remote sensing

MDPI

Article

Divide-and-Conquer Dual-Architecture Convolutional Neural Network for Classification of Hyperspectral Images

Jie Feng *, Lin Wang, Haipeng Yu, Licheng Jiao and Xiangrong Zhang

Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, Xidian University, Xi'an 710071, China; tywanglin940816@163.com (L.W.); yhpwsid@163.com (H.Y.); lchjiao@mail.xidian.edu.cn (L.J.); xrzhang@mail.xidian.edu.cn (X.Z.)

* Correspondence: jiefeng0109@163.com; Tel.: +86-298-820-2279

Received: 25 January 2019; Accepted: 22 February 2019; Published: 27 February 2019

Abstract: Convolutional neural network (CNN) is well-known for its powerful capability on image classification. In hyperspectral images (HSIs), fixed-size spatial window is generally used as the input of CNN for pixel-wise classification. However, single fixed-size spatial architecture hinders the excellent performance of CNN due to the neglect of various land-cover distributions in HSIs. Moreover, insufficient samples in HSIs may cause the overfitting problem. To address these problems, a novel divide-and-conquer dual-architecture CNN (DDCNN) method is proposed for HSI classification. In DDCNN, a novel regional division strategy based on local and non-local decisions is devised to distinguish homogeneous and heterogeneous regions. Then, for homogeneous regions, a multi-scale CNN architecture with larger spatial window inputs is constructed to learn joint spectral-spatial features. For heterogeneous regions, a fine-grained CNN architecture with smaller spatial window inputs is constructed to learn hierarchical spectral features. Moreover, to alleviate the problem of insufficient training samples, unlabeled samples with high confidences are pre-labeled under adaptively spatial constraint. Experimental results on HSIs demonstrate that the proposed method provides encouraging classification performance, especially region uniformity and edge preservation with limited training samples.

Keywords: Hyperspectral image classification; divide-and-conquer; dual-architecture convolutional neural network; homogeneous and heterogeneous regions; superpixel segmentation

1. Introduction

With the rapid development of hyperspectral sensors, hyperspectral remote sensing images have become more available. Hyperspectral images (HSIs) often contain hundreds of narrow and contiguous spectral bands in the same scene, with wavelengths spanning the visible to infrared spectrum [1]. The detailed spectral information provided by hyperspectral sensors improves the capacity to differentiate the interesting land-cover classes. It makes HSI classification one of the most promising techniques in many practical applications, including agriculture [2], military [3], astronomy [4], mineralogy [5], surveillance [6], and environmental sciences [7,8].

HSI classification involves two key aspects: feature extraction and classification. Feature extraction is crucial in addressing the "Hughes phenomenon" [9] caused by high-dimensional spectral bands of HSIs. In the early stage of HSI feature extraction, various spectral-based methods were proposed, such as principal component analysis (PCA) [10,11], independent component analysis (ICA) [12,13], manifold learning [14], sparse graph learning [15], and local Fisher's discriminant analysis (LFDA) [16]. These methods are implemented by transforming original high-dimensional data into an appropriate

low-dimensional space. However, it is difficult to precisely distinguish different land-cover classes only by spectral information. To address this issue, some researchers make use of spatial information to extract features, such as Gabor filters [17], wavelets [18,19], extended morphological profiles [20], morphological attribute profiles [21], and extended multi-attribute profiles (EMAPs) [22]. Besides, multitask learning has powerful feature extraction ability due to its ability to incorporate shared information across multiple tasks. In one study [23], the kernel low-rank multitask method is proposed to capture multiple features from the 2-D variational mode decomposition domain for multi-/hyperspectral image classification.

A series of representative machine learning-based classification methods are used as classifiers, including *k*- nearest neighbors [24], logistic regression (LR) [25], extreme learning machine [26], sparse representation-based classification [27–29], support vector machine (SVM) [30,31], etc. Among these methods, SVM maximizes the margin among different classes in a kernel-induced feature space. It achieves outstanding performance for HSI classification, especially with small-sized training set.

The mentioned-above methods complete feature extraction and classification individually. Besides, these methods adopt manually-extracted features, which involve massive effort in feature engineering. In 2006, Geoffery Hinton proposed deep learning [32], and deep learning obtained a great success in computer vision [33–37]. Compared with traditional methods, deep learning-based methods extract hierarchical features and train the classifier simultaneously. Moreover, these deep learning-based methods adopt two or more hidden layers to extract more abstract and invariant features of data automatically.

A series of deep learning-based models have been introduced into the classification of HSIs. In one study [38], the stacked autoencoder (SAE) was proposed to extract deep features from hierarchical architecture. Subsequently, sparse SAE [39], denoising SAE [40], and Laplacian SAE [41] were successively proposed. In another study [42], Chen et al. presented a deep belief network (DBN) by learning the restricted Boltzmann machine network layer-by-layer. However, these methods cannot make full use of spatial information, since flattening training samples destroys the spatial structure in HSIs. Besides, there are so many parameters produced by full connection (FC) in these networks that a large number of available training samples are required.

Compared with SAE and DBN, convolutional neural network (CNN) [33] exploits local connections to effectively extract the spatial feature representation and shared weights to significantly decrease the number of parameters. Inspired by these properties, a series of CNN methods [43–54] have emerged for HSI classification. Hu et al. proposed a 1-dimensional (1D) CNN-based method to learn hierarchical spectral features of HSIs [50]. Makantasis et al. combined randomized PCA and CNN to encode spatial information of HSIs [51]. However, these two methods only exploit spectral information or spatial information, respectively. Later, some joint spectral-spatial CNN-based methods were proposed [48,51,52]. A dual-channel CNN (DCNN) was constructed to extract spectral and spatial features by 1D-CNN and 2D-CNN separately, then extracted spectral and spatial features were concatenated together [51]. Chen et al. presented another type of joint spatial-spectral feature extraction, where a 3-dimensional (3D) CNN (3DCNN) model was adopted to extract spectral and spatial information simultaneously [52]. However, the performance of these CNN methods depends on the quantity of training samples greatly. Generally, the collection of training samples is difficult in HSIs. Recently, Li et al. proposed a pixel-pair CNN (PPF-CNN) method by reorganizing and relabeling existing training samples [53]. Besides, in several studies [55–57], tensor-based models significantly reduced the number of weight parameters required to train the model via tensor decomposition. When the number of training data is limited, tensor-based classification models can perform well. Makantasis et al. proposed tensor-based linear and nonlinear models for HSI classification [55]. The data from all the sensors was fused into a tensor, and damage-sensitive features were extracted for classification in tensor-based models [56]. Recently, some other deep learning models are introduced for HIS [58,59]. A new fully CNN was proposed to extract the deep features of HSIs. Then, the optimized extreme learning machine is used for classification [58].

All the mentioned CNN-based methods [43–54] adopt a single fixed network structure for HSI classification. The single network structure ignores the complex land-cover distributions of HSIs. In heterogeneous regions, a large-sized spatial window input covers some samples coming from different classes. These neighbor samples with different classes may lead to misclassification of samples located around the boundaries. In this case, spectral information is mainly required for heterogeneous regions. On the contrary, in homogeneous regions, neighbor samples have similar spectral signatures. A small spatial window input may lack enough contextual information for classification. In this case, spatial and spectral information are required to analyze homogeneous regions simultaneously. Therefore, single fixed network structure may hinder the excellent performance of CNNs for HSI pixel-wise classification.

To address this problem, a novel divide-and-conquer dual-architecture CNN (DDCNN) method is designed for HSI classification. In DDCNN, a new regional division strategy based on local and non-local decisions is devised to divide HSIs into homogeneous and heterogeneous regions, respectively. The non-local decision is performed to search the superpixel-pair similarity in the whole image, while the local decision is made by spatially adjacent samples in the superpixels. For the homogeneous regions, larger-sized spatial windows are selected to extract adequately contextual information. A multi-scale CNN architecture with larger spatial windows is constructed to learn joint spectral-spatial features. For the heterogeneous regions, smaller spatial windows are selected to guarantee the samples belonging to the same class. A fine-gained CNN architecture with smaller spatial windows is constructed to learn hierarchical spectral features. Then, to alleviate the problem of insufficient training samples, unlabeled samples are selected by measuring the spectral similarity under adaptively spatial constraint. The samples with high confidences on the spectral similarity are pre-labeled to expand the training set.

The main contributions of this paper can be summarized as follows. (1) A novel dual-architecture CNN is designed instead of traditional single architecture considering various land-cover distributions of HSIs. In DDCNN, a multi-scale CNN architecture is constructed to improve the uniformity of homogeneous regions, and a fine-grained CNN architecture is constructed to avoid edge over-smoothness. (2) Regional division method-based local and non-local decisions are designed to divide the homogeneous and heterogeneous regions effectively, where superpixel-to-superpixel similarity is utilized in the non-local searching. (3) DDCNN devises a new sample augmentation method based on spectral similarity under adaptively spatial constraints, which alleviates the over-fitting problem of CNNs caused by the imbalance between insufficient training samples and numerous parameters.

The rest of this paper is organized as follows. Section 2 reviews the CNN briefly. Section 3 describes the procedure of the proposed DDCNN method in detail. Then, the experimental validation and corresponding analysis on several hyperspectral datasets are discussed in Section 4. Finally, some concluding remarks and suggestions are provided for further work in Section 5.

2. The Review of Convolutional Neural Networks

CNN, one of the deep leaning models, gains outstanding performance in computer vision tasks, such as classification, detection, and recognition. The architecture of CNN is based on the inspirations from neuroscience [60]. In the biological visual system, the cells in the cortex are sensitive to small regions, known as receptive fields. The strong capability of cells within receptive fields is used to exploit the local spatial correlation in images.

In contrast to other deep learning models, CNN possesses three core ideas: local connections, shared weights, and pooling. Local connections can extract local spatial features effectively corresponding to the receptive fields. Shared weight—that is, the connections between neurons—are replicated across the entire layer, which can significantly reduce the parameters of deep networks. Pooling is also known as downsampling, which extracts more robust features in the translation and deformation.

A traditional CNN is constructed by stacking several convolutional layers, pooling layers, and full connection layers to form deep architecture, where the output of each layer is provided as the input of the next layer. In the convolutional layer, the value of a neuron v_{ij}^{xy} at position (x, y) of the jth feature map in the ith layer is denoted as follows:

$$v_{ij}^{xy} = g\left(b_{ij} + \sum_{m}\sum_{p=0}^{P_i-1}\sum_{q=0}^{Q_i-1} w_{ijm}^{pq} v_{(i-1)m}^{(x+p)(y+q)}\right) \quad (1)$$

$$g(x) = ReLu(x) = \max(x, 0) \quad (2)$$

where m indexes the feature map in the $i-1$th layer connected to the current feature map, w_{ijm}^{pq} is the weights of position (p, q) connected to the mth feature map, P_i and Q_i are the height and width of the spatial window, and b_{ij} is the bias of the jth feature map in the jth layer.

3. Divide-and-Conquer Dual-Architecture CNN(DDCNN)

The flowchart of the proposed DDCNN method is shown in Figure 1. As shown in Figure 1, DDCNN consists of three stages: regional division with local and non-local decisions, dual-architecture CNN-based classification, and data augmentation based on spectral similarity under adaptively spatial constraint. A HSI dataset contains M training samples $X_{train} = \{x_1, \cdots, x_m, \cdots, x_M\}$ in an $\mathbb{R}^{d\times 1}$ feature space, where d is the number of spectral bands, and $1 \le m \le M$. The class label of training samples is represented by $Y = \{y_1, \cdots, y_m, \cdots, y_M\}; y_m \in \{1, \cdots, k, \cdots, K\}$, where K is the number of classes, and $1 \le k \le K$. At the regional division stage, the HSIs are divided into homogeneous and heterogeneous regions by using local and non-local decisions. Then, for the homogeneous regions, a multi-scale CNN architecture with larger-sized spatial window inputs is constructed to learn joint spectral-spatial features. For the heterogeneous regions, a fine-grained CNN architecture with smaller-sized inputs is constructed to learn hierarchical spectral features. Moreover, unlabeled samples with high confidences are selected to expand the training set by measuring the spectral similarity under the adaptive spatial constraint.

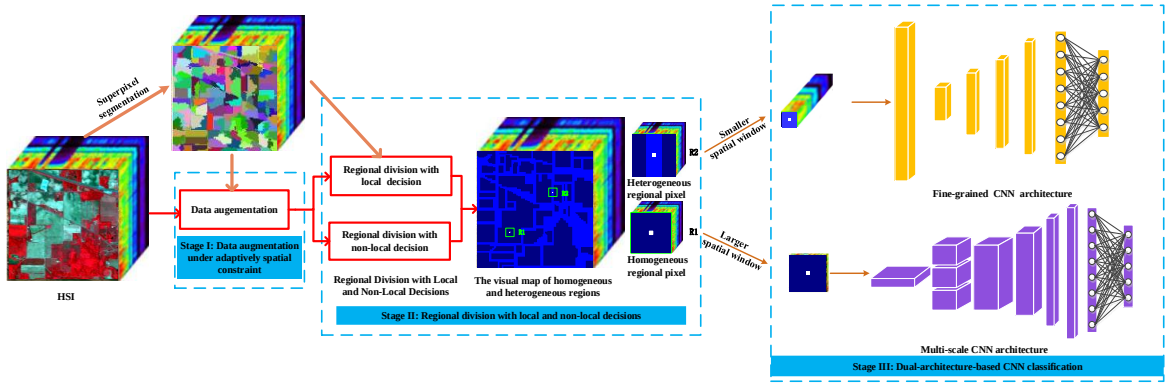


Figure 1. The flowchart of proposed divide-and-conquer dual-architecture convolutional neural network (DDCNN).

3.1. Superpixel Segmentation Based on Entropy Rate

In the superpixel segmentation, the images are divided into many superpixels. Each of them consists of spatially adjacent pixels with similar texture, color, brightness, or other characteristics [61]. Compared with pixel-based methods, superpixel-based methods utilize the spatial structure of the images and show good regional uniformity.

In this paper, the entropy rate method [62] is adopted to generate a 2-D superpixel map in HSIs. Compared with other superpixel segmentation methods, the entropy rate method is a graph-based clustering algorithm. It favors compact and homogenous nonoverlapping clusters, and has a fast

computation speed approximated as $O(|V|\log|V|)$, where V is the number of superpixels. More details of the entropy rate algorithm can be found in [62]. As shown in Figure 2, the first principal component of HSIs extracted by PCA is utilized as the base image for the superpixel segmentation. Then the base image is divided into V superpixels with adaptive sizes and shapes, denoted as $\{\pi_1, \pi_2, \cdots, \pi_V\}$. Each π_v $(1 \leq v \leq V)$ represents the vth superpixels. The segmentation result will be utilized in the regional division and data augmentation methods.

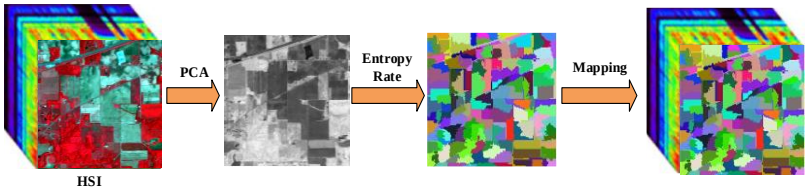


Figure 2. Procedure of the superpixel segmentation.

3.2. Regional Division with Local and Non-local Decisions

Most of CNN-based HSI classifications [43–48,50–52] are designed to exploit the spatial correlation in the neighborhood around the central pixel. That is, hyperspectral neighboring pixels in a spatial window are jointly represented by the CNN model for feature extraction. These CNN models commonly adopt a fixed-size spatial window as the input for feature extraction (e.g., 5×5, 27×27, etc.). This type of input hinders the excellent performance of CNNs for HSI classification. A large-sized spatial window input may include between-class samples in the heterogeneous regions, and a small-sized input may lead to extracting insufficient contextual information in the homogeneous regions.

Figure 3 illustrates an example for these two situations. In Figure 3, *i* and *j* are two samples in the HSIs. These two samples locate in the homogeneous and heterogeneous regions, respectively. Both them belong to the "GREEN" class. For the sample *i*, a larger spatial window (i.e., black box) contains some samples belonging to "BLUE", "PURPLE", and "YELLOW" classes instead of "GREEN" class. In this case, the sample *i* may be easily misclassified as the "BLUE", "PURPLE", or "YELLOW" class. If a smaller spatial window (i.e., red box) is selected, all the samples in the window belong to the "GREEN" class. For the sample *j*, all the samples in both larger and smaller spatial windows (i.e., black and red boxes) belong to the "GREEN" class. In the case, a larger spatial widow contains more adequately contextual information for feature extraction.

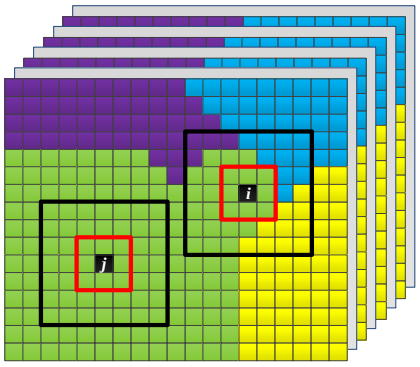


Figure 3. Illustration of samples in the homogeneous and heterogeneous regions.

To deal with these two situations, novel regional division method-based local and non-local decisions are designed to divide the HSIs into homogeneous and heterogeneous regions, where different CNN architectures are designed for homogeneous and heterogeneous regions, respectively. The divide and conquer strategy with homogeneous and heterogeneous regions is inspired by a visual attention-based model. Doulamis et al. proposed a fuzzy representation

of video content [63]. The divide and conquer concept was first proposed in the multiresolution recursive shortest spanning tree algorithm for video summarization and content-based retrieval [63]. Then, a neural network based scheme was used to select adaptive regions of interest (ROI) [64]. Then, a ROI-based motion-compensated discrete consine transform coder was proposed to extract foreground objects from background in videophones. Derived from the pioneering work on ROI [64], a neurobiological model of visual attention was proposed for video compression [65]. Later, visual attention based model was introduced into hyperspectral image processing [66,67].

(1) Regional Division with Local Decision: In the local decision, entropy rate-based superpixel segmentation is used to generate some homogeneous superpixels. Similar to the masking of edge detection, we choose a square frame (e.g., 3×3, 5×5) as the filter. If all the samples in the filter are within the same superpixel, the central sample is judged to be in the homogeneous regions. If these samples are divided into multiple superpixels, the central sample is located in the heterogeneous regions of the superpixel segmentation map. Actually, since the superpixel segmentation over-segments the HSIs, the central sample may be uncertain in the ground truth. It may belong to either the homogeneous or heterogeneous region.

Figure 4 illustrates the local regional division based on superpixel segmentation. Take the Indian Pines HSI as an example. Figure 4a shows the ground truth of the Indian Pines HSI. Figure 4b shows the results of entropy rate-based superpixel segmentation on the Indian Pines HSI. The samples *i*, *j*, and *k* represent the central samples located in the different regions. Figure 4c–e corresponds to the filters of the samples *i*, *j*, and *k*. In Figure 4d, since all neighbor samples in the filter belong to the same superpixel, the central sample *i* is judged to be in the homogeneous regions. In Figure 4c,e, the neighbor samples of the central samples *j* and *k* in the filters come from different superpixels. In the superpixel-based local decision, both sample *j* and *k* are judged to be in the heterogeneous regions. Actually, the sample *k* is located at the boundary area of superpixel segmentation map in Figure 4b rather than that of ground truth in Figure 4a. This is the "false boundary" phenomenon caused by the superpixel segmentation map. In the superpixel segmentation map, the samples belonging to the same class may be divided into several superpixels.

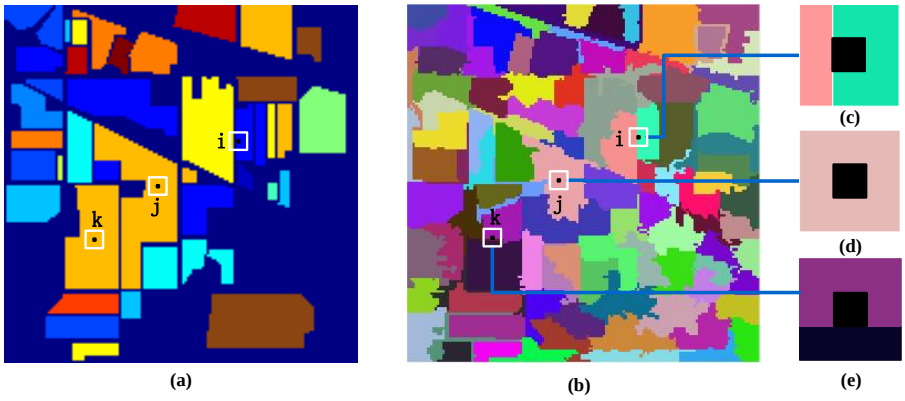


Figure 4. Illustration of local regional division based on superpixel segmentation: (**a**) ground truth; (**b**) superpixel segmentation map; (**c**) the filter of samples in the homogeneous region; (**d**) the filter of samples in the heterogeneous region; (**e**) the filter of samples in the "false boundary".

Let x_i be a central sample and N_i be the filter of x_i. If all the neighbor samples belong to the same superpixel π_v, the central sample x_i is judged to be in the homogeneous regions, and vice versa. The regional division based on local decision is formulated as follows:

$$x_i \in \begin{cases} X_{Ho} \text{ if } N_i \subseteq \pi_v(x_i) \\ X'_{He} \text{ else} \end{cases} \tag{3}$$

where $\pi_v(x_i)$ denotes the superpixel that the sample x_i belongs to. X_{Ho} represents the sample set in the homogeneous regions, and X'_{He} represents the sample set in the heterogeneous regions of superpixel segmentation map.

(2) Regional Division with Non-Local Decision: To alleviate the misdivision caused by the false boundary, a novel regional division based on non-local decisions is devised. In the HSIs, local information is used on the assumption that the samples in a local region belong to the same class. However, non-local information is also vital for HSI classification [68,69], since the samples belonging to the same class may be located in different regions.

In the non-local decision, pixel-similarity is extended to superpixel-similarity, which considers the structural information of current samples. For the samples judged in the heterogeneous regions by local decisions, the similarities of the neighbor samples and the current sample are calculated, where the current sample is represented by the samples with the same class in the global searching. Then, the similarities are compared with a calculated adaptive threshold. If the similarities of all the neighbors are larger than the threshold, the current sample is judged to be in the homogeneous region, and vice versa.

Let x_i represent a sample judged in heterogeneous regions by local decision, denoted as $x_i \in X'_{He}$. The filter N_i corresponding to x_i is divided into L_i small patches $N_i = \sum_{l=1}^{L_i} N_{i_l}$. The similarities of the neighbor samples and the current sample are calculated by superpixel-to-superpixel similarity $SS(\pi_v(x_i), \pi_v{}'(N_{i_l}))$. $\pi'_v(N_{i_l})$, which represents the superpixel π'_v containing the sample set N_{i_l}. If all the similarities are larger than the threshold T_k of the kth category, the sample x_i is judged to be in the homogeneous regions, and vice versa. T_k is a set as the minimum superpixel-based similarity of the samples in the kth category. If x_i is the unlabeled sample, k is set as the label of the training samples with most similarity. The regional division with non-local decisions is defined as follows:

$$\begin{aligned} x_i &\in \begin{cases} X_{Ho} \text{ if } SS\left(\pi_v(x_i), \pi'_v(N_{i_l})\right) \geq T_k, 1 \leq i_l \leq L_i \\ X_{He} \text{ else} \end{cases} \\ T_k &= \min\left\{ SS\left(\pi_v(x_i), \pi'_v(x_j)\right) \middle| x_i, x_j \in \psi_k \right\} \end{aligned} \tag{4}$$

where x_j is the sample in the kth category, and $\pi'_v(x_j)$ represents the superpixel correspond the sample x_j; ψ_k is the set of training samples in the kth category.

To measure the similarity of two superpixels, the average pooling strategy is applied to exploit the most significant information of superpixels. The similarity of two superpixels is calculated as:

$$SS(\pi_v(x_p), \pi'_v(x_q)) = S\left(\frac{1}{|\pi_v|} \sum_{x_p \in \pi_v} x_p, \frac{1}{|\pi'_v|} \sum_{x_q \in \pi'_v} x_q \right) \tag{5}$$

where $\pi_v(x_p)$ and $\pi'_v(x_q)$ represent two different superpixels corresponding to the samples x_p an x_q, respectively. The similarity measure is calculated by the heat kernel $S(x, x') = \exp\left(-\frac{\|x - x'\|^2}{\delta^2} \right)$.

Combining the local and global decisions (3) and (4), the sample is divided into homogeneous and heterogeneous regions according to (6):

$$\begin{aligned} x_i &\in \begin{cases} X_{Ho} \text{ if } N_i \subseteq \pi_v(x_i) \text{ or } SS\left(\pi_v(x_i), \pi'_v(N_{i_l})\right) \geq T_k \\ X_{He} \text{ else} \end{cases} \\ T_k &= \min\left\{ SS\left(\pi_v(x_i), \pi'_v(x_j)\right) \middle| x_i, x_j \in \psi_k \right\} \end{aligned} \tag{6}$$

where X_{He} is the set of samples in the heterogeneous regions.

3.3. Multi-Scale CNN Architecture

In the HSIs, the spectral signatures of samples in the same class may be different due to varied imaging conditions, e.g., changes in illumination, various environments, different atmospheric conditions, and temporal conditions. Therefore, spatial contexture information is critical for HSI classification. For the samples in the homogenous regions, a multi-scale CNN architecture with larger-sized spatial window inputs is constructed to extract joint spatial and spectral features. The multi-scale convolution consists of 1×1, 3×3, and 5×5 convolutional filters, where a 1×1 convolutional filter is used to extract spectral features, while 3×3 and 5×5 filters are utilized to extract various spatial contextual features.

In the multi-scale CNN architecture, a multi-scale convolutional filter is inspired by the Inception module [35]. The Inception module is used to exploit diverse local spatial structures of the input image, which enables the network to get deeper and wider and achieves state-of-the-art performance in image classification. The effectiveness of the inception module has been demonstrated in the large scale visual recognition challenge (LSVRC) 2014 [35]. The multi-scale convolutional filter is used to extract joint spectral-spatial features for HSI classification in this paper.

The architecture of multi-scale CNN network is shown in Figure 5. The input of multi-scale CNN architecture is larger-sized spatial windows with several principle components of PCA. A multi-scale filter is used in the first convolutional layer to jointly extract spatial structure and spectral correlation. Three feature maps are employed to perform cascade connection to form a joint spectral-spatial feature map. Subsequently, three convolutional layers are stacked one by one to extract hierarchical abstract features of HSIs. Then the extracted feature maps are flattened to a one-dimensional vector used as the input to two full connection layers. Finally, the extracted features are fed into the last soft-max classification layer.

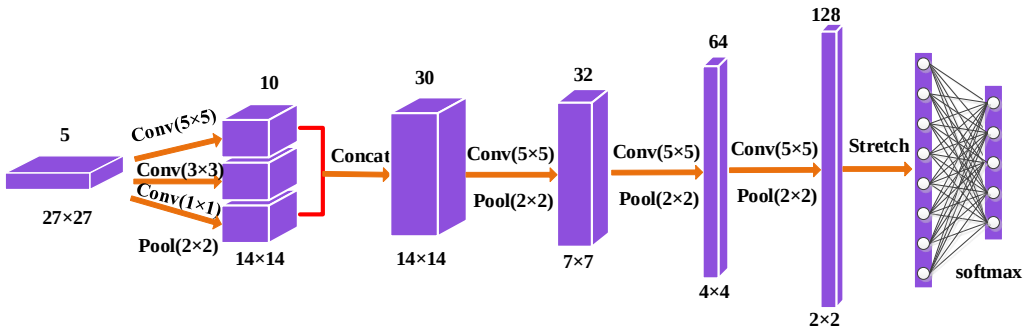


Figure 5. The construction of multi-scale convolutional neural network (CNN).

In this model, some regularization methods, data augmentation, dropout, early stop, and batch normalization (BN) are introduced to alleviate the over-fitting problem of CNNs. A new sample augmentation method is devised by pre-labeling the unlabeled samples based on spectral similarity under adaptive spatial constraint. Dropout is used in the second and third convolutional layers by preventing complex co-adaptations. It is used as the regularization technique to relieve the over-fitting problem. Early stop relieves the over-fitting problem by limiting the number of iterations. In addition, batch normalization is used in all the convolutional layers to accelerate the training of networks and reduce the internal covariate shift [70].

3.4. Fine-Grained CNN Architecture

For the samples in the heterogeneous regions, the spatial information is hard to use due to the distribution of different land-cover classes. The distinction for these samples mainly depends on hundreds of contiguous and narrow spectral bands. For these samples, a fine-grained CNN architecture with smaller-sized spatial window inputs is constructed to extract spectral information, where 1×1 convolution is used in all the convolutional layers.

The architecture of the fine-grained CNN network is shown in Figure 6. In the fine-grained CNN network, all the spectral bands are retained. The input of fine-grained CNN architecture is smaller-sized spatial windows with all the spectral bands. The 1×1 convolution is used in all the four convolutional layers. The 1×1 convolutional filter is proposed in Network In Network (NIN) [71], which allows complex and learnable interactions of cross channel information. Furthermore, it is also used to adjust the dimensionality of the feature maps. Here, 1×1 convolution is used to learn spectral correlations in the proposed network. Two full connection layers are stacked one by one after the convolutional layers. Finally, the extracted spectral features are fed into the soft-max classification layer. Similar to multi-scale CNN architecture, BN and dropout are used in the same position.

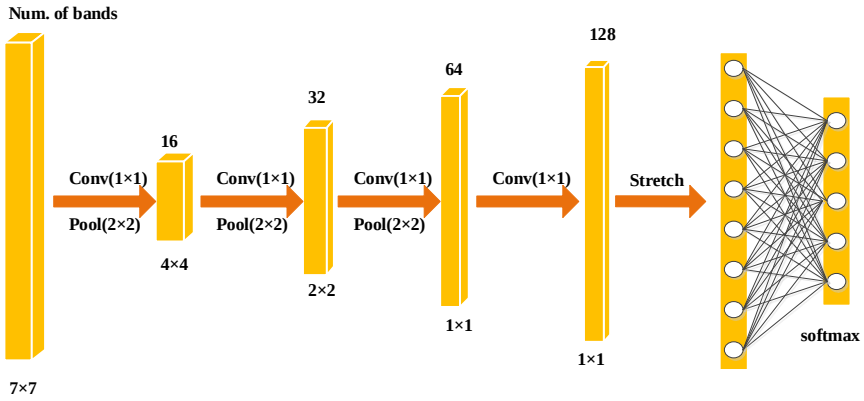


Figure 6. The construction of fine-grained CNN.

3.5. Data Augmentation Based on Spectral Similarity under the Adaptively Spatial Constraint

Deep learning models depend on a large quantity of training data due to the models being heavily parameterized. However, only limited training samples are available in HSI data. The CNN model tends to be over-fitting for HSI classification. To conquer this issue, a novel data augmentation method based on spectral similarity under adaptive spatial constraint is devised.

In the data augmentation method, superpixels with adaptive sizes and shapes are used for the spatial constraint. In the spatial constraint, unlabeled samples located in the same superpixel with training samples are considered as candidates. Then, unlabeled candidate samples with high confidence, which have the most spectral similarity with training samples, are selected. Finally, these selected unlabeled samples are pre-labeled as the same class with training samples, which are used to expand the training set.

Specifically, x_u denotes a current unlabeled sample, and $\pi_v(x_u)$ represents the superpixel where the sample x_u is located. For all the training samples $\{x_m | x_m \in \pi_v\}$ in the superpixel π_v, the similarities of current unlabeled sample x_u and all the training samples $\{x_m | x_m \in \pi_i\}$ are calculated. Then, the similarities are compared with a calculated threshold T_{π_v}, which is calculated by any two training samples in the superpixel π_v. If all the similarities are larger than the threshold, the current unlabeled sample is selected, and vice versa. The selected unlabeled samples are pre-labeled as the same label as the training samples $\{x_m | x_m \in \pi_v\}$, which is formulated as (7). These pre-labeled samples are used to expand the training set.

$$\begin{aligned} y_u &= \begin{cases} \underset{k}{\operatorname{argmax}} \sum\limits_{x_m \in \pi_v} I(y_m = k) & \text{if } \min\{S(x_u, x_m) | x_u, x_m \in \pi_v\} \geq T_{\pi_v} \\ 0 & \text{else} \end{cases} \\ T_{\pi_v} &= \min\{S(x_m, x_n) | x_m, x_n \in \pi_v, 1 \leq m, n \leq M, m \neq n\} \end{aligned} \tag{7}$$

where $I(\cdot)$ is the indictor function, and $y_u = 0$ represents that the unlabeled sample y_u, it is not selected to expand the training set.

3.6. The Procedure of DDCNN

The proposed DDCNN method uses the divide-and-conquer strategy to break the HSI classification into pixel-wise classification based on homogeneous and heterogeneous regions. Then, we solve the classification problems by two well-designed CNN networks separately and combine these solutions with the original classification problem. The proposed DDCNN method guarantees regional uniformity for homogeneous regions and edge preservation for heterogeneous regions of HSIs simultaneously. The procedure of DDCNN can be summarized in Table 1.

Table 1. The procedure of the proposed DDCNN method.

1.	INPUT: The training samples X_{train} and test samples X_{test} from K classes, the class labels of training samples, mini-batch size n, the number of training epochs E
2.	Begin
3.	Segment the whole HSI into V superpixels $\{\pi_1, \pi_2, \cdots, \pi_V\}$
4.	The training samples X_{train} are expanded to new training samples X'_{train} by (7)
5.	Training samples X'_{train} are divided into $X_{trainHo}$ and $X_{trainHe}$, and test samples X_{test} are divided into X_{testHo} and X_{testHe} by (6)
6.	initialize all the weight matrices and biases
7.	Input the training samples $X_{trainHo}$
8.	for every epoch
9.	for n training sample of every mini-batch
10.	compute the objective function l_{Ho} by the cross-entropy loss function
11.	update the parameters of the multi-scale CNN by minimizing loss function
12.	end for
13.	end for
14.	Input the training samples $X_{trainHe}$
15.	for every epoch
16.	for n training sample of every mini-batch
17.	compute the objective function l_{He} by the cross-entropy loss function
18.	update the parameters of the multi-scale CNN by minimizing loss function
19.	end for
20.	end for
21.	Count the labels Y_{test} by Y_{testHo} and Y_{testHe}
22.	END
23.	OUTPUT: the labels of the test samples classified by the trained DDCNN

4. Experimental Results

In this section, we validate the proposed DDCNN method on three benchmark HSI datasets. We investigate the performance of the proposed method from the following aspects: classification performance, running time, sensitivity analysis to the number of training samples, and sensitivity analysis of free parameters.

4.1. Data Description

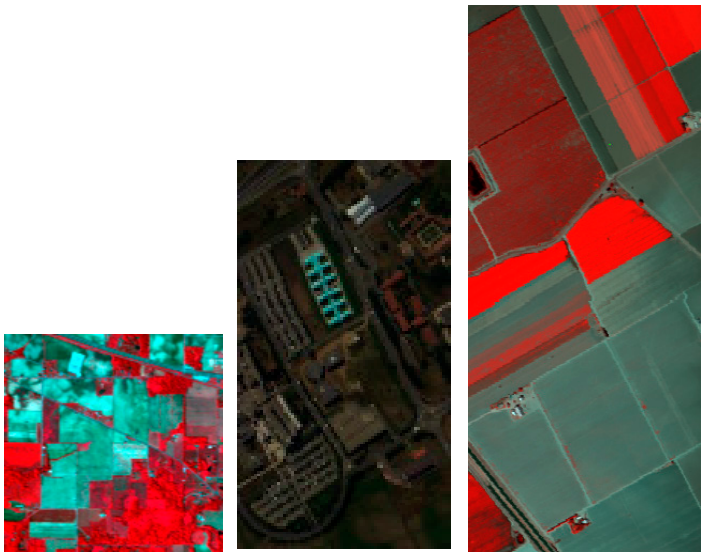
In this study, we adopt three HSI datasets for the experiment: the Indian Pines, Pavia University, and Salinas.

(1) The Indian Pines dataset is a mixed vegetation site over the Indian Pines test area in Northwestern India. It was acquired by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor, with the size of 145 × 145 pixels. There are 220 spectral bands in the wavelenth range of 0.4–2.5 μm in the visible and infrared spectrum. However, 200 spectral bands are preserved after 20 lower signal-to-noise ratio bands being diacarded. The dataset contains 16 different land-cover classes. The false-color composite image (bands 50, 27, 17) is shown in Figure 7a.

(2) The Pavia University dataset was gathered by the Reflective Optics System Imaging Spectrometer (ROSIS-3) sensor in an urban site over the city of Pavia, Italy. There are 610 × 340

pixels and 103 spectral bands after 20 water absorption bands beingremoved. The ROSIS tensor generates the spectral bands in the wavelength ranging from 0.43μm to 0.86μm. There are 9 different land-cover classes, and the false-color image (bands 53, 31, 8) is shown in Figure 7b.

(3) The Salinas dataset was collected by the AVIRIS sensor over Salinas Valley, California. The dataset comprises 512 × 217 pixels. It has the spatial resolution of 3.7m per pixel. The sensor system generates 224 bands in wavelength range of 0.4–2.5μm. In the experiments, 204 bands are preserved after 20 water absorption bands being omitted. The image contains 16 classes. The false-color composite image (bands 50, 170, 190) is shown in Figure 7c.



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

Figure 7. The false-color composite images of (**a**) the Indian Pines; (**b**) the Pavia University; (**c**) the Salinas valley.

4.2. Experimental Setting

The performance of the proposed DDCNN method is compared with some state-of-the-art HSI classification approaches, which includes five representative deep learning-based methods, SAE [39], DBN [42], CNN [49], PPF-CNN [53], 3D-CNN [52], and a classical SVM method with radial basis function (RBF-SVM) [30]. The classification performance of all the methods is measured by three common measurements: overall accuracy (OA), average accuracy (AA), and kappa coefficient (Kappa) [72]. The experiments are impemented over 20 independent runs with a random division of training and test sets. The average classification accuracy and the corresponding standard deviation over 20 independent runs are calucated. When the training samples change by using the random selection, the sample augmentation, regional division, and DDCNN model are affected. In this way, the robustness of the proposed method is validated. All the experiments are carried out using Python language and TensorFlow [73] library on a NVIDIA 1080Ti graphics card. TensorFlow is an open source software library for numerical computation using data flow graphs.

For RBF-SVM, one-against-all strategy is used to deal with multi-classification. The penalty and gamma parameters in RBF-SVM are determined by five-fold cross validation. For SAE and DBN, the radius of the spatial neighborhood window is set as 7. As suggested by the literature [49], the input of the spatail window is set as 5 × 5. For PPF-CNN, the size of block window of neighboring pixels is set to the default value in [53]. For 3DCNN, the spatial window size of 3-D input is resized to 27 × 27 × 100 [52]. For DDCNN, the size of spatial window for dual architecture network will be investigated in the next subsection.

Besides, there are also several important parameters in the deep learning models, such as learning rate, epochs, and the number of layers. For the learning rate, we set all the models as 0.01. For the

epochs, SAE, DBN, CNN, and DDCNN are trained with 1000 epochs. We train PPF-CNN with 300 epochs while we train 3DCNN with 500 epochs. SAE and DBN consist of 4 hidden layers. CNN, 3DCNN, and DDCNN include 3 convolutional layers and 2 full connection layers, while PPF-CNN consists of 8 convolutional layers and 2 full connection layers.

4.3. Classification Results of Hyperspectral Datasets

(1) Classification Results of the Indian Pines Dataset: The Indian Pines dataset is randomly divided into 5% training set and 95% test set. The numbers of training and test samples for each class are listed in Table 2. Table 3 records the class-specific accuracy, overall accuracy (OA), average accuracy (AA), and Kappa of all seven methods. The best classification results in the seven algorithms are emphasized in gray regions. Compared with RBF-SVM, deep learning-based methods SAE, DBN, CNN, PPF-CNN, 3DCNN, and DDCNN obtain better classification results due to hierarchical nonlinear feature extraction. Compared with SAE and DBN, CNN, PPF-CNN, 3DCNN, and DDCNN are superior by making full use of the spatial information in HSI. Among the seven methods, DDCNN achieves the best classification results in the majority of classes due to the power feature extraction capability of dual-architecture CNN for various land-cover distributions. Furthermore, DDCNN improves the classification performance more than the best baseline by 4.1% in the OA index, 7.2% in the AA index, and 4.4% in the Kappa index.

Figure 8 shows the classification maps of the seven algorithms on the Indian Pines dataset. As shown in Figure 8b–d,f, there are massive noisy scattered points in SVM, SAE, DBN, and PPF-CNN, especially in the corn-notill, corn-mintill, soybean-notill, and soybean-mintill classes. Compared with these methods, CNN, 3DCNN, and DDCNN improve the region uniformity significantly. Howerer, edge over-smoothness occurs in the visual maps of CNN and 3DCNN. Compared with CNN and 3DCNN, DDCNN obtains better boundary localization of the soybean-notill and soybean-mintill classes.

Table 2. The 16 Classes of the Indian Pines dataset and the numbers of training and test samples for each class.

Class		Number of Samples	
No	**Name**	**Training**	**Test**
1	Alfalfa	2	42
2	Corn-notill	71	1286
3	Corn-mintill	42	746
4	Corn	12	213
5	Grass-pasture	24	435
6	Grass-trees	36	658
7	Grass-pasture-mowed	1	26
8	Hay-windrowed	24	430
9	Oats	1	18
10	Soybean-notill	49	874
11	Soybean-mintill	123	2209
12	Soybean-clean	30	533
13	Wheat	10	185
14	Woods	63	1139
15	Buildings-Grass-Trees-Drives	19	348
16	Stone-Steel-Towers	5	83
	Total	512	9225

Table 3. Classification results of RBF-SVM, SAE, DBN, CNN, PPF-CNN, 3DCNN, and DDCNN on the Indian Pines dataset.

Class	RBF-SVM	SAE	DBN	CNN	PPF-CNN	3DCNN	DDCNN
1	6.1±11.2	10.0±6.4	13.6±5.6	78.4±10.2	50.4±8.4	83.8±13.4	99.3±1.6
2	72.9±3.6	79.7±2.3	79.8±2.9	75.4±2.4	89.2±2.1	92.7±3.5	94.3±2.5
3	58.0±3.6	74.9±4.8	70.5±2.2	82.8±3.3	77.1±2.7	87.2±10.4	99.0±0.8
4	39.0±15.0	62.8±8.3	71.3±6.6	89.2±3.5	87.7±3.7	83.4±8.3	95.0±3.3
5	87.0±4.5	84.2±3.3	80.1±4.1	69.0±4.6	92.7±1.0	84.0±5.7	92.7±2.8
6	92.4±2.0	94.3±1.7	94.2±2.4	92.8±2.5	93.1±1.9	93.4±2.5	98.8±0.8
7	0±0	24.4±18.8	28.1±22.6	51.1±12.3	0±0	97.2±4.8	100.0±0.0
8	98.1±1.4	98.8±0.4	98.5±1.5	97.1±1.6	99.6±0.3	97.4±2.8	99.8±0.6
9	0±0	11.1±10.1	9.5±2.4	41.6±9.9	0±0	77.0±11.1	97.8±5.4
10	65.8±3.7	73.6±3.8	73.2±4.7	81.0±2.6	85.6±2.8	93.3±5.0	93.8±1.2
11	85.3±2.9	83.4±2.0	82.7±2.2	87.2±1.5	83.8±1.6	94.9±2.7	98.1±1.3
12	69.6±6.5	70.4±8.0	62.0±5.8	84.4±2.3	90.4±3.1	89.8±4.3	94.4±2.5
13	92.3±4.1	94.2±4.3	89.7±10.6	83.1±4.2	97.8±0.9	92.8±5.9	99.9±0.2
14	96.6±1.0	94.2±1.5	94.4±1.6	98.2±0.8	95.5±1.1	98.3±1.3	99.5±0.4
15	41.7±7.0	66.1±5.6	64.2±6.5	84.7±4.5	78.0±2.4	77.8±13.4	95.7±2.6
16	75.2±9.0	87.6±8.1	80.5±13.2	76.0±8.1	97.3±1.3	88.4±5.3	89.7±9.4
OA (%)	77.8±0.8	81.9±0.1	80.6±0.1	85.4±0.8	87.9±0.8	92.8±0.8	96.9±0.6
AA (%)	61.3±1.4	69.4±1.9	68.3±1.7	79.4±1.6	76.5±0.6	89.4±1.4	96.6±0.7
Kappa (%)	74.5±1.0	79.3±1.1	77.8±1.3	84.3±2.9	86.3±0.9	91.9±0.9	96.3±0.8

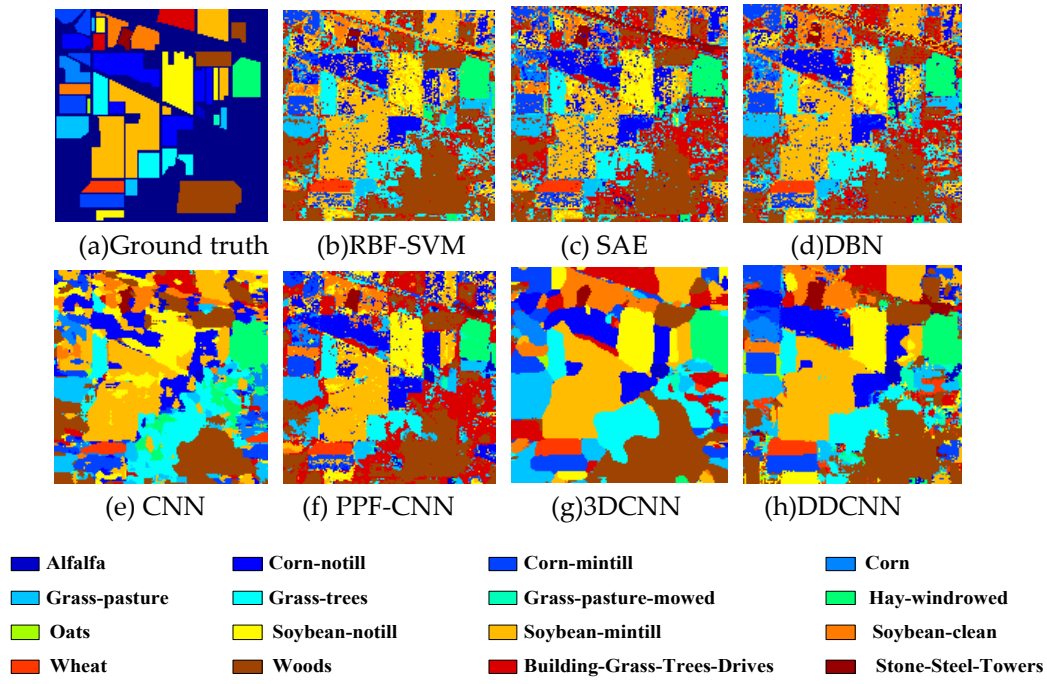


Figure 8. (**a**) Ground truth and (**b**–**h**) classification visual maps of the Indian Pines dataset by RBF-SVM, SAE, DBN, PPF-CNN, CNN, 3DCNN, and DDCNN, respectively.

(2) Classification results of the Pavia University dataset: The Pavia University dataset is randomly divided into a 3% training set and 97% test set. The numbers of training and test samples for each class are listed in Table 4. Table 5 records the classification results for the Pavia University dataset.

Table 4. 9 Classes of the Pavia University dataset and the numbers of training and test samples for each class.

Class		Number of Samples	
No	**Name**	**Training**	**Test**
1	Asphalt	199	6233
2	Meadows	559	17531
3	Gravel	63	1973
4	Trees	92	2880
5	Painted metal sheets	40	1265
6	Bare Soil	151	4727
7	Bitumen	40	1250
8	Self-Blocking Bricks	110	3462
9	Shadows	28	891
	Total	1282	40212

Table 5. Classification results of RBF-SVM, SAE, DBN, CNN, PPF-CNN, 3DCNN, and DDCNN on the Pavia University dataset.

Class	RBF-SVM	SAE	DBN	CNN	PPF-CNN	3DCNN	DDCNN
1	90.7±1.1	92.3±1.1	91.6±0.8	93.1±1.4	98.0±0.1	95.5±1.2	97.8±1.1
2	96.8±0.7	97.6±0.3	97.4±0.4	97.6±0.9	99.2±0.2	99.4±0..3	99.5±0.1
3	60.2±5.4	72.1±3.5	69.7±6.0	77.9±4.5	84.9±1.8	92.6±5.4	98.4±0.9
4	90.8±2.0	90.9±1.4	91.2±1.4	86.4±3.6	95.8±0.8	75.2±4.9	95.9±1.1
5	98.8±0.4	98.7±0.4	98.6±0.6	98.5±1.4	99.8±0.1	95.4±4.3	99.4±0.5
6	79.5±4.9	86.9±1.9	85.6±2.2	91.0±2.8	96.4±0.3	99.4±0.6	99.7±0.3
7	74.3±5.1	78.1±4.9	74.8±4.8	81.2±2.9	89.2±0.8	91.5±3.4	99.6±0.4
8	88.8±2.2	87.8±1.4	88.2±1.3	92.5±2.2	93.7±1.2	94.8±1.4	97.8±1.3
9	99.8±0.1	99.5±0.3	99.6±0.1	79.0±4.1	98.5±0.7	77.4±2.8	87.9±2.5
OA (%)	90.3±0.6	92.4±0.3	91.9±0.3	93.0±0.6	96.9±0.2	95.2±0.7	98.5±0.2
AA (%)	86.6±0.9	89.3±0.7	88.5±0.8	88.6±0.8	95.1±0.2	91.2±1.1	97.3±0.4
Kappa (%)	87.1±0.8	89.9±0.4	89.2±0.4	90.7±0.7	96.0±0.2	93.8±0.9	98.1±0.2

As shown in Table 5, compared with other methods, DDCNN gains a certain degree of improvement in most classes, especially in the gravel and bitumen classes. DDCNN improves 38.2% more than SVM in the gravel class, and improves 24.8% than DBN in the bitumen class. For all the classes, the proposed DDCNN method improves by 8.2%, 6.1%, 6.6%, 5.5%, 1.6%, and 3.3% more than the other six methods in the OA index.The visual classification maps of the Pavia University dataset are shown in Figure 9. As shown in Figure 9b–f, many samples belonging to the bitumen class are misclassified as the asphalt class because of similar spectral signatures. The proposed DDCNN method provides a better distinction for these two classes. Besides, the samples in the gravel class are misclassified as the class of the self-blocking bricks by SVM, SAE, and DBN, and as the class of the asphalt by 3DCNN. Compared with them, DDCNN obtains better classification performance for the gravel class. Compared with the other methods, DDCNN achieves better region uniformity in the bare soil class, and obtains better boundary localization in the gravel and bitumen classes.

(3) Classification results of the Salinas dataset: The Salinas dataset is randomly divided into 1% for training and 99% for testing. The numbers of training and test samples for each class are listed in Table 6. The classifcation results of all seven algorithms on the Salinas dataset are summarized in Table 7. It can be seen that many samples in the grapes_untrained and vinyard_untrained classes are misclassified by RBF-SVM, SAE, DBN, CNN, and PPF-CNN. Compared with these methods, DDCNN obviously improves the classification results. For the vinyard_untrained class, DDCNN improves by 42.6%, 20.7%, 27.1%, 16.5%, and 23.7%. For the broccoli_green_weeds_1 class, DDCNN achieves completely correct classification result. Among all the seven methods, DDCNN obtains the best classification performance by OA=98.8%, AA=98.6%, and Kappa=98.6%.

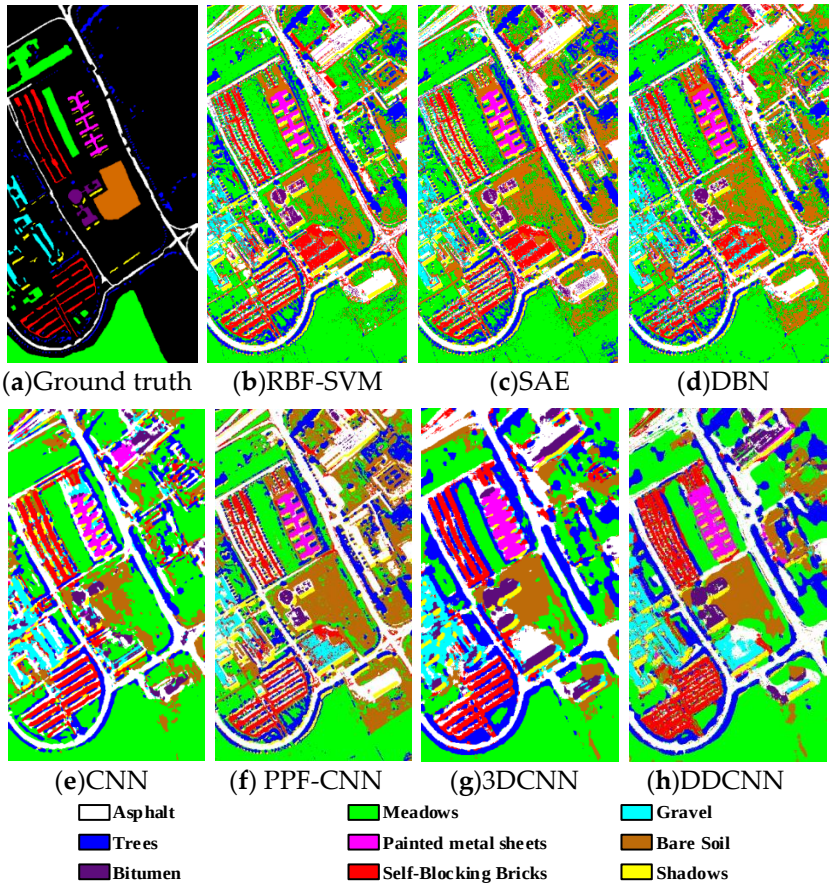


Figure 9. (**a**) Ground truth and (**b–h**) classification visual maps of the Pavia University dataset by RBF-SVM, SAE, DBN, CNN, PPF-CNN, 3DCNN, and DDCNN, respectively.

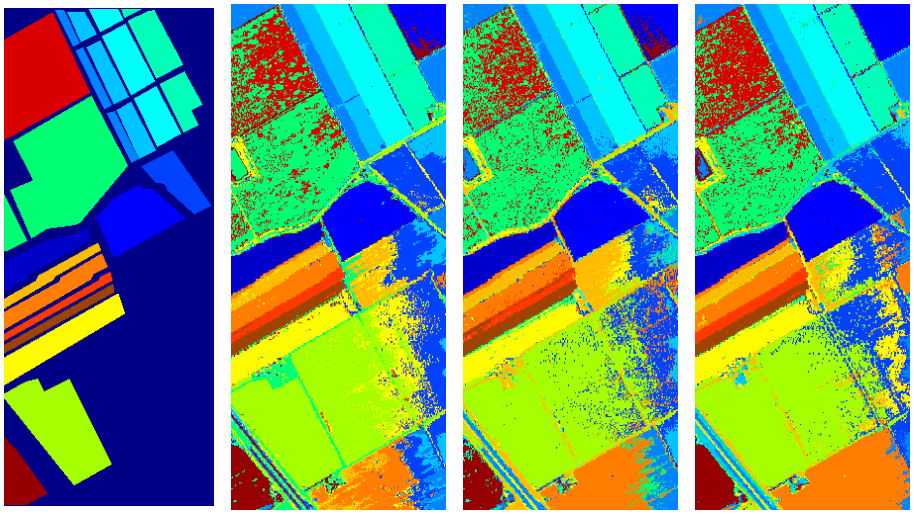
Table 6. The 16 Classes of the Salinas dataset and the numbers of training and test samples for each class.

Category		Number of samples	
No	**Name**	**Training**	**Test**
1	Brocoli_green_weeds_1	20	1969
2	Brocoli_green_weeds_2	37	3652
3	Fallow	20	1936
4	Fallow_rough_plow	14	1366
5	Fallow_smooth	27	2624
6	Stubble	40	3879
7	Celery	36	3507
8	Grapes_untrained	113	11045
9	Soil_vinyard_develop	62	6079
10	Corn_senesced_green	33	3212
11	Lettuce_romaine_4wk	11	1046
12	Lettuce_romaine_5wk	19	1889
13	Lettuce_romaine_6wk	9	898
14	Lettuce_romaine_7wk	11	1048
15	Vinyard_untrained	73	7122
16	Vinyard_vertical	18	1771
	Total	543	53043

Table 7. Classification results of RBF-SVM, SAE, DBN, CNN, PPF-CNN, 3DCNN, and DDCNN on the Salinas dataset.

Class	RBF-SVM	SAE	DBN	CNN	PPF-CNN	3DCNN	DDCNN
1	97.4±1.5	97.9±0.5	98.5±0.8	93.3±8.7	98.5±0.5	88.6±3.5	100.0±0.0
2	99.7±0.2	99.1±0.5	98.9±0.2	97.4±1.2	99.7±0.2	94.5±2.9	99.8±0.2
3	93.7±1.5	95.3±0.6	97.5±0.1	86.4±4.1	99.8±0.1	91.4±4.7	99.7±0.4
4	97.8±1.3	99.5±0.6	99.0±0.3	98.2±1.8	99.7±0.2	96.8±2.1	98.3±0.8
5	97.5±1.1	98.5±0.4	97.5±0.2	98.1±1.0	96.8±0.2	96.7±2.7	99.3±0.5
6	99.5±0.3	99.9±0.1	99.3±0.1	99.9±0.2	99.8±0.3	98.5±1.2	99.9±0.2
7	99.3±0.2	99.2±0.1	99.0±0.3	99.0±0.9	99.5±0.2	98.0±1.3	99.9±0.1
8	88.9±2.9	82.7±0.7	83.0±1.4	88.4±2.8	89.9±0.9	92.3±1.1	99.4±0.3
9	99.2±0.3	99.2±0.1	99.0±0.1	95.1±0.7	99.8±0.2	98.9±0.4	99.9±0.1
10	88.8±1.9	88.9±0.9	92.8±0.1	93.6±2.4	88.3±2.7	99.1±0.8	95.3±1.2
11	87.5±4.6	93.6±7.0	91.7±0.1	97.6±1.0	93.4±2.9	97.6±2.1	99.8±0.2
12	98.0±2.3	98.6±1.0	99.0±0.1	98.9±1.1	99.7±0.7	96.6±2.7	98.6±0.9
13	98.0±0.8	99.2±0.7	99.3±0.2	94.7±2.4	98.6±0.7	90.9±0.7	99.7±0.2
14	89.6±2.6	94.8±0.2	92.0±7.4	92.5±3.9	92.3±1.7	98.2±1.0	93.8±1.8
15	53.9±7.6	75.9±2.4	69.5±0.2	80.1±5.3	72.9±2.5	98.6±0.8	96.6±0.5
16	90.8±5.0	96.1±1.9	96.1±2.5	93.6±2.6	95.7±1.8	96.3±3.0	98.0±1.0
OA (%)	89.3±0.7	91.5±0.1	90.8±0.6	92.3±1.2	92.8±0.4	95.9±0.2	98.8±0.2
AA (%)	92.5±0.6	94.9±0.2	94.5±0.8	94.2±0.9	95.5±0.7	95.8±0.2	98.6±0.2
Kappa (%)	88.0±0.8	90.6±0.4	89.7±0.7	91.4±1.4	91.9±0.4	95.5±0.2	98.6±0.2

Figure 10 shows the classification visual maps of the seven algorithms on the Salinas dataset. As shown in Figure 10b–f, many samples belonging to the grapes_untrained and vinyard_untrained classes are confused by RBF-SVM, SAE, DBN, CNN, and PPF-CNN. Compared with them, 3DCNN and DDCNN provide better distinction for these two classes. Compared with 3DCNN, DDCNN obtains better boundary localization for these two classes.



(**a**) Ground truth (**b**) RBF-SVM (**c**) SAE (**d**) DBN

Figure 10. *Cont.*

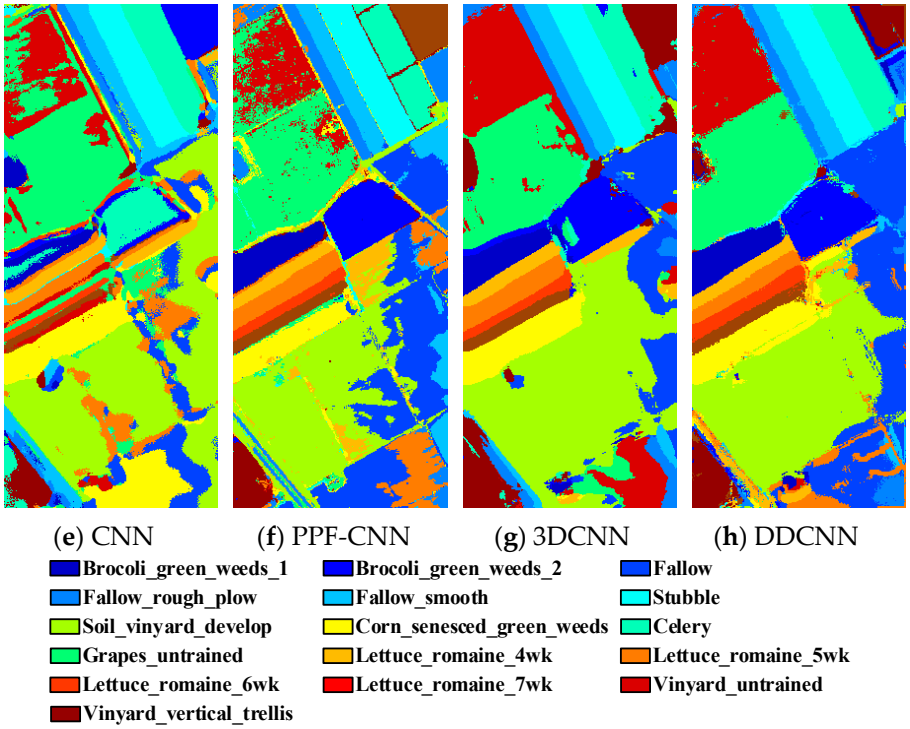


Figure 10. (**a**) Ground truth and (**b–h** classification visual maps of the Salinas dataset by RBF-SVM, SAE, DBN, CNN, PPF-CNN, 3DCNN, and DDCNN, respectively.

4.4. Investigation on Running Time and Parameters

Tables 8–10 list the training and test times of the seven methods on the Indian Pines, Pavia University, and Salinas datasets, respectively. Futhermore, the number of parameters involved with the seven methods are listed. As shown in Tables 8–10, compared with RBF-SVM, six deep learning-based methods, SAE, DBN, PPF-CNN, CNN, 3DCNN, and DDCNN, cost more training time due to heavily parameterized models. Among all the comparison methods, 3DCNN costs lots of time in the training process because three-dimensional convolution operation involves a large number of parameters. PPF-CNN is time-consuming due to the expansion of a large number of training samples, especially when the number of training samples is large. DDCNN involve two CNN architectures, which cost more time than CNN but less time than 3DCNN and PPF-CNN. The number of parameters for DDCNN is almost 376,000, where multi-scale CNN has nearly 347,000 paremeters and fine-grained CNN has nearly 29,000 parameters. In the testing procedure, DDCNN is more time-consuming than SAE, DBN, and CNN due to the computation burden in double CNN architectures. Compared with PPF-CNN and 3D-CNN, DDCNN has obvious advantage because PPF-CNN uses the voting strategy with the adjacent samples and 3D-CNN uses a complex 3D convolution operation. DDCNN costs 0.7s, 2.3s, and 4.7s on the Indian Pines, Pavia University, and Salinas datasets, respectively.

Table 8. Running time and Parameters of RBF-SVM, SAE, DBN, CNN, PPF-CNN, 3DCNN, and DDCNN on the Indian Pines dataset.

Dataset	Method	Training Time (s)	Test Time (s)	Parameters
Indian Pines	RBF-SVM	0.4±0.1	1.2±0.1	200
	SAE	76.3±8.4	0.2±0.1	26160
	DBN	114.3±20.1	0.2±0.1	24060
	CNN	220.7±27.9	0.5±0.1	81408
	PPF-CNN	2056.0±36.7	5.3±0.3	61870
	3DCNN	2690.2±57.9	16.0±0.1	44961792
	DDCNN	587.2±22.7	0.7±0.1	376932

Table 9. Running time and Parameters of RBF-SVM, SAE, DBN, CNN, PPF-CNN, 3DCNN, and DDCNN on the Pavia University dataset.

Dataset	Method	Training Time (s)	Test Time (s)	Parameters
Pavia University	RBF-SVM	0.5±0.1	3.5±0.1	200
	SAE	82.2±5.3	0.3±0.1	19920
	DBN	147.0±10.6	0.4±0.2	21420
	CNN	371.8±15.3	1.2±0.1	61249
	PPF-CNN	4367.9±29.5	7.2±0.4	61310
	3DCNN	1979.0±12.6	31.4±5.5	5866224
	DDCNN	682.1±10.6	2.3±0.1	375140

Table 10. Running time and Parameters of RBF-SVM, SAE, DBN, CNN, PPF-CNN, 3DCNN, and DDCNN on the Salinas dataset.

Dataset	Method	Training Time (s)	Test Time (s)	Parameters
Salinas	RBF-SVM	0.4±0.1	2.7±0.1	204
	SAE	70.1±2.4	0.6±0.1	26160
	DBN	102.6±9.1	0.5±0.2	24060
	CNN	165.1±2.1	0.7±0.1	82216
	PPF-CNN	1940.1±17.4	64.6±1.5	61870
	3DCNN	1157.7±25.7	28.1±0.4	5867520
	DDCNN	657.2±20.6	4.7±0.2	376932

4.5. Sensitivity to the Number of Training Samples

Figure 11 shows the classification performance with different numbers of training samples. The classification performance of deep learning-based methods depends on the number of training samples greatly. Thus, it's necessary to investigate the sensitivity to the number of training samples. In the experiment, the number of training samples per class is changed from 1% to 9% with an interval of 2% on the Indian Pines dataset, 1% to 5% with an interval of 1% on the Pavia University dataset, and 1% to 3% with an interval of 0.5% on the Salinas dataset. Generally, deep learning-based methods are usually heavily parameterized and a large number of training samples are required to guarantee the performance. When the ratio of training samples is larger than 9% on the Indian Pines, 5% on the Pavia University, and 3% on the Salinas, the training samples are sufficient to estimate the models. CNN-based methods, CNN, PPF-CNN, 3DCNN, and DDCNN, perform better than the other three methods. When the ratio of training samples decreases, the classification performance of all the seven algorithms declines. In this case, deep learning-based methods SAE, DBN, and CNN have no obvious advantage over RBF-SVM. Compared with them, 3D-CNN, PPF-CNN, and DDCNN show better classification performance for the small-sized sample set. Among these methods, DDCNN consistently provides superior performance with different ratios of training samples. DDCNN improves by at least 6.8%, 5.6%, and 2.9% on the Indian Pines, Pavia University, and Salinas datasets, respectively, when the ratio of training sample is 1%. Thus, DDCNN is a better choice when the number of training samples is limited.

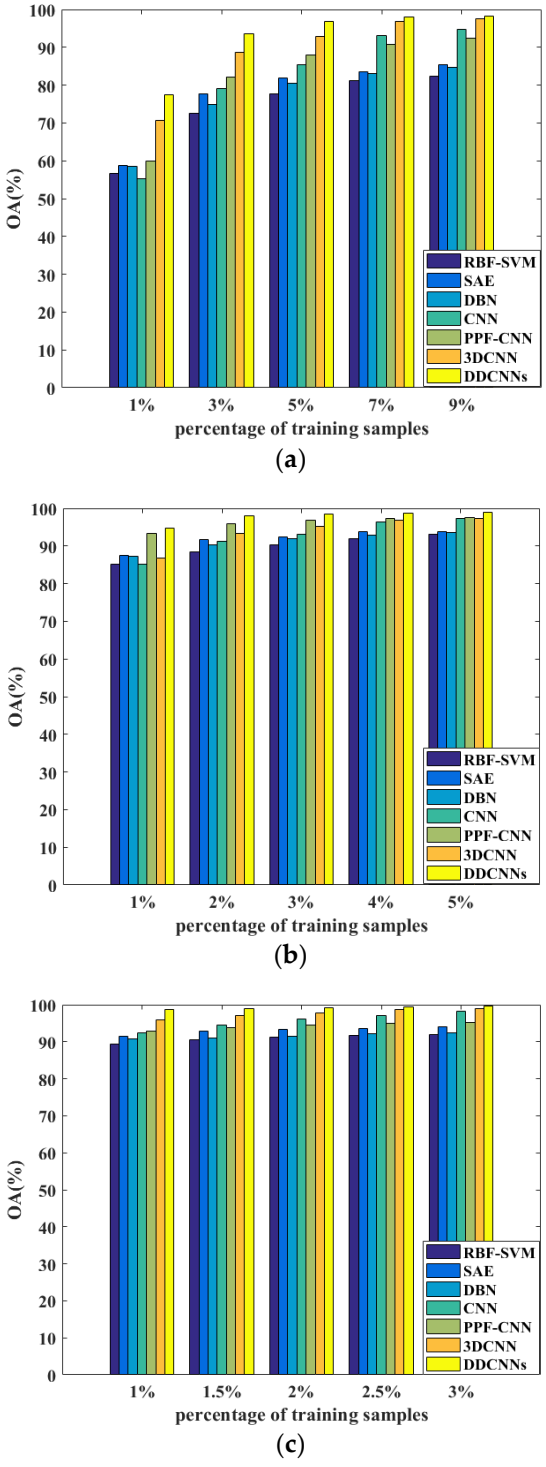


Figure 11. The OA results of RBF-SVM, SAE, DBN, CNN, PPF-CNN, 3DCNN, and DDCNN with different ratios of training samples on the (**a**) Indian Pines, (**b**) Pavia University, and (**c**) Salinas datasets.

4.6. Comparison with Other Classification Techniques

Table 11 shows the classification results of different methods on three HSI datasets. RPCA-RNN obtains better classification results than CNN because RPCA-RNN makes full use of spatial information. Compared with CNN and RPCA-CNN, DCNN improves the classification performance by extracting joint spatial-spectral features. Compared with RPCA-CNN and DCNN, DDCNN obtains better classification results by using divide-and-conquer dual-architecture CNN and effective sample

augmentation. It increases by 17.4% and 3.5% on the Indian Pines datasets, 19.7% and 7.1% on the Pavia University dataset, and 7.1% and 4.3% on the Salinas dataset in terms of OA index.

Table 11. Classification results of CNN, RPCA-CNN, DCNN, and DDCNN on the Indian Pines, Pavia University, and Salinas Datasets.

Data set	Classification Index	CNN	RPCA-CNN	DCNN	DDCNN
Indian Pines Dataset	OA (%)	85.4±0.8	88.6±0.6	93.4±0.5	96.9±0.6
	AA (%)	79.4±1.6	82.6±2.3	89.5±1.7	96.6±0.7
	Kappa (%)	84.3±2.9	86.1±0.7	92.5±0.5	96.3±0.8
Pavia University Dataset	OA (%)	93.0±0.6	94.2±0.2	95.7±0.8	98.5±0.2
	AA (%)	88.6±0.8	91.6±0.2	95.3±0.8	97.3±0.4
	Kappa (%)	90.7±0.7	92.4±0.5	95.2±0.9	98.1±0.2
Salinas Dataset	OA (%)	92.3±1.2	92.9±0.6	94.5±0.6	98.8±0.2
	AA (%)	94.2±0.9	94.2±0.9	94.5±0.6	98.6±0.2
	Kappa (%)	91.4±1.4	91.4±1.4	93.9±0.7	98.6±0.2

4.7. Effectiveness Analysis to Dual-Architecture CNN and Data Augmentation in DDCNN

To verify the effectiveness of data augmentation, we have added the proposed method without data augmentation (DDCNN-WDA) as the comparison method. To validate the structure effectiveness of the proposed dual-architecture CNN method, a multi-scale CNN (MCNN) and a fine-gained CNN (FCNN) have been added as the comparison methods. The experimental results on the Indian Pines, Pavia University, and Salinas datasets are recorded in Table 12.

Table 12. Classification results of DDCNN, MCNN, FCNN, and DDCNN-WDA on the Indian Pines, Pavia University, and Salinas Datasets.

Data set	Classification Index	DDCNN	FCNN	MCNN	DDCNN-WDA
Indian Pines Dataset	OA (%)	96.9±0.6	93.3±0.7	95.8±0.5	95.9±0.7
	AA (%)	96.6±0.7	90.6±2.1	92.6±1.9	93.2±2.0
	Kappa (%)	96.3±0.8	92.4±0.8	95.3±0.6	95.3±0.8
Pavia University Dataset	OA (%)	98.5±0.2	97.4±0.2	97.8±0.4	97.7±0.9
	AA (%)	97.3±0.4	95.9±0.5	95.5±1.1	95.9±0.9
	Kappa (%)	98.1±0.2	96.6±0.3	97.1±0.5	96.9±1.3
Salinas Dataset	OA (%)	98.8±0.2	96.3±0.7	97.1±1.5	98.4±0.3
	AA (%)	98.6±0.2	97.4±0.5	97.1±1.7	97.7±0.6
	Kappa (%)	98.6±0.2	95.9±0.9	96.8±1.7	97.9±1.4

As shown in Table 12, compared with FCNN, DDCNN increases by 3.6%, 1.1%, and 2.5% on the Indian Pines, Pavia University, and Salinas datasets. Compared with MCNN, DDCNN increases by 1.1%, 0.7%, and 1.7% on three HSI datasets. It is shown that dual-architecture is more effective than single network architecture for HSI classification. DDCNN exploits dual-architecture CNN to improve the classification performance of HSIs. Compared with DDCNN-WDA, DDCNN increases by 1.0%, 0.8%, and 0.4% on the Indian Pines, Pavia University, and Salinas datasets. It is shown that data augmentation is effective for HSI classification. DDCNN improves the classification performance of HSIs by exploiting the data augmentation.

4.8. Analysis of Free Parameters in DDCNN

There are two important parameters w_1 and w_2 in DDCNN; w_1 and w_2 represent the size of spatial window in multi-scale CNN and fine-grained CNN, respectively. In Figure 12, w_1 is set to [23, 25, 27, 29, 31], while w_2 is set to [1, 3, 5, 7, 9]; w_1 and w_2 control the input size of samples in the homogeneous and heterogeneous regions. Figure 12a–c shows the OA results of DDCNN on the Indian Pines,

Pavia University, and Salinas datasets under different parameters w_1 and w_2. As shown in Figure 12, when w_1 and w_2 are selected as 27 and 7 on the Indian Pines, 31 and 9 on the Pavia University, and 31 and 9 on the Salinas, the classification performance reaches the peak values. The Pavia University and Salinas dataset have higher spatial resolution than the Indian Pines dataset. Therefore, the sizes of w_1 and w_2 in the Pavia University and Salinas datasets are larger than that in the Indian Pines dataset.

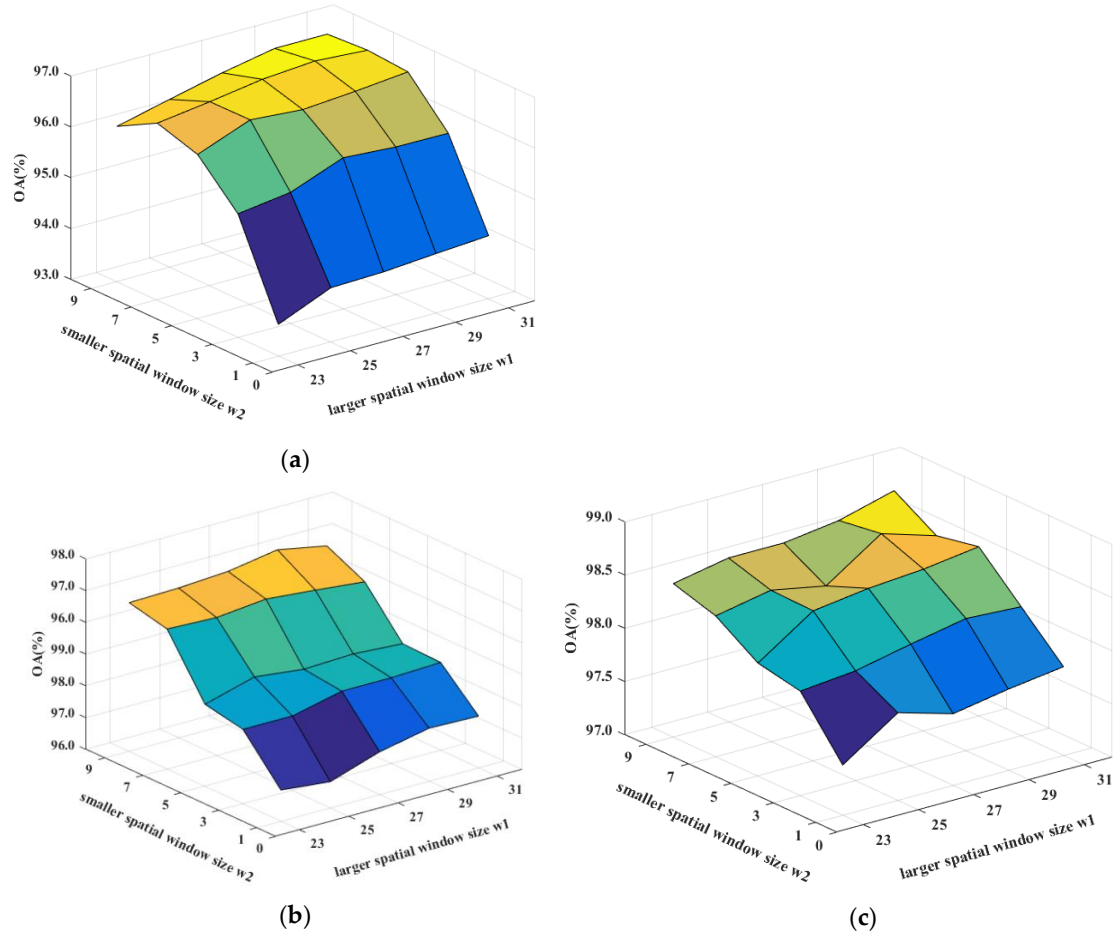


Figure 12. Sensitivity analysis to the spatial window sizes w_1 and w_2 for DDCNN on (**a**) the Indian Pines, (**b**) the Pavia University, and (c) the Salinas datasets.

The depth of the network plays an important role because it determines the quality of extracted features. Table 13 shows the classification results of DDCNN as the number of convolutional layers increases from 1 to 5. The experimental results show that the model achieves the best classification results when 4 convolutional layers are chosen for hyperspectral datasets. When the number of layers is large enough, the model extracts abstract and invariant features.

The number of superpixel is an important free parameter. The superpixel segmentation is utilized in the regional division and data augmentation of DDCNN. As shown in Table 14, DDCNN obtains the best classification performance when the number of superpixels is set as 100 on the Indian Pines dataset and Salinas dataset, and 1000 on the Pavia University dataset. The number of superpixels on the Pavia University dataset is larger than that on other datasets due to more complex distribution on the Pavia University dataset. When the number of superpixels is too small, the same superpixel may contain different classes. In this case, the classification results would deteriorate due to misdivision of homogeneous and heterogeneous regions. On the contrary, when the number of superpixels is too large, fewer unlabeled samples are pre-labeled to augment the data. In this case, DDCNN has limited ability to alleviate the overfitting problem.

Table 13. The sensitivity analysis of numbers of convolutional layers.

Dataset	Classification Index	the Number of Convolutional Layers				
		1	**2**	**3**	**4**	**5**
Indian Pines Dataset	OA (%)	93.6±0.4	95.5±0.4	96.0±0.1	96.9±0.6	96.4±0.3
	AA (%)	91.6±1.1	94.5±0.3	95.5±0.3	96.6±0.7	95.3±0.7
	Kappa (%)	92.8±0.4	94.9±0.4	95.4±0.1	96.3±0.8	95.8±0.4
Pavia University Dataset	OA (%)	95.8±0.1	97.4±0.2	98.7±0.1	98.5±0.2	98.4±0.1
	AA (%)	95.1±0.4	96.7±0.3	98.3±0.1	97.3±0.4	98.2±0.2
	Kappa (%)	94.5±0.1	96.6±0.3	98.4±0.2	98.1±0.2	97.8±0.1
Salinas Dataset	OA (%)	94.3±0.2	95.5±0.6	98.5±0.2	98.8±0.2	98.6±0.2
	AA (%)	96.2±0.3	97.1±0.5	98.5±0.2	98.6±0.2	98.5±0.3
	Kappa (%)	93.7±0.3	95.1±0.7	98.4±0.2	98.6±0.2	98.5±0.2

Table 14. The sensitivity analysis of numbers of superpixels in DDCNN.

Dataset	Classification Index	The Number of Superpixels				
		50	**100**	**500**	**1000**	**5000**
Indian Pines Dataset	OA (%)	94.7±0.4	96.9±0.6	96.3±0.3	93.7±0.3	92.7±0.5
	AA (%)	92.1±1.6	96.6±0.7	95.4±0.9	91.6±1.8	91.1±1.3
	Kappa (%)	94.0±0.4	96.3±0.8	95.8±0.3	92.8±0.4	91.4±0.9
Pavia University Dataset	OA (%)	97.1±0.2	97.5±0.2	98.3±0.1	98.5±0.2	97.3±0.3
	AA (%)	95.5±0.4	96.9±0.2	97.8±0.1	97.3±0.4	96.3±0.2
	Kappa (%)	96.4±0.3	96.7±0.3	97.8±0.1	98.1±0.2	96.4±0.5
Salinas Dataset	OA (%)	97.2±0.4	98.8±0.2	97.3±0.4	95.9±0.9	94.9±0.5
	AA (%)	96.5±1.0	98.6±0.2	96.7±0.4	94.5±1.6	92.9±0.8
	Kappa (%)	96.9±0.5	98.6±0.2	97.0±0.4	95.5±0.9	94.9±0.5

4.9. Analysis of the Thresholds in DDCNN

There are two thresholds, T_k and T_{π_v}, involved in the proposed method. T_k is a threshold involved in the regional division with non-local decision. The threshold T_k is not empirically set. It can be calculated by the equation $T_k = \min\{SS(\pi_v(x_i), \pi'_v(x_j)) | x_i, x_j \in \psi_k\}$. T_k is the minimum value of similarities between any two superpixels containing the training samples of the kth category. For each class, an adaptive threshold T_k can be obtained by considering all the training samples of this class. When the value of T_k is too large or small, the classification performance would degrade due to misdivision of homogeneous and heterogeneous regions. Compared with empirical setting, the proposed adaptive calculation is a better choice due to considering data distribution.

T_{π_v} is a threshold involved in the data augmentation. It is calculated as the minimum value of the similarities between any two training samples in the superpixel π_v. For three hyperspectral datasets, T_{π_v} is calculated as 0.921, 0.903, and 0.915 in the experiment. We have added the analysis of classification performance under different thresholds T_{π_v} in Figure 13. In Figure 13, the OA results of DDCNN on three hyperspectral datasets are shown as T_{π_v} increases from 0.5 to 1.0. When the value of T_{π_v} is too large, the spatial constraint of sample augmentation becomes strict. Fewer unlabeled samples are selected to pre-label. In this case, DDCNN has limited ability to alleviate the overfitting problem. Conversely, when the value of T_{π_v} is too small, unlabeled samples having low confidence may be selected. In this case, pre-labeled unlabeled samples would deteriorate the classification performance. When T_{π_v} is in the range of [0.88, 0.93], DDCNN can obtain promising classification results on three hyperspectral datasets. On three hyperspectral datasets, T_{π_v} is calculated as 0.921, 0.903, and 0.915 in the experiment. It can be seen that the calculated values of T_{π_v} fall within this range.

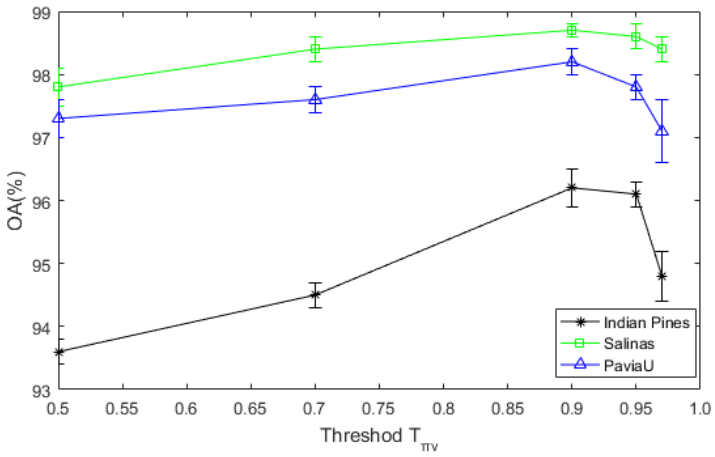


Figure 13. The sensitivity analysis of DDCNN to the threshold T_{π_v}.

5. Conclusions

In this paper, a novel divide-and-conquer dual-architecture CNN (DDCNN) method is proposed for HSI classification. In DDCNN, a regional division method based on local and non-local decisions is designed to divide the HSIs into homogeneous and heterogeneous regions, respectively. A multi-scale CNN architecture and a fine-grained CNN architecture are constructed to learn spectral-spatial features on the homogeneous and heterogeneous regions. Dual-architecture CNN guarantees region uniformity and edge preservation of HSI classification simultaneously. Moreover, to alleviate the problem of insufficient training samples, the unlabeled samples with high confidence are selected under adaptive spatial constraints. The experimental results on several hyperspectral datasets demonstrated the effectiveness of the proposed method for HSI classification.

In the future, more varied CNN architecture will be considered in DDCNN for complex land-cover distributions in HSIs.

Author Contributions: Conceptualization, J.F. and L.W.; Methodology, J.F. and L.W.; Software, H.Y. and L.W.; Validation, H.Y., L.W. and L.J.; Formal Analysis, X.Z.; Investigation, X.Z.; Resources, L.J.; Data Curation, L.J.; Writing-Original Draft Preparation, J.F. and L.W.; Writing-Review & Editing, J.F. and L.W.; Visualization, H.Y.; Supervision, L.J. and J.F.; Project Administration, X.Z.; Funding Acquisition, L.J. and J.F.

Funding: This work was supported in part by the National Natural Science Foundation of China under Grant 61871306, Grant 61772400, and Grant 61773304, in part by the Project Funded by China Postdoctoral Science Foundation under Grant 2015M570816 and Grant 2016T90892, in part by the State Key Program of National Natural Science of China under Grant 61836009, in part by the Open Research Fund of Key Laboratory of Spectral Imaging Technology, Chinese Academy of Sciences, under Grant LSIT201803D, in part by the Fundamental Research Funds for the Central Universities under Grant JBX181707, in part by the Postdoctoral Research Program in Shaanxi Province of China, and in part by the Joint Fund of the Equipment Research of Ministry of Education.

Acknowledgments: The authors would like to thank the Editor who handled our paper and the three anonymous reviewers for providing truly outstanding comments and suggestions that significantly helped us improve the technical quality and presentation of our paper.

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

References

1. Chang, C.I. *Hyperspectral Data Exploitation: Theory and Applications*; Wiley: Hoboken, NJ, USA, 2007; pp. 441–442, ISBN 9780471746973.
2. Gevaert, C.M.; Suomalainen, J.; Tang, J.; Kooistra, L. Generation of spectral–temporal response surfaces by combining multispectral satellite and hyperspectral UAV imagery for precision agriculture applications. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2015**, *8*, 3140–3146. [CrossRef]
3. Makki, I.; Younes, R.; Francis, C.; Bianchi, T.; Zucchetti, M. A survey of landmine detection using hyperspectral imaging. *ISPRS J. Photogramm. Remote Sens.* **2017**, *124*, 40–53. [CrossRef]

4. Brown, A.J.; Walter, M.R.; Cudahy, T.J. Hyperspectral imaging spectroscopy of a Mars analogue environment at the North Pole Dome, Pilbara Craton, Western Australia. *Aust. J. Earth Sci.* **2005**, *52*, 353–364. [CrossRef]
5. Meer, F.V.D. Analysis of spectral absorption features in hyperspectral imagery. *Int. J. Appl. Earth Obs. Geoinf.* **2004**, *5*, 55–68. [CrossRef]
6. Yuen, P.W.; Richardson, M. An introduction to hyperspectral imaging and its application for security, surveillance and target acquisition. *Imaging Sci. J.* **2010**, *58*, 241–253. [CrossRef]
7. Malthus, T.J.; Mumby, P.J. Remote sensing of the coastal zone: An overview and priorities for future research. *Int. J. Remote Sens.* **2003**, *24*, 2805–2815. [CrossRef]
8. Bioucas-Dias, J.M.; Plaza, A.; Camps-Valls, G.; Scheunders, P.; Nasrabadi, N.; Chanussot, J. Hyperspectral remote sensing data analysis and futurechallenges. *IEEE Geosci. Remote Sens. Mag.* **2013**, *1*, 6–36. [CrossRef]
9. Hughes, G.F. On the mean accuracy of statistical pattern recognizers. *IEEE Trans. Inf. Theory* **1968**, *14*, 55–63. [CrossRef]
10. Kang, X.D.; Xiang, X.L.; Li, S.T.; Benediktsson, J.A. PCA-based edge-preserving features for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 7140–7151. [CrossRef]
11. Agarwal, A.; El-Ghazawi, T.; El-Askary, H.; Le-Moigne, J. Efficient hierarchical-PCA dimension reduction for hyperspectral imagery. In Proceedings of the IEEE International Symposium on Signal Processing and Information Technology, Giza, Egypt, 15–18 December 2007; pp. 353–356.
12. Villa, A.; Benediktsson, J.A.; Chanussot, J.; Jutten, C. Hyperspectral image classification with independent component discriminant analysis. *IEEE Trans. Geosci. Remote Sens.* **2011**, *49*, 4865–4876. [CrossRef]
13. Wang, J.; Chang, C.-I. Independent component analysis-based dimensionality reduction with applications in hyperspectral image analysis. *IEEE Trans. Geosci. Remote Sens.* **2006**, *44*, 1586–1600. [CrossRef]
14. Xu, C.; Lu, C.; Gao, J.; Zheng, W.; Wang, T.; Yan, S. Discriminative analysis for symmetric positive definite matrices on lie groups. *IEEE Trans. Circuits Syst. Video Technol.* **2015**, *25*, 1576–1585. [CrossRef]
15. Chen, P.H.; Jiao, L.C.; Liu, F. Dimensionality reduction of hyperspectral imagery using sparse graph learning. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2017**, *10*, 1165–1181. [CrossRef]
16. Bandos, T.V.; Bruzzone, L.; Camps-Valls, G. Classification of hyperspectral images with regularized linear discriminant analysis. *IEEE Trans. Geosci. Remote Sens.* **2009**, *47*, 862–873. [CrossRef]
17. Rajadell, O.; García-Sevilla, P.; Pla, F. Spectral–spatial pixel characterization using gabor filters for hyperspectral image classification. *IEEE Geosci. Remote Sens. Lett.* **2013**, *10*, 860–864. [CrossRef]
18. Shen, L.; Jia, S. Three-dimensional Gabor wavelets for pixel-based hyperspectral imagery classification. *IEEE Trans. Geosci. Remote Sens.* **2011**, *49*, 5039–5046. [CrossRef]
19. Tang, Y.Y.; Lu, Y.; Yuan, H. Hyperspectral image classification based on three-dimensional scattering wavelet transform. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 2467–2480. [CrossRef]
20. Fauvel, M.; Benediktsson, J.A.; Chanussot, J.; Sveinsson, J.R. Spectral and spatial classification of hyperspectral data using SVMs and morphological profiles. *IEEE Trans. Geosci. Remote Sens.* **2008**, *46*, 3804–3814. [CrossRef]
21. Mura, M.D.; Benediktsson, J.A.; Waske, B.; Bruzzone, L. Extended profiles with morphological attribute filters for the analysis of hyperspectral data. *Int. J. Remote Sens.* **2010**, *31*, 5975–5991. [CrossRef]
22. Ghamisi, P.; Benediktsson, J.A.; Cavallaro, G.; Plaza, A. Automatic framework for spectral–spatial classification based on supervised feature extraction and morphological attribute profiles. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 2147–2160. [CrossRef]
23. Zhi, H.; Li, J.; Liu, K.; Liu, L.; Tao, H. Kernel low-rank multitask learning in variational mode decomposition domain for multi-/hyperspectral classification. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 4193–4208.
24. Cariou, C.; Chehdi, K. Unsupervised nearest neighbors clustering with application to hyperspectral images. *IEEE J. Sel. Top. Signal Process.* **2015**, *9*, 1105–1116. [CrossRef]
25. Khodadadzadeh, M.; Li, J.; Plaza, A.; Bioucas-Dias, J.M. A Subspace-Based Multinomial Logistic Regression for Hyperspectral Image Classification. *IEEE Geosci. Remote Sens. Lett.* **2014**, *11*, 2105–2109. [CrossRef]
26. Li, W.; Chen, C.; Su, H.; Du, Q. Local binary patterns and extreme learning machine for hyperspectral imagery classification. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 3681–3693. [CrossRef]
27. Liu, J.; Wu, Z.; Wei, Z.; Xiao, L.; Sun, L. Spatial-spectral kernel sparse representation for hyperspectral image classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2013**, *6*, 2462–2471. [CrossRef]
28. Chen, Y.; Nasrabadi, N.M.; Tran, T.D. Hyperspectral image classification using dictionary-based sparse representation. *IEEE Trans. Geosci. Remote Sens.* **2011**, *49*, 3973–3985. [CrossRef]

29. Wang, Q.; He, X.; Li, X. Locality and Structure Regularized Low Rank Representation for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens. (T-GRS)* **2019**, *57*, 911–923. [CrossRef]
30. Melgani, F.; Bruzzone, L. Classification of hyperspectral remote sensing images with support vector machines. *IEEE Trans. Geosci. Remote Sens.* **2004**, *42*, 1778–1790. [CrossRef]
31. Gualtieriand, J.A.; Chettri, S. Support vector machines for classification of hyperspectral data. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Honolulu, HI, USA, 24–28 July 2000; pp. 813–815.
32. Hinton, G.E.; Osindero, S.; Teh, Y. A fast learning algorithm for deep belief nets. *Neural Comput.* **2016**, *18*, 1527–1554. [CrossRef] [PubMed]
33. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In Proceedings of the Neural Information Processing Systems Conference NIPS, Lake Tahoe, NV, USA, 3–6 December 2012.
34. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. In Proceedings of the International Conference on Learning Representations ICLR, San Diego, CA, USA, 7–9 May 2015.
35. 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 CVPR, Boston, MA, USA, 7–12 June 2015.
36. 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, 27–30 June 2016; pp. 770–778.
37. Huang, G.; Liu, Z.; Weinberger, K.Q.; van der Maaten, L. Densely connected convolutional networks. *arXiv* **2016**, arXiv:1608.08993.
38. Chen, Y.S.; Lin, Z.H.; Zhao, X. Deep learning-based classification of hyperspectral data. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 2094–2107. [CrossRef]
39. Ng, A. Sparse autoencoder. *CS294A Lect. Notes* **2011**, *72*, 1–9.
40. Vincent, P.; Larochelle, H.; Lajoie, I.; Bengio, Y.; Manzagol, P.-A. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.* **2010**, *11*, 3371–3408.
41. Jia, K.; Sun, L.; Gao, S.; Song, Z.; Shi, B.E. Laplacian auto-encoders: An explicit learning of nonlinear data manifold. *Neurocomputing* **2015**, *160*, 250–260. [CrossRef]
42. 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]
43. Ghamisi, P.; Chen, Y.S.; Zhu, X.X. A self-improving convolution neural network for the classification of hyperspectral data. *IEEE Trans. Geosci. Remote Sens.* **2016**, *13*, 1537–1541. [CrossRef]
44. Paoletti, M.E.; Haut, J.M.; Plaza, J.; Plaza, A. A new deep convolutional neural network for fast hyperspectral image classification. *ISPRS J. Photogramm. Remote Sens.* **2018**, *145*, 120–147. [CrossRef]
45. Zhao, W.Z.; Du, S.H. Learning multiscale and deep representations for classifying remotely sensed imagery. *ISPRS J. Photogramm. Remote Sens.* **2016**, *113*, 155–165. [CrossRef]
46. Jia, P.Y.; Zhang, M.; Yu, W.B. Convolutional neural networks based classification for hyperspectral data. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Beijing, China, 10–15 July 2016.
47. Yu, S.Q.; Jia, S.; Xu, C.Y. Convolutional neural networks for hyperspectral image classification. *Neurocomputing* **2017**, *219*, 88–98. [CrossRef]
48. Zhou, Y.C.; Wei, Y.T. Learning hierarchical spectral–spatial features for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2016**, *46*, 1667–1678. [CrossRef] [PubMed]
49. Hu, W.; Huang, Y.; Wei, L.; Zhang, F.; Li, H. Deep convolutional neural networks for hyperspectral image classification. *J. Sens.* **2015**. [CrossRef]
50. Makantasis, K.; Karantzalos, K.; Doulamis, A.; Doulamis, N. Deep supervised learning for hyperspectral data classification through convolutional neural networks. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium IGARSS, Milan, Italy, 26–31 July 2015; pp. 4959–4962.
51. Zhang, H.; Li, Y.; Zhang, Y.; Shen, Q. Spectral-spatial classification of hyperspectral imagery using a dual-channel convolutional neural network. *Remote Sens. Lett.* **2017**, *8*, 438–447. [CrossRef]
52. Chen, Y.S.; Jiang, H.L.; Li, C.Y. Deep feature extraction and classification of hyperspectral images based on convolutional neural networks. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 6232–6251. [CrossRef]

53. Li, W.; Wu, G.D.; Zhang, F. Hyperspectral image classification using deep pixel-pair features. *IEEE Trans. Geosci. Remote Sens.* **2016**, *55*, 844–853. [CrossRef]
54. Wang, Q.; Yuan, Z.; Li, X. GETNET: A general end-to-end two-dimensional CNN framework for hyperspectral image change detection. *IEEE Trans. Geosci. Remote Sens. (T-GRS)* **2019**, *57*, 3–13. [CrossRef]
55. Makantasis, K.; Doulamis, A.D.; Doulamis, N.D.; Nikitakis, A. Tensor-based classification models for hyperspectral data analysis. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 6884–6898. [CrossRef]
56. Anaissi, A.; Braytee, A.; Naji, M. Gaussian kernel parameter optimization in one-class support vector machines. In Proceedings of the International Joint Conference on Neural Networks (IJCNN), Rio, Brazil, 8–13 July 2018; pp. 1–8.
57. Ghamisi, P.; Plaza, J.; Chen, Y.; Li, J.; Plaza, A.J. Advanced spectral classifiers for hyperspectral images: A review. *IEEE Geosci. Remote Sens. Mag.* **2017**, *5*, 8–32. [CrossRef]
58. Li, J.; Zhao, X.; Li, Y.; Du, Q.; Xi, B.; Hu, J. Classification of Hyperspectral Imagery Using a New Fully Convolutional Neural Network. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 292–296. [CrossRef]
59. Wang, Q.; Liu, S.; Chanussot, J.; Li, X. Scene Classification with Recurrent Attention of VHR Remote Sensing Images. *IEEE Trans. Geosci. Remote Sens. (T-GRS)* **2019**, *57*, 1155–1167. [CrossRef]
60. Kruger, N.; Janssen, P.; Kalkan, S.; Lappe, M.; Leonardis, A.; Piater, J.; Rodriguez-Sanchez, A.J.; Wiskott, L. Deep hierarchies in primate visual cortex what can we learn for computer vision. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 1847–1871. [CrossRef] [PubMed]
61. Achanta, R.; Shaji, A.; Smith, K.; Lucchi, A.; Fua, P.; Süsstrunk, S. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 2274–2282. [CrossRef] [PubMed]
62. Liu, M.-Y.; Tuzel, O.; Ramalingam, S.; Chellappa, R. Entropy rate superpixel segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Colorado Springs, CO, USA, 20–25 June 2011.
63. Doulamis, A.D.; Doulamis, N.D.; Kollias, S.D. A fuzzy video content representation for video summarization and content-based retrieval. *Signal Process.* **2000**, *80*, 1049–1067. [CrossRef]
64. Itti, L. Automatic foveation for video compression using a neurobiological model of visual attention. *IEEE Trans. Image Process.* **2004**, *13*, 1304–1318. [CrossRef]
65. Doulamis, N.; Doulamis, A.; Kalogeras, D.; Kollias, S. Low bit-rate coding of image sequences using adaptive regions of interest. *IEEE Trans. Circuits Syst. Video Technol.* **1998**, *8*, 928–934. [CrossRef]
66. Liu, D.; Wang, L. Visual attention based hyperspectral imagery visualization. In Proceedings of the 2012 Symposium on Photonics and Optoelectronics, SOPO 2012, Shanghai, China, 21–23 May 2012.
67. Yan, H.; Zhang, Y.; Wei, W.; Zhang, L.; Li, Y. Salient object detection in hyperspectral imagery using spectral gradient contrast. In Proceedings of the International Geoscience and Remote Sensing Symposium (IGARSS), Beijing, China, 10–15 July 2016; pp. 1560–1563.
68. Zhang, H.; Li, J.; Huang, Y.; Zhang, L. A nonlocal weighted joint sparse representation classification method for hyperspectral imagery. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 2056–2065. [CrossRef]
69. Jia, M.; Gong, M.; Zhang, E.; Li, Y.; Jiao, L. Hyperspectral image classification based on nonlocal means with a novel class-relativity measurement. *IEEE Geosci. Remote Sens. Lett.* **2014**, *11*, 1300–1304.
70. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the International Conference on Machine Learning ICML, Lille, France, 6–11 July 2015.
71. Lin, M.; Chen, Q.; Yan, S. Network in network. In Proceedings of the International Conference on Learning Representations ICLR, Banff, AB, Canada, 14–16 April 2014.
72. Foody, G.M. Status of land cover classification accuracy assessment. *Remote Sens. Environ.* **2002**, *80*, 185–201. [CrossRef]
73. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.S.; Davis, A.; Dean, J.; Devin, M.; et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. 2015. Software. Available online: https://arxiv.org/867abs/1603.04467 (accessed on 26 February 2019).