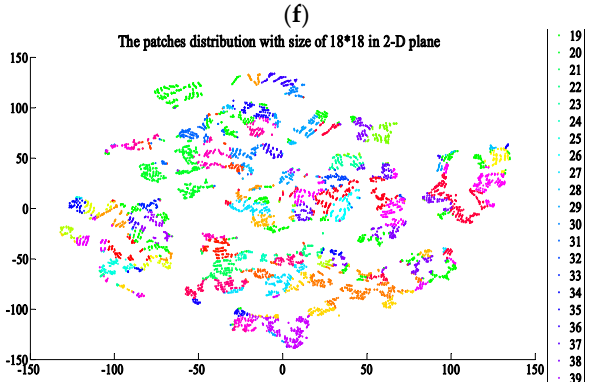
(**d**)



(**e**)



(**f**)



(**g**)

Figure 7. *Cont.*

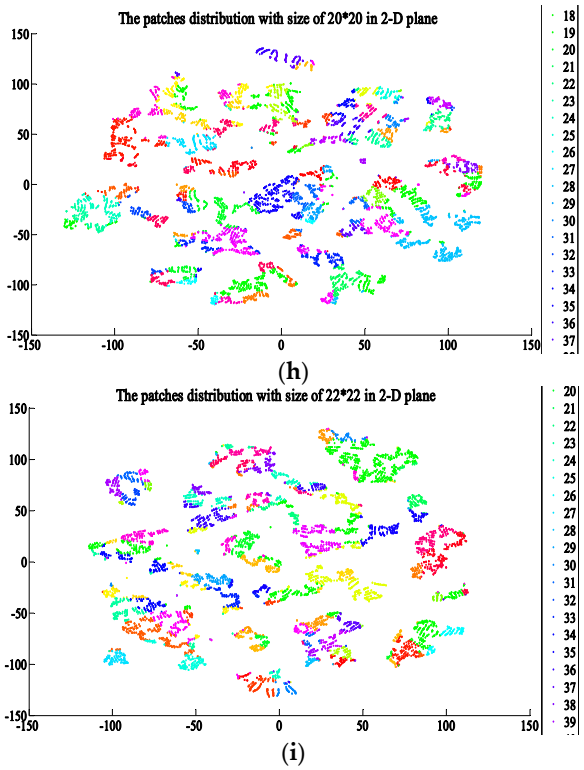


Figure 7. The different performances in 2-D plane of the patches with different sizes. The x and y axis represent the position coordinates. The performances of patches with sizes (**a**) size 6 × 6; (**b**) size 8 × 8; (**c**) size 10 × 10; (**d**) size 12 × 12; (**e**) size 14 × 14; (**f**) size 16 × 16; (**g**) size 18 × 18; (**h**) size 20 × 20; (**i**) size 22 × 22.

3.3.2. Effectiveness of the Adaptive Kernels Size Determined by CNNs Based K-Means Clustering

To demonstrate the effectiveness of the adaptive kernel size determined by an evaluation indicator of K-means, we compare the value of evaluation indicator and classification accuracy by the CNNs based on K-means algorithm on two HSI datasets. Dataset 1 and Dataset 2 are used in the experiment.

We report the evaluation indicator value and the testing accuracy with different patches sizes on each dataset in Tables 3 and 4.

Table 3. The evaluation indicator value and the testing accuracy of K-means Net with different kernel sizes on Dataset 1.

	Kernel Size	D_{inter}	D_{inner}	Evaluation Indicator	Test Accuracy (%)
Indian Pines	6	10.4284	0.6168	**16.9080**	**99.7945**
	8	10.6035	0.9067	11.6951	99.7358
	10	10.7961	1.1377	9.4892	99.6673
	12	11.3785	1.4560	7.8149	99.5344
	14	12.0906	1.6901	7.1536	99.4325
	16	12.4440	1.8427	6.7531	98.9973
	18	12.3634	2.3429	5.2769	98.0333
	20	12.6928	2.5880	4.9045	97.7104
	22	12.4696	2.6117	4.7745	95.3327

Table 4. The evaluation indicator value and the testing accuracy of K-means Net with different kernel sizes on Dataset 2.

	Kernel Size	D_{inter}	D_{inner}	Evaluation Indicator	Test Accuracy (%)
Pavia University	6	8.9495	0.3671	**24.3768**	**99.7624**
	8	8.8101	0.5358	16.4416	99.7545
	10	9.6805	0.7095	13.6431	99.7086
	12	9.9316	0.9145	10.8599	99.6263
	14	10.0838	1.0294	9.7956	99.3855
	16	10.3295	1.2930	7.9889	98.7093
	18	9.8187	1.4530	6.7575	97.6609
	20	10.4964	1.6737	6.2714	95.2189
	22	10.2319	1.8970	5.3938	90.0024

In Table 3, the proposed method determines the adaptive kernel size as 6×6 on Dataset 1. The chosen kernel sizes in this experiment are 6×6, 8×8, ..., 22×22. The kernel size with the largest value of evaluation indicator is 6×6, and the corresponding evaluation indicator value is 16.9080. The evaluation indicator value shows the samples with size of 6×6 owns the best clustering result. It can be seen that adaptive kernels size 6×6 via the proposed method shows the best testing classification accuracy, 99.7945%. Similar observations can be made from Table 4. Therefore, the proposed method is demonstrated to have the potential to determine the adaptive kernel size in the other datasets.

3.3.3. Performance Evaluation of K-Means Net

In this part, the proposed method is compared with three state-of-the art pre-learned kernels based CNNs methods, including PCA-Net [26], Random Net and MCSFDP Net [27]. For fair comparison, the same CNNs architecture is designed in all the compared methods. The number of kernels is set to 50, while the adaptive kernel with size of 6×6 is determined by the proposed method. These parameters in these four networks are determined through tuning parameters such as learning rate and moment value, the iteration is set to 400.

The average testing classification accuracy of our proposed algorithm, PCA-Net and Random Net and MCFSFDP Net on the Dataset 1 and Dataset 2, is given in Tables 5 and 6. The results obviously show that the Random Net, MCFSFDP Net and K-means Net with the adaptive kernel size obtain the acceptable accuracy. It reveals that our proposed method can produce the second best classification result in the four comparison methods. Moreover, the MCFSFDP Net acquired the best classification accuracy, which relies on the more advanced clustering method for pre-learning convolutional kernels.

Table 5. The testing accuracy of different CNNs methods which compared with K-means Net on Dataset 1.

Methods	PCA-Net-50	Random Net-50	MCFSFDP Net-50	K-Means Net-50
Accuracy (%)	89.4912	99.7652	**99.9413**	**99.7945**

Table 6. The testing accuracy of different CNNs methods which compared with K-means Net on Dataset 2.

Methods	PCA-Net-50	Random Net-50	MCFSFDP Net-50	K-Means Net-50
Accuracy (%)	91.1711	99.6844	**99.9251**	**99.7624**

Specifically, K-means Net has the fast speed and data-determined character for clustering the kernels. These are two advantages of the K-means Net. In the other methods, PCA-Net is data-determined for learning kernels. However, with the increase of sample dimension, the effect of PCA reduction will be dropped. In other words, the parameter of reduction dimension is hard to be

designed, which influences the kernel performance of PCA-Net. What's more, the process of learning kernels via PCA-Net is slower than the process of K-means Net. The kernels are learned via randomly initialization in Random Net. In this reason, the kernels are not data-determined. Therefore, Random Net is not applicable in a large extent. In MCFSFDP Net, the kernels are learned via the MCFSFDP method. These kernels are data-determined by MCFSFDP Net and this clustering method is based on density and distance, which has the advanced clustering performance than K-means. However, the MCFSFDP algorithm needs a step of calculating a distance matrix, this step needs more time and memory than the K-means Net, PCA-Net and Random Net.

4. Discussion

In the kernel-size determination scheme, the relationship of the clustering results with evaluation indicator and the testing accuracy on Dataset 1 and Dataset 2 are shown in Figures 8 and 9. Figures 8a and 9a show the evaluation indicator value of clustering results achieved with different patch sizes that are calculated by the value of inter distance and inner distance on two Datasets. Figures 8b and 9b show the classification results with variation of kernel sizes on Dataset 1 and Dataset 2.

In Figures 8a and 9a, the inner distance is increased when the kernel size is increased. Nevertheless, the inter distance cannot be increased with the regular of the inner distance, furthermore, it presents an unobvious variation trend in inter distance. However, the evaluation indicator value calculated by inner and inter distance can be reduced evidently on Dataset 1 and Dataset 2 when the kernel size is increased.

In Figures 8b and 9b, the classification results and the evaluation indicator value are reduced at the same time with the increased kernel size. The evaluation indicator value and classification results have the same reduced trend.

In this case, the proposed evaluation indicator value is more suitable than both inner distance and inter distance for determining the adaptive size of convolutional kernels.

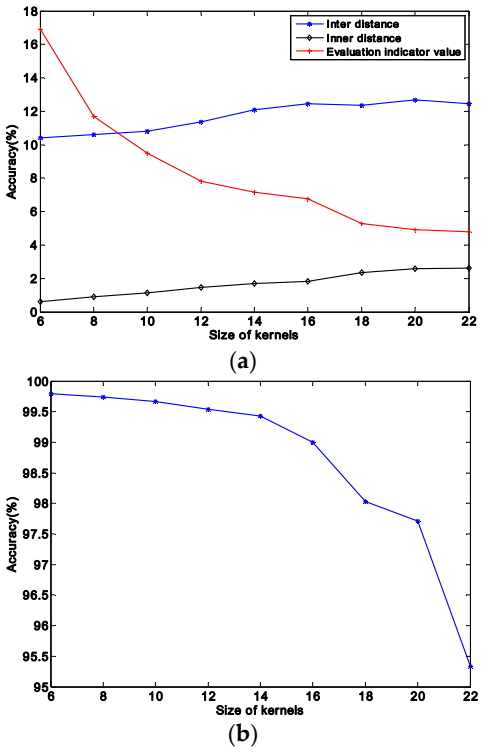


Figure 8. The classification accuracy influence with the kernel size. (**a**) the evaluation indicator value, the inter distance and inner distance with different kernel sizes on Dataset 1; (**b**) the classification accuracy with the evaluation indicator value on Dataset 1.

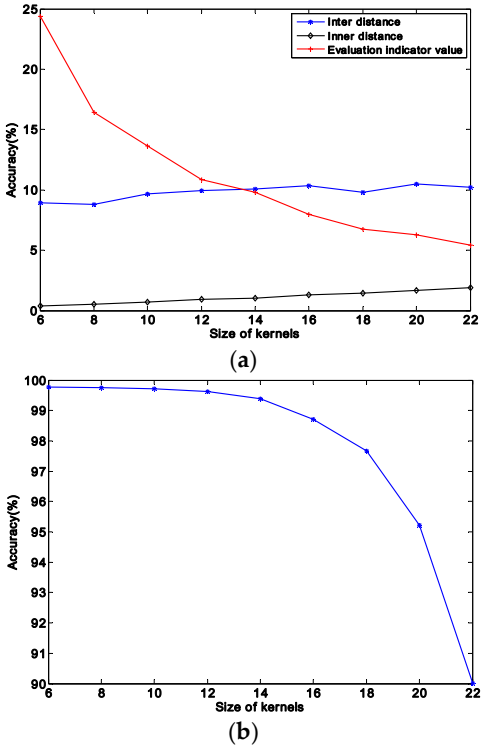


Figure 9. The classification accuracy influence with the kernel size. (**a**) the evaluation indicator value, the inter distance and inner distance with different kernel sizes on Dataset 2; (**b**) the classification accuracy with the evaluation indicator value on Dataset 2.

5. Conclusions

In this paper, we propose a novel CNNs classification framework based on K-means for HSI classification, which can determine the size of the kernels from training data through an adaptive manner. Specifically, this framework utilizes the K-means algorithm to cluster groups of patches with different patch sizes extracted from the training data, and then the convolutional kernel size can be determined adaptively by the proposed evaluation indicator of the clustering results. The clustering centers with the adaptive kernel size can be considered as the pre-learned kernels in the CNNs framework based on K-means. The experimental results demonstrate that the proposed method is able to seek a good kernel size for the datasets, which can help to define a more suitable CNNs architecture for good feature extraction and classification.

Acknowledgments: This work was supported by the Key Project of the National Natural Science Foundation of China (grant no. 61231016), the National Key Research and Development Program of China (grant no. 2016YFB0502502), the National Natural Science Foundations of China (grant no. 61471297, grant no. 61671385, grant no. 61301192, and grant no. 61771397), the China 863 Program (grant no. 2015AA016402) and Shaanxi International Cooperation Project (grant no. 2017KW-006).

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, 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. Landgrebe, D. Hyperspectral image data analysis. *IEEE Signal Process. Mag.* **2002**, *19*, 17–28. [CrossRef]
2. Richards, J.A.; Richards, J. *Remote Sensing Digital Image Analysis*; Springer: Berlin/Heidelberg, Germany, 1999; Volume 3.
3. Zortea, M.; De Martino, M.; Serpico, S. A SVM Ensemble Approach for Spectral-Contextual Classification of Optical High Spatial Resolution Imagery. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Barcelona, Spain, 23–28 July 2007; pp. 1489–1492.
4. 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]
5. 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]
6. 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]
7. 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]
8. 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]
9. 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]
10. 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]
11. Chen, Y.; Lin, Z.; Zhao, X.; Wang, G.; Gu, Y. Deep learning-based classification of hyperspectral data. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 2094–2107. [CrossRef]
12. 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]
13. Hinton, G.E.; Osindero, S.; Teh, Y.W. A fast learning algorithm for deep belief nets. *Neural Comput.* **2006**, *18*, 1527. [CrossRef] [PubMed]
14. Schölkopf, B.; Platt, J.; Hofmann, T. Greedy Layer-Wise Training of Deep Networks. In Proceedings of the International Conference on Neural Information Processing Systems, Hong Kong, China, 3–6 October 2006; pp. 153–160.
15. 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]
16. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In Proceedings of the International Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1097–1105.
17. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *Comput. Sci.* **2014**.
18. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going Deeper with Convolutions. In Proceedings of the IEEE Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
19. 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, Boston, MA, USA, 7–12 June 2015; pp. 770–778.
20. Zhang, L.; Wei, W.; Zhang, Y.; Shen, C.; Aton, V.; Qin, S. Cluster Sparsity Field: An Internal Hyperspectral Imagery Prior for Reconstruction. *Int. J. Comput. Vis.* **2018**. accepted.
21. Wang, C.; Zhang, L.; Wei, W.; Zhang, Y. When Low Rank Representation Based Hyperspectral Imagery Classification Meets Segmented Stacked Denoising Auto-Encoder Based Spatial-Spectral Feature. *Remote Sens.* **2018**, *10*, 284. [CrossRef]
22. Yu, S.; Jia, S.; Xu, C. Convolutional neural networks for hyperspectral image classification. *Neurocomputing* **2017**, *219*, 88–98. [CrossRef]
23. Chen, Y.; Jiang, H.; Li, C.; Jia, X.; Ghamisi, P. Deep Feature Extraction and Classification of Hyperspectral Images Based on Convolutional Neural Networks. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 6232–6251. [CrossRef]

24. 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]
25. Chen, D.; Ying, L.; Yong, X.; Wei, W.; Lei, Z.; Zhang, Y. Convolutional Neural Networks Based Hyperspectral Image Classification Method with Adaptive Kernels. *Remote Sens.* **2017**, *9*, 618. [CrossRef]
26. Coates, A.; Ng, A.Y. *Learning Feature Representations with K-Means*; Springer: Berlin/Heidelberg, Germany, 2012.
27. Längkvist, M.; Kiselev, A.; Alirezaie, M.; Loutfi, A. Classification and Segmentation of Satellite Orthoimagery Using Convolutional Neural Networks. *Remote Sens.* **2016**, *8*, 329. [CrossRef]
28. Li, S.; Xing, J.; Niu, Z.; Shan, S.; Yan, S. Shape Driven Kernel Adaptation in Convolutional Neural Network for Robust Facial Trait Recognition. In Proceedings of the IEEE Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 222–230.
29. Postavaru, S.; Stoean, R.; Stoean, C.; Caparros, G.J. Adaptation of Deep Convolutional Neural Networks for Cancer Grading from Histopathological Images. In Proceedings of the International Work-Conference on Artificial Neural Networks, Cadiz, Spain, 14 June 2017; pp. 38–49.
30. Fahad, A.; Alshatri, N.; Tari, Z.; Alamri, A.; Khalil, I.; Zomaya, A.Y.; Foufou, S.; Bouras, A. A survey of clustering algorithms for big data: Taxonomy and empirical analysis. *IEEE Trans. Emerg. Top. Comput.* **2014**, *2*, 267–279. [CrossRef]
31. Estivill-Castro, V. Why so many clustering algorithms: A position paper. *ACM SIGKDD Exp. Newslett.* **2002**, *4*, 65–75. [CrossRef]
32. Färber, I.; Günnemann, S.; Kriegel, H.-P.; Kröger, P.; Müller, E.; Schubert, E.; Seidl, T.; Zimek, A. On Using Class-Labels in Evaluation of Clusterings. In Proceedings of the MultiClust: 1st International Workshop on Discovering, Summarizing and Using Multiple Clusterings Held in Conjunction with KDD, Washington, DC, USA, 25–28 July 2010; p. 1.
33. Xu, D.; Tian, Y. A comprehensive survey of clustering algorithms. *Ann. Data Sci.* **2015**, *2*, 165–193. [CrossRef]