remote sensing

MDPI

Article

3-D Convolution-Recurrent Networks for Spectral-Spatial Classification of Hyperspectral Images

Majid Seydgar [1], Amin Alizadeh Naeini [1,*], Mengmeng Zhang [2] and Wei Li [3] and Mehran Satari [1]

1 Department of Geomatics Engineering, Faculty of Civil Engineering and Transportation, University of Isfahan, Isfahan 8174673441, Iran; m.seydgar@trn.ui.ac.ir (M.S.); sattari@eng.ui.ac.ir (M.S.)
2 College of Information Science and Technology, Beijing University of Chemical Technology, Beijing 100029, China; 2016400138@mail.buct.edu.cn
3 School of Information and Electronics, Beijing Institute of Technology, Beijing 100081, China; liwei089@ieee.org
* Correspondence: a.alizadeh@eng.ui.ac.ir

Received: 31 January 2019; Accepted: 6 April 2019; Published: 11 April 2019

Abstract: Nowadays, 3-D convolutional neural networks (3-D CNN) have attracted lots of attention in the spectral-spatial classification of hyperspectral imageries (HSI). In this model, the feed-forward processing structure reduces the computational burden of 3-D structural processing. However, this model as a vector-based methodology cannot analyze the full content of the HSI information, and as a result, its features are not quite discriminative. On the other hand, convolutional long short-term memory (CLSTM) can recurrently analyze the 3-D structural data to extract more discriminative and abstract features. However, the computational burden of this model as a sequence-based methodology is extremely high. In the meanwhile, the robust spectral-spatial feature extraction with a reasonable computational burden is of great interest in HSI classification. For this purpose, a two-stage method based on the integration of CNN and CLSTM is proposed. In the first stage, 3-D CNN is applied to extract low-dimensional shallow spectral-spatial features from HSI, where information on the spatial features are less than that of the spectral information; consequently, in the second stage, the CLSTM, for the first time, is applied to recurrently analyze the spatial information while considering the spectral one. The experimental results obtained from three widely used HSI datasets indicate that the application of the recurrent analysis for spatial feature extractions makes the proposed model robust against different spatial sizes of the extracted patches. Moreover, applying the 3-D CNN prior to the CLSTM efficiently reduces the model's computational burden. The experimental results also indicated that the proposed model led to a 1% to 2% improvement compared to its counterpart models.

Keywords: convolutional neural network (CNN); recurrent neural network (RNN); hyperspectral image classification; convolutional long short-term memory (CLSTM)

1. Introduction

Hyperspectral imagery (HSI) has a vast range of applications, such as in mineral exploration [1], anomaly detection [2], and supervised classification [3–5]. Among these applications, the classification of different land covers has attracted lots of attention in the remote sensing community due to the existence of rich spectral and spatial information in HSI. However, classification methods and algorithms need to be improved to be able to handle the large number of spectral bands and spatial information in HSI.

Traditional methods of classification, such as maximum likelihood (ML) [6], support vector machines (SVMs) [3], and random forest (RF) [7], have been and are being widely applied for HSI classification tasks. However, the major drawback of these methods is the fact that they classify data in a spectral domain, that is, they ignore spatial information during the classification process. To address this drawback, the SVM together with composite kernels (SVM-CK) was proposed by Camps-Valls et al. and Fauvel et al. [8,9] in order to enhance traditional SVM by considering the spatial information. Although SVM-CK uses features like the mean or the standard deviation to consider the spatial information, these features cannot represent the full content and semantic features of this kind of information. In parallel to this concept, morphological profiles (MPs) [10], extended MPs (EMPs) [11], and Markov random fields (MRFs) [12] were proposed to extract the spatial information. In Reference [13], advanced versions of these methods were applied on a number of HSI datasets. Although these methods significantly improved the classification performance, their spatial features were manually extracted, which means that prior knowledge and experts' experiences are of essence [14–16].

Recently, deep learning (DL) methods, in which both spectral and spatial information are automatically extracted, have been employed for HSI classification tasks [17–20]. Chen et al. [21] proposed a spectral-spatial classification auto-encoder model in which first, an HSI data cube is preprocessed using principal component analysis (PCA). After that, target pixels along with their neighbors, called PCA-cubes, are extracted from the principal components. Finally, PCA-cubes are unfolded to a 1-D vector form and entered into several stack auto-encoder layers. This scheme, that is, using the local regions to train a 1-D structural method, is also used in Reference [22]. In another study, Li et al. [23] developed a novel version of the scheme based on a 1-D convolutional neural network (CNN) and the pixel-pair strategy, to consider both the spectral and spatial information of neighboring pixels.

A major limitation of these methods is that they do not jointly employ both spectral and spatial information for classification. Instead of using a 1-D structure, Zhao et al. [24] used a novel dimensionality reduction method and a 2-D CNN model to extract spectral and spatial features. These features were stacked together and used for HSI classification. The PCA dimensionality reduction along with EMP were applied in Reference [13] to enhance both the spectral and spatial feature extractions of the 2-D CNN model. Ma et al. [14] also applied a spectral dimensionality reduction method to an HSI data cube. Afterwards, the reduced data was fed to a deep 2-D CNN model architecture. The model included several 2-D convolution, deconvolution, pooling, and unpooling layers with the residual connections. Nevertheless, due the application of dimensional reduction processes, these methods may discard some spectral information.To preserve the spectral information and to reduce the computational burden, the stacking spectral patches strategy was proposed in Reference [15]; nonetheless, the shallow CNN model of this method cannot extract spatial information in an efficient manner. The 1-D CNN and 2-D CNN were proposed by Reference [25] to extract the spectral and spatial information for classification in a separate manner. He et al. [26] proposed the application of a 2-D CNN framework with a new manual feature extraction method. These methods obtain good results, despite not using the full capacity of DL methods like joint or automatic spectral-spatial feature extraction.

To address the abovementioned issue, the 3-D structural methods are presented on extracted cubes/patches of the HSI where both spectral and spatial information are jointly extracted. In this context, Chen et al. [17] applied a 3-D CNN model to jointly extract the spectral and spatial features from the data. Their model contains the 3-D convolution and pooling layers to extract the spectral-spatial features and to decrease their dimensionality, respectively. The 3-D CNN model is also equipped with the residual connections [27]. Moreover, Liu et al. [28] applied the transfer learning and virtual sampling strategies to improve the 3-D CNN performance. The feed-forward processing structure let these models reduce the computational burden. As a drawback, these models, as the vector-based methodologies, cannot analyze the full content of spectral-spatial information.

Due to limited spatial information in the HSI extracted patches, losing this information leads to a side effect in the classification performance. In general, setting the appropriate spatial size of patches is a controversial issue in these methods [16,27–29].

The recurrent neural network (RNN) as a sequence-based methodology was proposed in Reference [30]. The intuition behind the use of RNN is to utilize the high computational power of a recurrent analysis for HSI classification. The RNN was combined with CNN by Reference [31] to boost the model performance. These methods considered the spectral bands of each pixel as a sequence of inputs to recurrently analyze them in the RNN. However, they only considered the spectral information while discarding the spatial one. To address the spatial consideration in the recurrent processing of the spectral information, the Bi-convolutional long short-term memory (CLSTM) was proposed in Reference [32]. This model applied a bidirectional connection to enhance the process of the spectral feature extraction. The convolutional kernel was also incorporated into the model to consider the neighboring information as well. Nevertheless, the concern of the recurrent processing structure of this model was the spectral feature extraction. That means that the extraction of spatial information is out of the concern of this model. In addition, the training process of this model is extremely time-consuming.

In summary, a joint extraction of the spectral and spatial features in a robust manner with a reasonable computational burden is the main concern in HSI classification. For this purpose, we develop a classification framework with two feature extraction stages. In the first stage, the 3-D CNN is applied to process a high-dimensional HSI cube to effectively extract low-dimensional spectral-spatial features. In the second stage, an innovative structure is designed to consider different pixels of the 3-D-CNN-driven features as a sequence of inputs for the CLSTM. By doing so, deep semantic spectral-spatial features are generated. The main contributions and novelties of the proposed model are briefed as follows.

1. Unlike previous studies, considering different bands as a sequence for the recurrent analysis, here, for the first time, neighboring pixels are regarded as a sequence to the recurrent procedure.
2. We build a novel, 3-D HSI classification framework which can jointly extract spectral and spatial features from HSI data. The architecture of this framework enables our model to take full advantage of vector-based and sequence-based learning methodologies in HSI classification.
3. To deal with the computational burden of CLSTM which makes the training process time-consuming, we take advantage of the 3-D CNN prior to the CLSTM to reduce the huge volume of spectral dimensionality. This strategy efficiently reduces the number of parameters in the recurrent processing of the CLSTM.

The remainder of this paper is organized as follows. The architecture of the proposed classification framework is briefly described in Section 2. Section 3 provides the experimental results, analysis, and comparisons. Lastly, Section 4 draws the concluding remarks of this paper.

2. Methodology

As illustrated in Figure 1, our DL classification model has three blocks. The first block is applied to extract the shallow low-dimensional spectral-spatial features. The second block is applied to extract the more abstract and semantic spectral-spatial features in a continuous form, and the third one runs the classification task.

For this purpose, let us assume $X = \{x_1, x_2, ..., x_n\} \in R^{1\times1\times B}$, where B is the number of spectral bands and n is the number of pixels having a ground truth label, namely $Y = \{y_1, y_2, ..., y_n\}$. For each member of X set, the image patches with the size of $w \times w \times B$ (w is the window size) are extracted, where x_i is its centered pixel. Accordingly, the X set can be represented as $X^* = \{x_1^*, x_2^*, ..., x_n^*\} \in R^{w\times w\times B}$. After the extraction of the image patches, each patch, x_i^* is fed into Block 1. In this block, the great spectral dimension of the input patch is processed in two convolutional layers, namely CNN1 and CNN2, containing hundreds of 3-D convolutional kernels. The details of these kernels are summarized

in Section 2.1. The function of these layers is to reduce the spectral dimension and to extract shallow spectral-spatial features. After completing the process of Block 1, its output $Z_1 \in R^{w \times w \times 1 \times K_1}$ (K_1 is the number of filters) is fed into the next block.

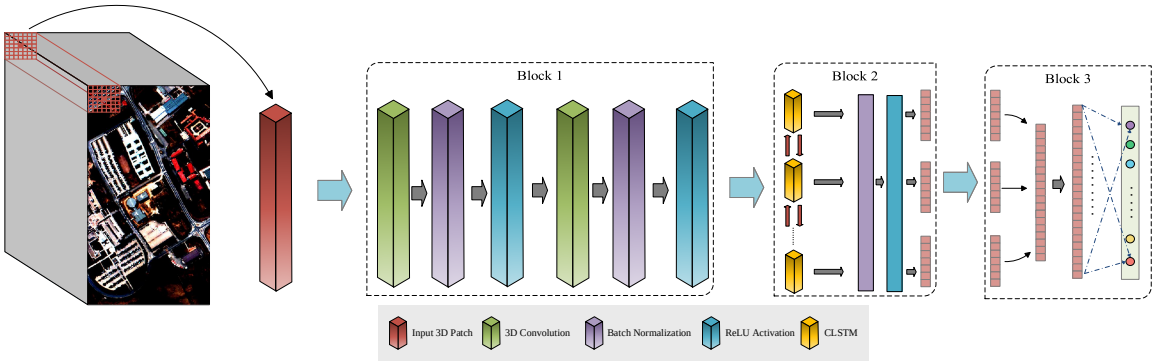


Figure 1. The framework of the proposed model for Hyperspectral imagery (HSI) classification.

The outputs of Block 1, are entered into Block 2 with a CLSTM layer in order to make the features more abstract and discriminative. In this layer, the neighboring pixels are regarded as a sequence of inputs (3-D tensors) for the recurrent procedure. To comprehend the 3-D structural processing of this layer, its inputs and states can be presumed as vectors standing on a spatial grid; consequently, the inputs and past states of neighboring pixels can determine the future state in this grid through a recurrent analysis. By doing this, the output of this layer can be represented as $Z_2 \in R^{w \times 1 \times 1 \times K_2}$, where K_2 is the number of outputs for Block 2. More detail on this layer is described in Section 2.2.

Finally, in Block 3, spectral-spatial features, consecutively extracted through pervious blocks, are considered for the membership probability of the classification task. This task is performed in a fully connected layer by a softmax as an activation function.

2.1. Convolutional Layer

According to Reference [33], a convolutional kernel in a 3-D CNN is applied to simultaneously extract the spectral and spatial features of the input. For the jth feature map in the ith layer, O_{ij} at an (x, y, z) position can be formulated as

$$O_{ij}^{xyz} = \varphi\left(b_{ij} + \sum_{k} \sum_{e=0}^{E_i-1} \sum_{f=0}^{F_i-1} \sum_{g=0}^{G_i-1} W_{ijk}^{efg} O_{(i-1)k}^{(x+e)(y+f)(z+g)}\right) \quad (1)$$

where e, f, and g are the width, height, and depth of the 3-D kernel, respectively; W is the weight of position (e, f, and g) connected to the k feature map, b denotes bias, and φ stands as an activation function. For the convolutional layers of the proposed model, the biases were omitted and batch normalization (BN) [34] along with the L_2 norm regularization technique were used. The BN is applied (i) to tackle the internal covariant shift phenomenon and (ii) to make model training possible in the presence of small initial learning rates. Furthermore, a rectified linear unit (ReLU) [35] is employed as an activation function. After applying these operations, the formulation of the convolutional layer can be rewritten as follows:

$$\hat{O}_{ij}^{xyz} = BN\left(max\left\{0, \left(\sum_k \sum_{e=0}^{E_i-1} \sum_{f=0}^{F_i-1} \sum_{g=0}^{G_i-1} \tilde{W}_{ijk}^{efg} O_{(i-1)k}^{(x+e)(y+f)(z+g)}\right)\right\}\right) \quad (2)$$

where

$$BN = \gamma\left(\frac{\tilde{O}_{ij}^{xyz} - E(\tilde{O}_{ij}^{xyz})}{Var(\tilde{O}_{ij}^{xyz}) - \varepsilon}\right) + \beta \quad (3)$$

Note that $\tilde{W}$ is a regularized weight matrix and that $\tilde{O}$ is the output of ReLU activation function. $\hat{O}$ presents the output of the BN function. Furthermore, $E(.)$ and $Var(.)$ respectively display the expectation and variance of each input, and γ and β stand as learnable parameters.

In the proposed model, we applied two 3-D CNN layers, CNN1 and CNN2, to process high-dimensional spectral information and the produce shallow spectral-spatial features. After completing the feature extraction process of these two layers, K_1 features the spectral dimension of which is 1 are generated. Subsequently, these features are entered to Block 2 where the spatial information is recurrently analyzed in this block.

2.2. Recurrent Layer

In the literature of sequence-based methodology, a long short-term memory [36] (LSTM) network has been used in a wide range of researches [36–40]. The main contribution of the LSTM architecture is to make the network more resistant to either the vanishing or exploding gradient phenomenon, which are the main concerns of an RNN's training process [41–43]. To do so, an LSTM defines a memory cell, C_t (Equation (4)), which contains the state information at time t. An input gate i_t (Equation (5)) and a forget gate f_t (Equation (6)), respectively, stores and discards a part of the memory cell information. The memory cell, C_t, is updated by a current memory cell information, named $\overline{C}$ (Equation (7)). Lastly, the cell, C_t, is tuned by an output gate, O_t (Equation (8)). To sum up, these features trap a gradient into a cell to preserve it from vanishing or exploding quickly [36]. However, as observed in Figure 2, an LSTM network has one 1-D structure because it applies matrix multiplication ("." in Figure 2). Therefore, the LSTM converts the inputs to 1-D vectors prior to any processing. This issue causes the spatial information of the data to be lost.

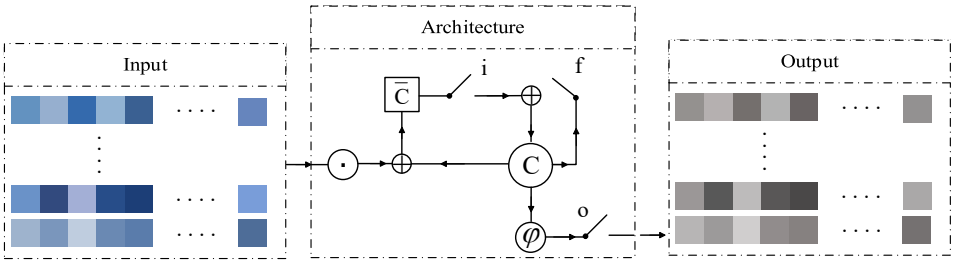


Figure 2. The input, architecture, and output of the long short-term memory (LSTM).

To address this problem, recently, Shi et al. [44] proposed a CLSTM network using convolutional kernels to perform an LSTM analysis for 3-D structural data. As observed in Figure 3, by replacing the convolutional operation ("*") with matrix multiplication, the spatial information is considered in the process of the sequence prediction. Accordingly, the CLSTM is taken into account for Block 2 to analyze the spatial contents of HSI in a recurrent manner while considering the spectral contents. In general, the convolutional operator of the CLSTM is set to consider the spatial information in the recurrent analysis of a temporal observation. However, in this study, the recurrent analysis of CLSTM is directly applied to the spatial dimension. Concurrently, the convolutional operator is incorporated into the model to consider a temporal observation in a recurrent processing of spatial information. According to this strategy, columns of HSI input patches are considered as the sequences of inputs. In the meanwhile, the kernel size of this layer determines how many sequences should be involved in the output generation. Setting the first dimension of the kernel size equal to 1, each sequence generates a one-step prediction. Consequently, by entering the $w \times w \times 1$ neighboring pixels' information to this layer, a $w \times 1 \times 1$ output feature is produced. Therefore, the sequence-based methodology can be applied to capture spatial information for the HSI classification. It should be mentioned that the biases of the original form of the CLSTM network are omitted and that the BN is applied to the output of this layer (Equation (9)). The equations of this layer are arranged as follows:

$$C_t = f_t \circ C_{t-1} + i_t \circ \overline{C}_t \tag{4}$$

$$i_t = \varphi'(W_{xi} * x_t + W_{hi} * H_{t-1} + W_{ci} * C_{t-1}) \tag{5}$$

$$f_t = \varphi'(W_{xf} * x_t + W_{hf} * H_{t-1} + W_{cf} * C_{t-1}) \tag{6}$$

$$\overline{C}_t = \varphi(W_{xc} * x_t + W_{hc} * H_{t-1}) \tag{7}$$

$$O_t = \varphi'(W_{xo} * x_t + W_{ho} * H_{t-1} + W_{co} \circ C_{t-1}) \tag{8}$$

$$H_t = BN(\varphi(C_t) \circ O_t) \tag{9}$$

where φ and φ' are the activation functions, H is the hidden state, $\circ$ is the point-wise product, $*$ is convolutional operator, and W is the weight matrix; e.g., W_{xc} is the input-memory cell weight matrix.

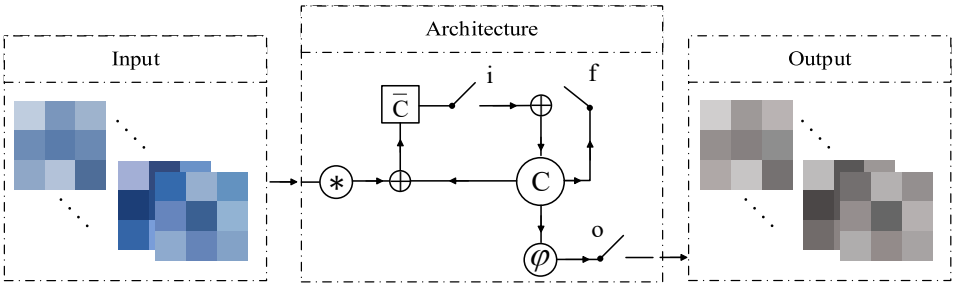


Figure 3. The input, architecture, and output of the convolutional long short-term memory (CLSTM).

We establish this architecture to recurrently analyze the spatial content of input cubes according to the assumption that neighboring pixels might belong to the same material/category. Note that it is quite possible that some cubes are extracted from the boundary of two or more categories. Accordingly, there are two different scenarios in an HSI classification using the CLSTM, both of which are discussed in the following.

In the first scenario, which has a homogeneous area assumption, pixels in a data cube belong to the same material. Since each pixel is entered into the model along with its neighboring pixels, extracted data cubes/patches are expected to have overlaps. These overlaps (i.e., common pixels among patches) have been considered by CLSTM. Since the proposed model has the capability to consider correlations among joint pixels in different patches, we call it patch-related convolutional long short-term memory (PRCLSTM).

In the second scenario, which is based on a heterogeneous area assumption, the patches processed in the model contain different materials. In this case, the inner structure of the CLSTM, which is controlled by gate units, can switch off irrelevant input connections and expose the relevant part of the memory information in the state-to-state transition. From the perspective of these two scenarios, it can be concluded that CLSTM guarantees an excellent performance in the both homogeneous and heterogeneous areas commonly available in HSI datasets.

3. Experimental Results

3.1. Experiment Data

To comprehensively assess the performance of the proposed model, three different types of publicly available hyperspectral datasets, Indian Pines, University of Pavia, and Salinas, were chosen and are described next:

1. *Indian Pines*: The Indian Pines dataset (Indiana) was collected by Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) over Northwestern Indiana in June 1992. This data includes 145 × 145 pixels in 220 spectral bands; however, 20 water absorption and low signal-to-noise bands were removed in our experiments. The spatial resolution of this data is 20 m, and the spectral bands cover wavelengths in the range of 0.4–2.5 µm. Figure 4a illustrates a false color composite and a ground truth map of the data. The ground truth map contains 16 land-cover classes, including the different types of vegetation species. For the training process, 30% of the labeled data was randomly chosen, and the rest was left just for the testing process. Table 1 indicates the number of training and test samples for each class.
2. *University of Pavia*: The University of Pavia (PaviaU) dataset was captured by Reflective Optics System Imaging Spectrometer (ROSIS-3) over this Italian university's engineering school in 2001. This dataset originally contained 113 spectral bands, with the size of 610 × 340 and a spatial resolution of 1.3 m. However, after removing noisy and damaged spectral bands, 103 bands in the range of 0.43–0.86 µm remained. Figure 4b shows a false color composite of the data and its ground truth map. The ground truth map contains nine different types of urban classes. As tabulated in Table 2, 20% of the labeled samples were randomly selected for the training process, while the remaining samples were only used for the testing process.
3. *Salinas*: The Salinas dataset consists of 512 × 217 pixels, captured by the AVIRIS sensor over the Salinas Valley in California, USA, in 1992. After removing noisy spectral bands, 204 bands, each with the spatial resolution of 3.7 m, were used in our experiments. As shown in Figure 4c, sixteen classes related to different agricultural crops were collected in the ground truth map. According to Table 3, about 18% of the labeled samples were used for the training, and the rest were considered completely for the testing.

Table 1. The number of training and testing samples for the Indiana dataset.

No.	Class	Train	Test
1	Alfalfa	14	32
2	Corn-notill	418	1010
3	Corn-min	250	580
4	Corn	67	170
5	Grass/Pasture	142	341
6	Grass/Trees	214	516
7	Grass/Pasture-mowed	7	21
8	Hay-windrowed	151	327
9	Oats	6	14
10	Soybeans-notill	305	667
11	Soybeans-min	733	1722
12	Soybeans-clean	174	419
13	Wheat	61	144
14	Woods	377	888
15	Bldg.-Grass-Tree-Drives	127	259
16	Stone-steel towers	27	66
	Total	3073	7176

Table 2. The number of training and testing samples for the PaviaU dataset.

No.	Class	Train	Test
1	Asphalt	1337	5294
2	Meadows	3742	14907
3	Gravel	404	1695
4	Trees	619	2445
5	Painted metal sheets	264	1081
6	Bare Soil	1028	4001
7	Bitumen	256	1074
8	Self-Blocking Bricks	718	2964
9	Shadows	194	753
	Total	8562	34214

Table 3. The number of training and testing samples for the Salinas dataset.

No.	Class	Train	Test
1	Brocoli-green-weeds-1	348	1661
2	Brocoli-green-weeds-2	665	3061
3	Fallow	346	1630
4	Fallow-rough-plow	242	1152
5	Fallow-smooth	471	2207
6	Stubble	711	3248
7	Celery	636	2943
8	Grapes-untrained	2038	9233
9	Soil-vinyard-develop	1129	5074
10	Corn-senesced-green-weeds	586	2692
11	Lettuce-romaine-4wk	183	885
12	Lettuce-romaine-5wk	333	1594
13	Lettuce-romaine-6wk	178	738
14	Lettuce-romaine-7wk	196	874
15	Vinyard-untrained	1277	5991
16	Vinyard-vertical-trellis	305	1502
	Total	9644	44485

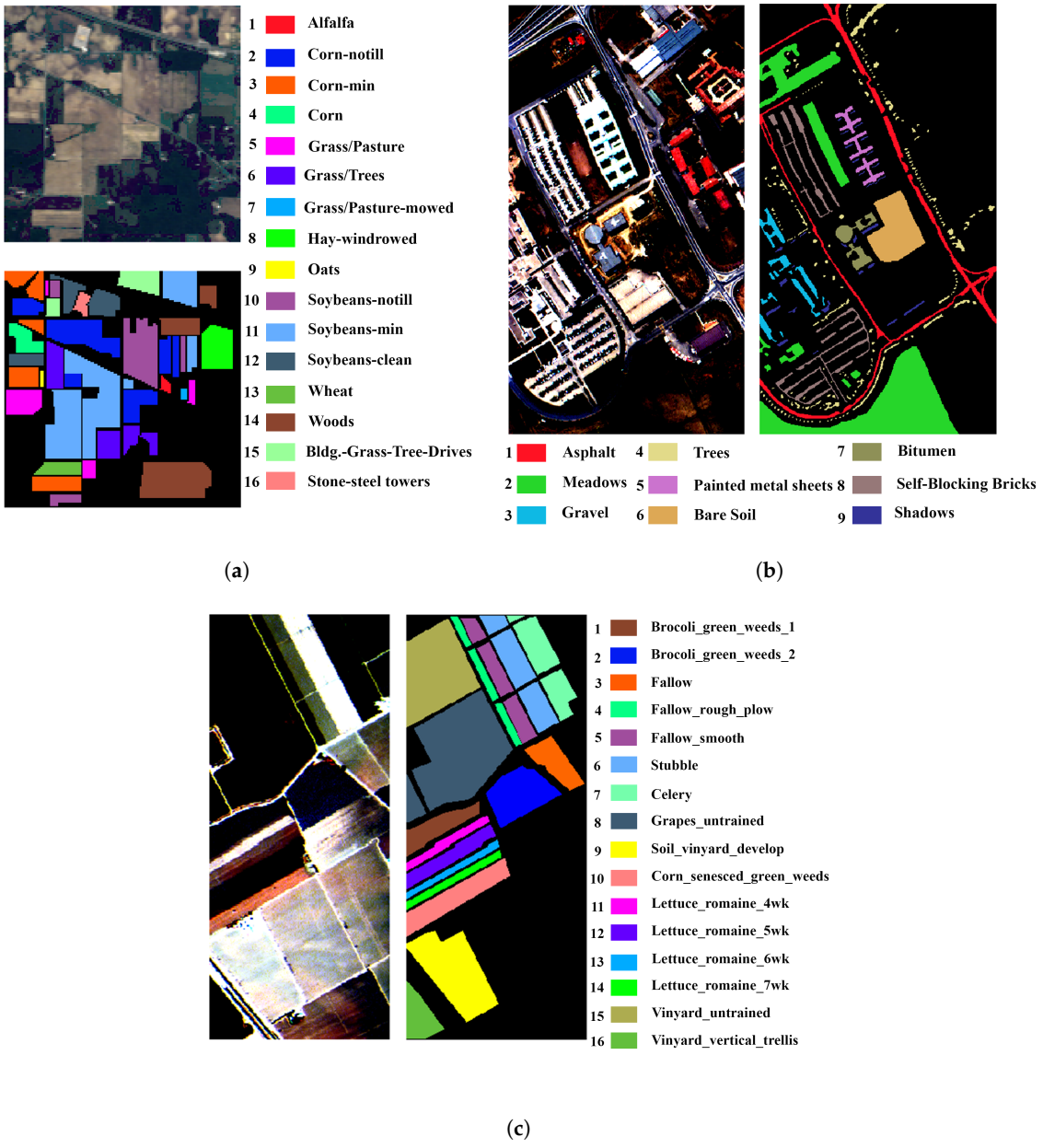


Figure 4. The false color composite and the ground truth map of the experiments data: (**a**) Indiana, (**b**) PaviaU, and (**c**) Salinas.

3.2. Experimental Settings

To comprehensively qualify the performance of the PRCLSTM model, four classification metrics, namely Overall Accuracy (OA), Kappa Coefficient (κ), Average Accuracy (AA), and Test Error (TE), are used in this study. The first three classification metrics are extracted from the confusion matrix, whereas the last one is the expected value of error on a new input of the network. This last criterion indicates the generalization capacity of the model, and its lower value demonstrates a better performance. To assess the independency of the proposed model on training and test data distributions, all experiments are repeated ten times with a random train and test splitting. Finally, the average and the standard deviation of the experiments are estimated and reported.

Before discussing the parameter setting for the proposed model, we show the model's framework and configuration for the Salinas dataset in detail (Figure 5 and Table 4). Inspired by Reference [27], we set the parameters of Block 1. To do so, the number of filters of CNN1 and CNN2 are respectively

considered as 24 and 128 in this block. At the same time, the kernel sizes of CNN1 and CNN2 are also respectively set as $1 \times 1 \times 7$ and $1 \times 1 \times K$, where K is the third dimension of the output of the previous layer. In Block 2, to completely use the spatial information of an input cube, the kernel size of the CLSTM layer is set to 1×1. To set the number of outputs of Block 2, the different number of kernels in the range of 10 to 22 with a step size of four are assessed. As illustrated in Figure 6, the model with 18 kernels has the higher mean of OA in the datasets in question. Thus, we choose this number for the output of Block 2 in our proposed model.

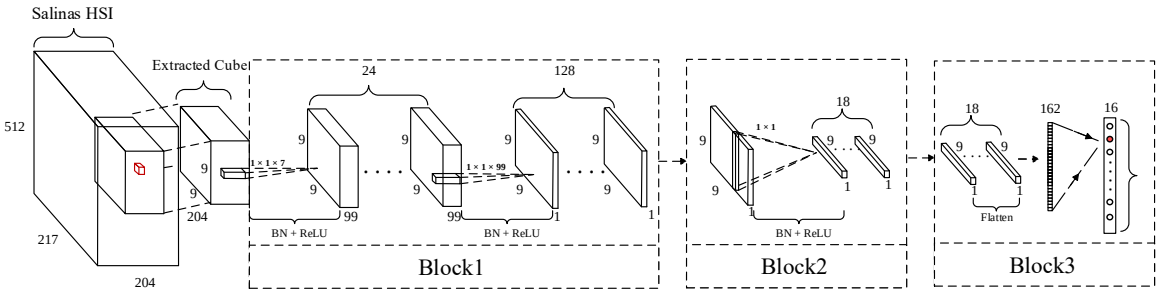


Figure 5. The patch-related convolutional long short-term memory (PRCLSTM) framework for the classification of Salinas dataset with a $9 \times 9 \times 204$ window size for the input cube.

Table 4. The configuration of the PRCLSTM for the classification of the Salinas dataset with a $9 \times 9 \times 204$ window size for the input cube.

Section	Unit	Input Shape	Kernel Size	Regularization	Output Shape
Block1	CNN1 + BN + ReLU	$9 \times 9 \times 204$	$1 \times 1 \times 7$	$L_2(0.0001)$	$9 \times 9 \times 99 \times 24$
	CNN2 + BN + ReLU	$9 \times 9 \times 99 \times 24$	$1 \times 1 \times 99$	$L_2(0.0001)$	$9 \times 9 \times 1 \times 128$
Block2	CLSTM + BN + ReLU	$9 \times 9 \times 1 \times 128$	1×1	Dropout (30%)	$9 \times 1 \times 18$
Block3	Flatten	$9 \times 1 \times 18$	-	-	162
	Fully Connected	162	-	Dropout (50%)	16

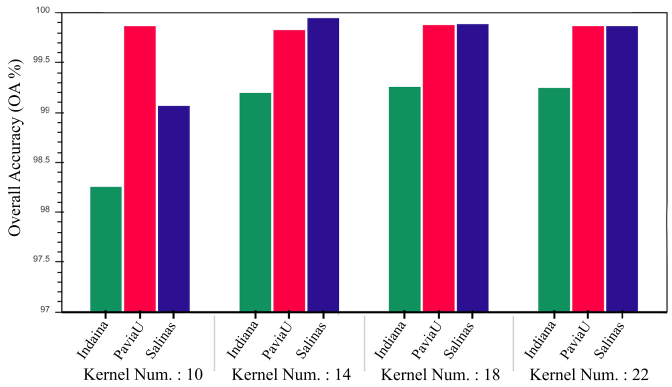


Figure 6. The overall accuracy (OA) of the PRCLSTM with different numbers of kernels for Block 2.

In the second part, the optimal parameters for the training process, specifically the batch size and learning rate, are obtained via a grid search algorithm. Note that the search space of these two parameters is set to References [16,27,29,45,46]. As a result of the grid search, the learning rate was set to 0.0001, 0.0003, and 0.0001 for Indiana, PaviaU, and Salinas, respectively. We also consider 0.00001 as the learning decay only for PaviaU because this parameter does not show a positive effect on the other datasets. The batch size is also set to 16 for all the experiment datasets.

In the third part, we set the parameters of the regularization techniques. In the PRCLSTM model, two types of regularization techniques—Batch Normalization (BN) [34] and dropout [47]—are taken

into account. As already mentioned, BN is used to tackle the internal covariant shift phenomenon [34], and the dropout is applied to increase the generalization capacity. Accordingly, BN is applied to the fourth dimension of each layer output to make the training process more efficiently. In addition, according to the number of the inputs, 30% of the connections and nodes are discarded by a dropout method for the CLSTM layer. Through the dropout method, 50% of the nodes are ignored in order to prevent overfitting in the classification layer. Finally, in the last part, we initialize the kernels of the CNN and CLSTM layers by using the truncated normal distribution [48]. Then, these initial values are optimized using a root mean squared error propagation (RMSProp) [49] as an optimizer to back propagate the gradient of the categorical cross-entropy cost function. In the training phase, after each epoch, the trained model is evaluated by the validation set, and the model with the lowest TE is preserved. Subsequently, the obtained model is applied for the testing procedure which provides the evaluation results.

3.3. Competing Methods

In order to evaluate the performance of the PRCLSTM, the model is compared to different types of classifiers like SVM, 1-D, 2-D, and 3-D CNN and RNN. Moreover, to analyze the effect of spectral-spatial feature extraction blocks, two additional models are also implemented. The configurations of the competing methods are as follows:

1. P-SVM-RBF: an SVM classifier with an radial basis function (RBF) kernel used to estimate class membership probabilities by an expensive K-fold cross-validation.
2. P-CNN [50]: a 2-D CNN layers with a PCA as a preprocessing procedure.
3. CRNN [31]: 1-D CNN layers, followed by two fully connected RNN layers.
4. 3-D-CNN [51]: 3-D CNN layers, followed by fully connected layers.
5. 3-D-CNN-LSTM: Our proposed model in which a CLSTM layer is replaced with a fully connected LSTM. This model was performed to demonstrate the efficacy of Block 2.
6. L-CLSTM: Our proposed model in which all layers are replaced with the CLSTM (L-CLSTM is the long CLSTM). This model was implemented to indicate the Block 1 efficiency.

To evaluate the performance of PRCLSTM, as a 3-D-LSTM-based model, we first compare that to two implemented 3-D-LSTM-based models, namely 3-D-CNN-LSTM and L-CLSTM. These models are then comprehensively compared with the other competitive classifier methods (cases 1–4 above). The best results based on the four classification metrics have been bolded in the different tables.

3.4. Analyses of the 3-D-LSTM-Based Models

In the first phase, the effect of Block 2 is analyzed through a comparison of 3-D-CNN-LSTM and PRCLSTM. Figure 7 illustrates the performance of the two models in terms of the validation loss (validation error) during the training process for the Indiana, PaviaU, and Salinas datasets. As is clear from Figure 7, the performance of the PRCLSTM is more stable than that of the 3-D-CNN-LSTM model. Moreover, our model, having a CLSTM layer, converges to a far better solution than that of the 3-D-CNN-LSTM. According to the test results (Tables 5–7), compared with the 3-D-CNN-LSTM, the PRCLSTM leads to 0.98%, 1.14%, 2.48%, and 0.05 improvements in the OA, κ, AA, and TE respectively, averaged on all three datasets. For example, the improvements for Indiana in terms of OA, κ, AA, and TE have respectively been 1.73%, 1.98%, 6.16%, and 0.08. In addition, as is obvious from the classification maps in Figures 8–10, the 3-D-CNN-LSTM, unlike our proposed model, leads to a great deal of noise and a malicious effect, both from the flatting out of the 3-D data by the fully connected LSTM layer for the PaviaU dataset and, especially, for the Indiana and Salinas datasets. As our experimental results indicate, considering the spatial information of data during sequence-based prediction leads to a more-effective learning process.

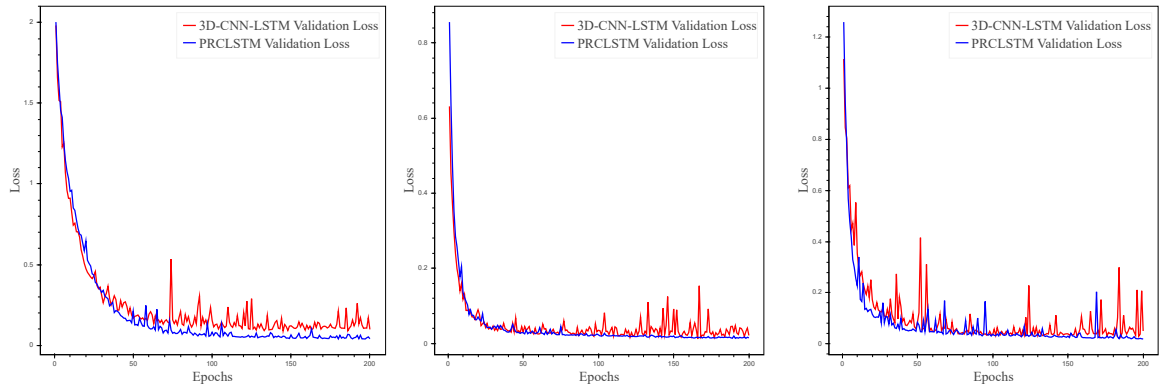


Figure 7. The validation loss during the training of the 3-D convolutional neural network (CNN)-LSTM and PRCLSTM for (**a**) Indiana, (**b**) PaviaU, and (**c**) Salinas.

Table 5. A comparison between the classification results of the Indiana dataset with a fixed spatial size of the input cube and a sufficient number of training samples.

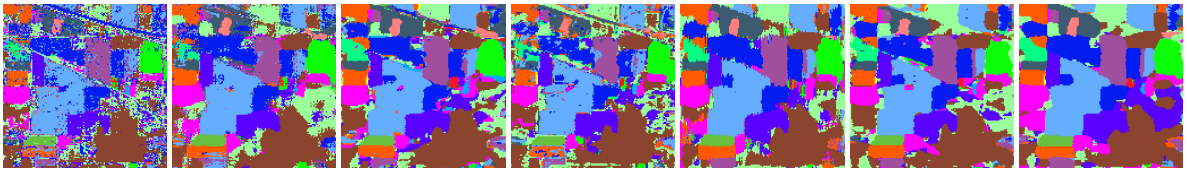
No.	P-SVM-RBF	P-CNN	CRNN	3-D-CNN	3-D-CNN-LSTM	L-CLSTM	PRCLSTM
1	85.10 ± 13.96	100 ± 0.0	96.86 ± 2.05	59.37 ± 13.34	100 ± 0.00	100 ± 0.0	100 ± 0.0
2	71.55 ± 2.26	79.49 ± 1.83	97.24 ± 0.59	87.98 ± 1.04	94.29 ± 3.49	93.90 ± 3.39	99.66 ± 0.27
3	83.20 ± 1.83	86.13 ± 2.42	96.97 ± 0.82	86.38 ± 1.71	97.92 ± 0.78	95.84 ± 2.26	99.19 ± 0.48
4	69.13 ± 4.90	95.57 ± 2.15	95.12 ± 2.44	64 ± 3.47	96.8 ± 2.09	97.42 ± 1.78	99.8 ± 0.30
5	93.41 ± 1.05	94.7 ± 1.37	97.42 ± 0.66	95.28 ± 0.70	98.31 ± 1.19	96.90 ± 1.31	98.07 ± 0.51
6	91.30 ± 1.28	98.49 ± 0.54	99.25 ± 0.42	99.83 ± 0.11	99.05 ± 0.61	98.44 ± 0.45	99.94 ± 0.10
7	94.99 ± 4.78	99 ± 3.16	93.05 ± 7.14	77.62 ± 7.79	100 ± 0.0	97.04 ± 4.8	100 ± 0.0
8	90.41 ± 0.97	94.14 ± 1.54	98.44 ± 1.19	98.84 ± 0.37	96.75 ± 0.94	96.30 ± 1.07	96.83 ± 0.22
9	83.65 ± 15.18	90 ± 31.62	97.78 ± 6.67	77.14 ± 11.57	20 ± 42.16	100 ± 0.0	100 ± 0.0
10	80.55 ± 2.62	85.96 ± 1.7	95.43 ± 0.81	87.23 ± 1.60	99.41 ± 0.57	93.29 ± 1.79	99.8 ± 0.33
11	80.16 ± 1.31	85.43 ± 0.88	97.87 ± 0.75	94.17 ± 0.43	98.77 ± 0.46	98.70 ± 0.68	99.18 ± 0.33
12	74.21 ± 2.74	82.37 ± 4.5	95.39 ± 0.59	67.59 ± 1.24	91.06 ± 4.45	96.66 ± 2.31	98.21 ± 1.01
13	95.53 ± 2.67	99.93 ± 0.21	99.24 ± 0.87	99.31 ± 0.57	99.13 ± 0.98	98.63 ± 1.12	99.93 ± 0.22
14	92.16 ± 0.90	96.18 ± 0.51	98.8 ± 1.37	97.26 ± 0.69	98.89 ± 0.38	99.40 ± 0.34	99.4 ± 0.15
15	80.10 ± 2.79	91.43 ± 2.21	98.88 ± 0.78	92.32 ± 1.63	98.56 ± 1.14	99.29 ± 0.75	99.79 ± 0.39
16	99.54 ± 0.97	98.49 ± 1.21	96.38 ± 2.94	85.60 ± 6.07	98.14 ± 1.46	90.20 ± 7.02	95.77 ± 1.26
OA	82.51 ± 0.51	88.52 ± 0.66	97.51 ± 0.32	90.53 ± 0.30	97.46 ± 0.5	96.86 ± 0.50	**99.19 ± 0.11**
κ	79.92 ± 0.59	86.85 ± 0.76	97.16 ± 0.36	89.16 ± 0.35	97.1 ± 0.57	96.42 ± 0.57	**99.08 ± 0.13**
AA	85.31 ± 1.36	92.33 ± 1.96	97.13 ± 0.84	85.62 ± 1.37	92.94 ± 2.86	97.0 ± 0.43	**99.1 ± 0.08**
TE	-	0.3200 ± 0.019	0.0906 ± 0.011	0.5605 ± 0.024	0.1292 ± 0.023	0.1416 ± 0.051	**0.0492 ± 0.003**

Table 6. A comparison between the classification results of the PaviaU dataset with a fixed spatial size of the input cube and a sufficient number of training samples.

No.	P-SVM-RBF	P-CNN	CRNN	3-D-CNN	3-D-CNN-LSTM	L-CLSTM	PRCLSTM
1	94.92 ± 0.44	95.44 ± 0.49	97.37 ± 1.82	95.42 ± 0.24	99.65 ± 0.21	96.27 ± 3.83	99.74 ± 0.07
2	96.30 ± 0.21	98.85 ± 0.13	99.69 ± 0.10	99.03 ± 0.09	99.88 ± 0.05	99.53 ± 0.23	99.97 ± 0.02
3	86.95 ± 1.25	92.99 ± 1.88	95.99 ± 1.96	89.54 ± 0.83	98.35 ± 2.01	93.48 ± 3.93	99.99 ± 0.03
4	96.91 ± 0.36	99.50 ± 0.18	98.33 ± 2.04	99.41 ± 0.21	99.76 ± 0.12	99.86 ± 0.14	99.98 ± 0.05
5	99.98 ± 0.04	99.72 ± 0.27	99.47 ± 0.06	95.39 ± 1.06	99.84 ± 0.1	99.92 ± 0.1	99.88 ± 0.08
6	91.56 ± 0.89	98.67 ± 0.48	99.54 ± 0.27	96.53 ± 0.47	99.96 ± 0.04	99.86 ± 0.2	100 ± 0.0
7	89.95 ± 1.93	96.04 ± 0.82	97.25 ± 1.24	93.75 ± 1.01	96.23 ± 3.81	99.84 ± 0.23	99.73 ± 0.83
8	86.37 ± 0.91	90.04 ± 1.08	94.97 ± 1.65	90.61 ± 0.39	96.95 ± 4.24	98.72 ± 0.87	99.47 ± 0.47
9	100 ± 0.0	99 ± 0.59	99.22 ± 0.66	99.71 ± 0.30	99.02 ± 0.83	99.65 ± 0.39	99.09 ± 0.55
OA	94.26 ± 0.14	97.24 ± 0.12	98.51 ± 0.46	96.75 ± 0.06	99.35 ± 0.43	98.69 ± 0.65	**99.87 ± 0.07**
κ	92.36 ± 0.19	96.33 ± 0.17	98.02 ± 0.62	95.69 ± 0.09	99.14 ± 0.57	98.26 ± 0.86	**99.82 ± 0.09**
AA	93.66 ± 0.24	96.70 ± 0.23	97.98 ± 0.49	95.49 ± 0.19	98.85 ± 0.66	98.57 ± 0.6	**99.76 ± 0.15**
TE	-	0.0846 ± 0.004	0.0683 ± 0.021	0.3091 ± 0.008	0.044 ± 0.030	0.0811 ± 0.040	**0.0176 ± 0.003**

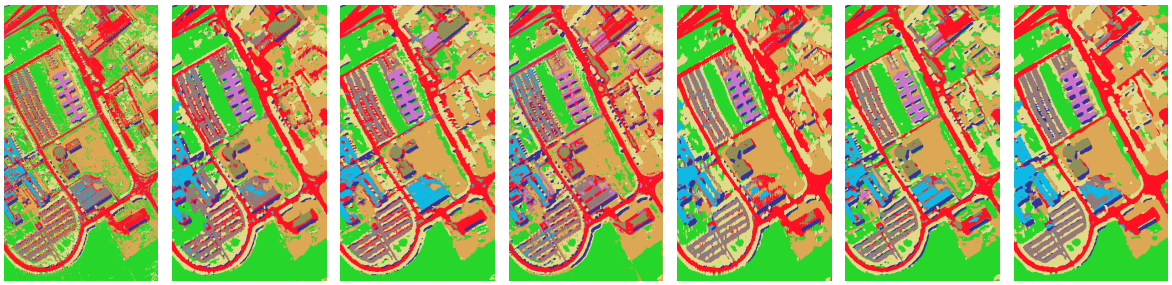
Table 7. A comparison between the classification results of the Salinas dataset with a fixed spatial size for the input cube and a sufficient number of training samples.

No.	P-SVM-RBF	P-CNN	CRNN	3-D-CNN	3-D-CNN-LSTM	L-CLSTM	PRCLSTM
1	99.99 ± 0.03	100 ± 0.0	100 ± 0.0	99.9 ± 0.15	99.98 ± 0.04	99.92 ± 0.14	100 ± 0.0
2	99.76 ± 0.11	99.94 ± 0.04	99.88 ± 0.15	99.78 ± 0.24	99.87 ± 0.08	89.60 ± 8.38	99.92 ± 0.16
3	97.91 ± 0.68	99.72 ± 0.24	98.66 ± 1.86	99.63 ± 0.12	99.92 ± 0.08	99.90 ± 0.22	100 ± 0.0
4	98.91 ± 0.24	99.61 ± 0.23	98.68 ± 1.59	99.94 ± 0.08	99.66 ± 0.26	98.80 ± 0.75	99.68 ± 0.19
5	99.61 ± 0.09	99.39 ± 0.30	99.84 ± 0.17	99.19 ± 0.15	99.93 ± 0.05	100 ± 0.0	99.95 ± 0.05
6	100 ± 0.0	100 ± 0.0	99.86 ± 0.10	99.93 ± 0.04	100 ± 0.0	99.89 ± 0.21	99.99 ± 0.03
7	99.99 ± 0.03	100 ± 0.0	99.60 ± 0.95	99.26 ± 0.39	99.95 ± 0.06	99.91 ± 0.11	100 ± 0.0
8	82.35 ± 0.52	95.91 ± 0.78	96.53 ± 0.61	94.77 ± 0.39	99.63 ± 0.21	96.40 ± 1.89	99.80 ± 0.27
9	99.54 ± 0.06	99.90 ± 0.15	99.97 ± 0.07	99.78 ± 0.12	99.98 ± 0.04	99.94 ± 0.07	100 ± 0.0
10	96.70 ± 0.49	99.59 ± 0.26	98.60 ± 0.69	97.52 ± 0.46	99.82 ± 0.11	99.67 ± 0.35	99.86 ± 0.05
11	98.24 ± 0.53	97.59 ± 1.67	98.52 ± 2.62	97.12 ± 0.54	97.73 ± 0.60	98.95 ± 0.91	98.92 ± 0.52
12	99.07 ± 0.36	99.73 ± 0.10	99.96 ± 0.06	99.99 ± 0.02	99.93 ± 0.06	99.95 ± 0.03	99.97 ± 0.04
13	98.78 ± 0.57	99.54 ± 0.36	99.81 ± 0.13	99.93 ± 0.12	99.53 ± 0.68	99.97 ± 0.06	99.92 ± 0.14
14	98.37 ± 0.56	99.88 ± 0.14	99.36 ± 0.28	99.53 ± 0.30	99.93 ± 0.11	99.48 ± 0.45	99.78 ± 0.35
15	80.49 ± 0.83	90.45 ± 1.44	94.99 ± 0.46	93.18 ± 0.90	95.62 ± 3.88	95.79 ± 4.51	99.82 ± 0.57
16	99.84 ± 0.19	99.98 ± 0.06	98.33 ± 0.32	99.24 ± 0.26	100 ± 0.0	99.62 ± 0.54	100 ± 0.0
OA	93.15 ± 0.11	97.67 ± 0.09	98.28 ± 0.34	97.6 ± 0.11	99.19 ± 0.58	97.65 ± 0.72	**99.88 ± 0.08**
κ	92.36 ± 0.12	97.41 ± 0.10	98.08 ± 0.38	97.33 ± 0.13	99.10 ± 0.64	97.38 ± 0.80	**99.87 ± 0.09**
AA	96.85 ± 0.08	98.83 ± 0.12	98.91 ± 0.48	98.67 ± 0.08	99.47 ± 0.25	98.61 ± 0.51	**99.85 ± 0.05**
TE	-	0.0628 ± 0.003	0.0545 ± 0.012	0.2405 ± 0.012	0.0505 ± 0.028	0.1358 ± 0.73	**0.0196 ± 0.002**



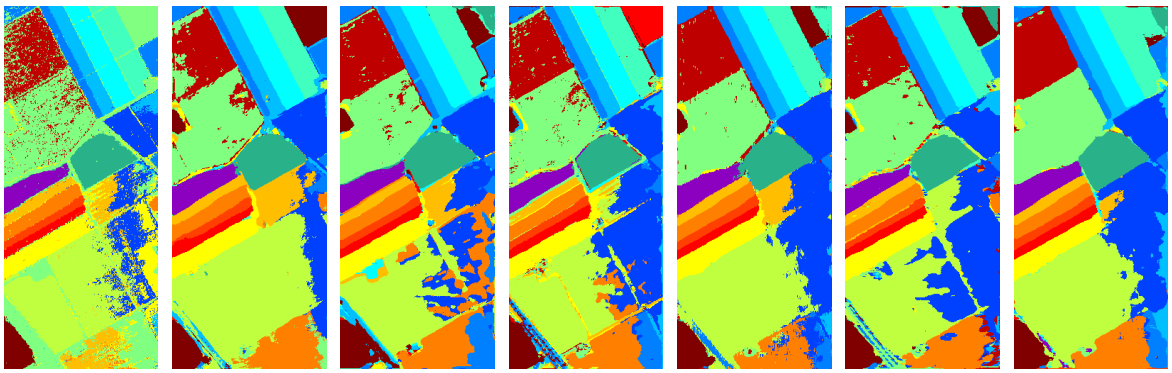
(**a**) (**b**) (**c**) (**d**) (**e**) (**f**) (**g**)

Figure 8. The classification maps of Indiana: (**a**) P-SVM-RBF, (**b**) P-CNN, (**c**) CRNN, (**d**) 3-D-CNN, (**e**) 3-D-CNN-LSTM, (**f**) L-CLSTM, and (**g**) PRCLSTM.



(**a**) (**b**) (**c**) (**d**) (**e**) (**f**) (**g**)

Figure 9. The classification maps of PaviaU: (**a**) P-SVM-RBF, (**b**) P-CNN, (**c**) CRNN, (**d**) 3-D-CNN, (**e**) 3-D-CNN-LSTM, (**f**) L-CLSTM, and (**g**) PRCLSTM.



(**a**) (**b**) (**c**) (**d**) (**e**) (**f**) (**g**)

Figure 10. The classification maps of Salinas: (**a**) P-SVM-RBF, (**b**) P-CNN, (**c**) CRNN, (**d**) 3-D-CNN, (**e**) 3-D-CNN-LSTM, (**f**) L-CLSTM, and (**g**) PRCLSTM.

The second phase analyses the effect of the spectral feature extraction and dimension reduction, performed by the 3-D-CNN layers in Block 1. For this purpose, the L-CLSTM model, which contains only CLSTM layers, is selected. To achieve a fair comparison, the parameters of the L-CLSTM model are considered to be approximately equal to the PRCLSTM model. As illustrated in Figure 11, due to the high dimension of the input data, the L-CLSTM model, unlike the PRCLSTM model, cannot converge to the proper solution in terms of the validation loss during the training process for any of the datasets. Tables 5–7 show that the PRCLSTM leads to on average 1.91%, 2.24%, 1.51%, and 0.09 improvements in the OA, κ, AA, and TE, respectively, for all of the three datasets compared to L-CLSTM. From the perspective of computational burden, as seen in Table 8, the computational cost of the L-CLSTM model is much higher than that of the PRCLSTM model. The results in this phase imply that the extraction of low-dimensional shallow features, as inputs of Block 2, play a key role in improving the performance of the recurrent analysis.

Table 8. The computational time (min: minutes, s: second) of the 3-D-LSTM-based models for the three datasets.

Method	Time	Indiana	PaviaU	Salinas
P-CNN	Train (m)	2.2	3.5	3.9
	Test (s)	1	2.5	3.1
CRNN	Train (m)	2.5	4.7	7.1
	Test (s)	1	2.5	7.7

Table 8. *Cont.*

Method	Time	Indiana	PaviaU	Salinas
3-D-CNN	Train (m)	8.2	12	22.3
	Test (s)	2.7	8	17.4
3-D-CNN-LSTM	Train (m)	20.7	37.3	54.6
	Test (s)	5	20	37
L-CLSTM	Train (m)	37.1	122.9	171.8
	Test (s)	37	69	108
PRCLSTM	Train (m)	20.8	38.6	55.7
	Test (s)	5	21	39

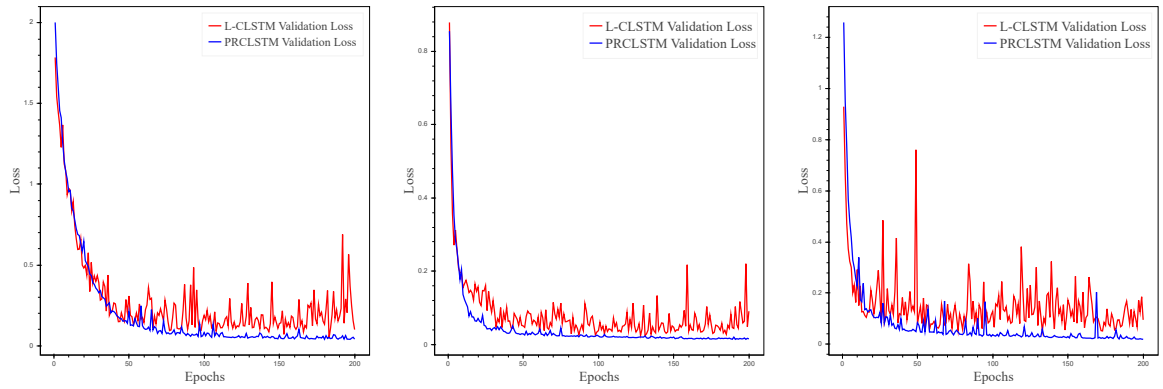


Figure 11. The validation loss during the training of the L-CLSTM and PRCLSTM for (**a**) Indiana, (**b**) PaviaU, and (**c**) Salinas.

3.5. The PRCLSTM Performance

The two main factors that influence the performance of DL methods in an HSI classification task are the spatial size of an input cube and the number of training samples. Accordingly, we establish a three-part analysis to evaluate the performance of the proposed model in terms of these parameters. In the first part, the performance of the models is assessed in the presence of a fixed and sufficient number of training samples, as well as a fixed spatial size for the input cube. To do so, the training and test samples are determined as reported in Tables 1–3. To prevent over-fitting and to enhance the training procedure, 35%, 50%, and 50% of the training samples are selected for the validation of the learned parameters of Indiana, PaviaU, and Salinas, respectively, during the training process. It is worth mentioning that the test samples are involved in neither the validation nor the training procedures and that they are just used for testing. The hyper-parameters of the 3-D-CNN-LSTM and L-CLSTM models are set the same as the PRCLSTM model, and for the other models, these parameters are set based on their original implementations. To have a fair comparison, a $9 \times 9 \times B$ (B is the number of spectral bands) window size is selected as the input cubes of all models, and 200 is set as the number of training epochs. The results of the evaluation criteria for all models in hand have been illustrated in Tables 5–7 for the Indiana, PaviaU, and Salinas datasets, respectively. These tables demonstrate that the PRCLSTM model achieves the best performance in all three datasets. For instance, in the Indiana dataset, the PRCLSTM achieves about a 2% higher accuracy in terms of the OA as well as a standard deviation three times lower than the best result gained by the other competitors. Moreover, according to Figures 8–10, the PRCLSTM produces the most accurate and noiseless classification maps.

In the second part, the training processes with different spatial sizes of input cubes are performed. We set the extracted window to a range of sizes from 3×3 to 11×11 and report the OA and TE for each

model. Table 9 and Figure 12 indicate that the proposed model achieves the best results in most cases. However, for the Salinas dataset, the 3-D-CNN-LSTM performs relatively better than the PRCLSTM in terms of the OA for the 3×3 window size, but the PRCLSTM still shows the lowest TE, guaranteeing a better generalization capacity for our model. These results indicate that the proposed model performs better in the various sizes of input cubes compared to the convolutional feed-forward-based models, thanks to the use of a convolutional recurrent structure to capture the spatial information of data. For example, the 3-D-CNN model, as a competing method, performs the best in the case of the 5×5 window size, whereas its performance deteriorates for the other window sizes.

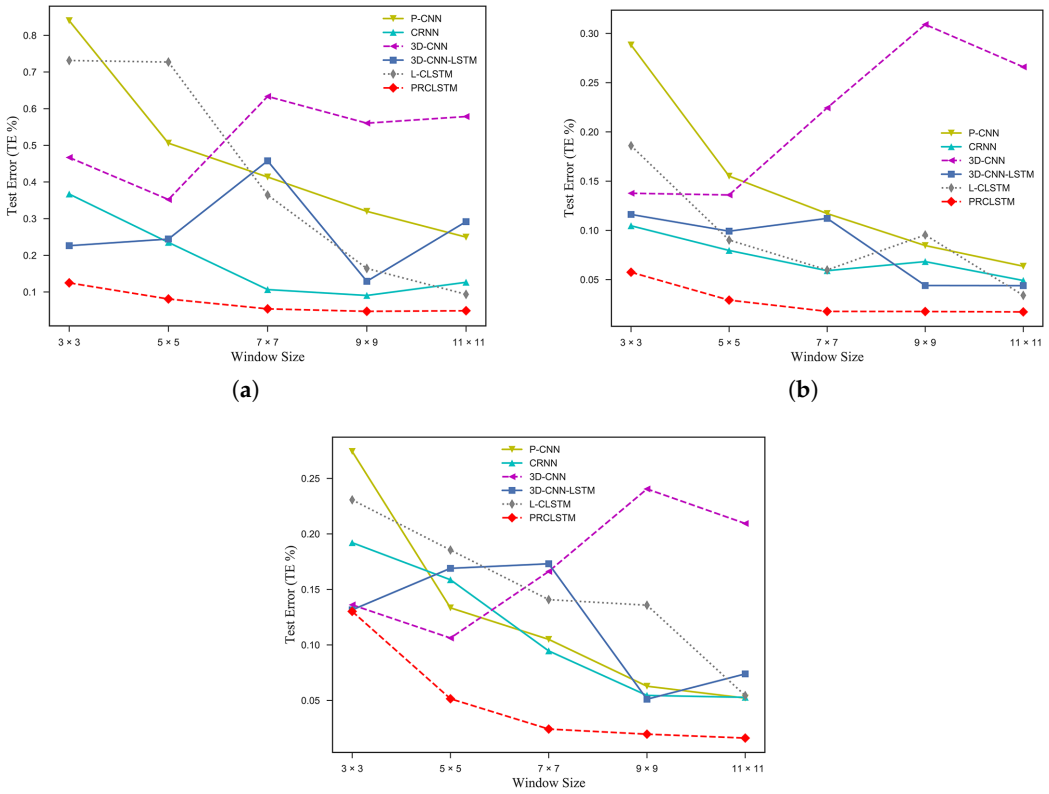


Figure 12. The test Errors (TEs) of the various methods with different window sizes of the input cube for (**a**) Indiana, (**b**) PaviaU, and (**c**) Salinas.

Table 9. The overall accuracy (OA%) of different window sizes of the input cube for the three datasets.

Method	Dataset	3 × 3	5 × 5	7 × 7	9 × 9	11 × 11
P-CNN	Indiana	70.81 ± 0.72	81.96 ± 0.41	85.06 ± 0.4	88.52 ± 0.66	91.52 ± 0.47
CRNN		89.57 ± 0.11	93.76 ± 1.02	96.82 ± 0.63	97.51 ± 0.32	96.96 ± 1.08
3-D-CNN		93.44 ± 0.33	95.58 ± 0.16	90.46 ± 0.34	90.53 ± 0.3	90.68 ± 0.22
3-D-CNN-LSTM		95.15 ± 0.65	94.62 ± 1.63	88.98 ± 1.28	97.46 ± 0.5	92.89 ± 1.19
L-CLSTM		75 ± 1.20	75.83 ± 4.28	88.16 ± 4.31	96.86 ± 0.50	97.79 ± 0.56
PRCLSTM		**96.94 ± 0.35**	**98.38 ± 0.22**	**99.02 ± 0.32**	**99.19 ± 0.11**	**99.32 ± 0.19**

Table 9. *Cont.*

Method	Dataset	3 × 3	5 × 5	7 × 7	9 × 9	11 × 11
P-CNN	PaviaU	89.49 ± 0.11	94.31 ± 0.11	95.89 ± 0.16	97.24 ± 0.12	97.86 ± 0.14
CRNN		97.18 ± 0.04	98.05 ± 0.16	98.63 ± 0.06	98.51 ± 0.46	99 ± 0.09
3-D-CNN		97.25 ± 0.08	97.79 ± 0.08	97.63 ± 0.08	96.75 ± 0.06	96.66 ± 0.17
3-D-CNN-LSTM		98.39 ± 0.25	98.77 ± 0.1	98.55 ± 0.35	99.35 ± 0.43	99.4 ± 0.22
L-CLSTM		92.93 ± 1.33	97.43 ± 0.63	98.51 ± 0.30	98.85 ± 0.73	99.35 ± 0.27
PRCLSTM		**99.05 ± 0.09**	**99.66 ± 0.09**	**99.92 ± 0.03**	**99.87 ± 0.07**	**99.86 ± 0.03**
P-CNN	Salinas	90.27 ± 0.2	94.96 ± 0.11	96.01 ± 0.12	97.67 ± 0.09	98.02 ± 0.09
CRNN		92.45 ± 0.34	94.29 ± 1.66	96.85 ± 0.59	98.28 ± 0.34	98.38 ± 0.91
3-D-CNN		96.53 ± 0.43	98.02 ± 0.19	97.22 ± 0.22	97.60 ± 0.11	97.11 ± 0.77
3-D-CNN-LSTM		**97.37 ± 0.13**	97.32 ± 0.28	97.34 ± 0.4	99.2 ± 0.59	98.89 ± 0.65
L-CLSTM		92.30 ± 0.9	94.69 ± 0.46	96.16 ± 0.41	97.65 ± 0.72	98.73 ± 1.01
PRCLSTM		96.75 ± 0.4	**99.13 ± 0.21**	**99.80 ± 0.08**	**99.88 ± 0.08**	**99.93 ± 0.03**

In the third part, we assess the capability of the proposed model when a small amount of training data is available. For this purpose, the Salinas dataset is selected and the models are trained by a range of 0.5% to 3% of data as training samples. Then the OA, κ, and AA are reported for the rest of the samples. From the experiments, first of all, one can conclude that the models which contain a convolutional recurrent unit perform better than the models with a convolutional feed-forward unit when limited training samples are accessible. For example, as is obvious from Figure 13, where the competing methods, such as the CRNN and 3-D-CNN, lead to about an 80% accuracy in terms of the OA, the convolutional recurrent models, that is, the PRCLSTM and L-CLSTM, result in a higher than 91% accuracy. It should be noted that, in the case of very limited training samples (i.e., 0.5% and 1%), the L-CLSTM model results in a relatively higher classification accuracy; however, this improvement is not reliable because the accuracy of the L-CLSTM fluctuates with an increase in the number of training samples. Generally speaking, the results of the PRCLSTM are more accurate and more stable in both scenarios of having sufficient and limited training samples.

In addition to the abovementioned analyses, the training and testing time of the 3-D-LSTM model for the constant spatial size of the input cubes (i.e., $9 \times 9 \times B$) and a sufficient amount of training samples are tabulated in Table 8. Thanks to the Graphical Processor Unit (GPU) resources, the computational cost of DL methods is reduced. In general, recurrent-based models are still computationally expensive; however, as observed in Table 8, two 3-D-LSTM-based models (3-D-CNN-LSTM and PRCLSTM), by taking advantage of the dimensionality reduction of the 3-D CNN are able to reduce the computational burden of CLSTM. Furthermore, to statistically compare the proposed method with its best rivals in each dataset, the wilcoxon rank sum test [52] was used here. This test, which is a common statistical test, can be applied to compare two independent sets of samples without having any assumption about the statistical distribution of the sets. In our case, a set is a 10×1 vector of κ values, coming from 10 runs of each classifier. The obtained result, tabulated in Table 10 indicates that the differences between PRCLSTM and the competing models are statistically significant at the confidence level of 95%.

It is worth mentioning that all the experiments here were conducted on an ASUS FX-553 laptop that has a NVIDIA GeForce GTX-1050Ti GPU (4GB GDDR5). Moreover, all the models have been implemented in the Python (https://www.python.org) language, which has utilized Keras (https://www.keras.io), TensorFlow (https://www.tensorflow.org), and libsvm (https://www.csie.ntu.edu.tw/~cjlin/libsvm/) as its machine learning libraries.

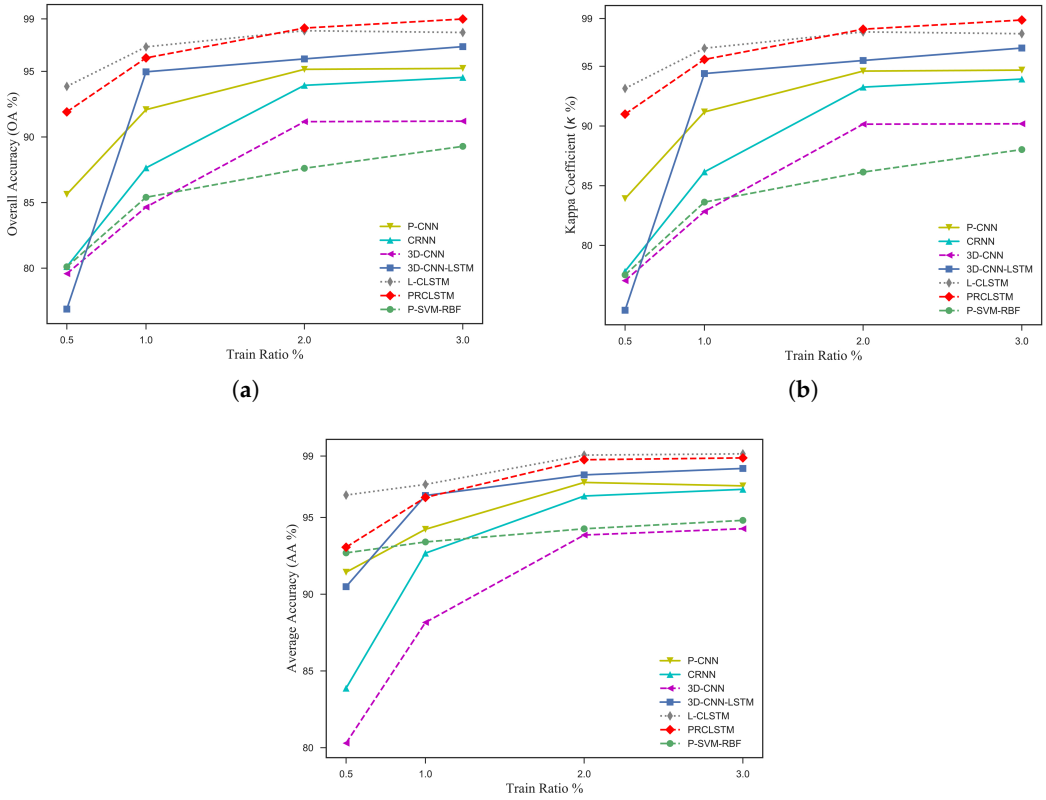


Figure 13. The classification result (%) with a limited size of training samples: (**a**) OA, (**b**) Kappa Coefficent (κ), and (**c**) Average Accuracy (AA).

Table 10. The statistical significance of the PRCLSTM and the best competitors on the datasets in question.

Method	Dataset	κ	p-Value	Statistical Significance
CRNN	Indiana	97.16 ± 0.36	0.00018	**Significant**
PRCLSTM		99.08 ± 0.13		
3-D-CNN-LSTM	PaviaU	99.14 ± 0.57	0.0022	**Significant**
PRCLSTM		99.82 ± 0.09		
3-D-CNN-LSTM	Salinas	99.10 ± 0.64	0.00018	**Significant**
PRCLSTM		99.87 ± 0.09		

4. Conclusions

In recent years, deep learning (DL) methods are of major concern in hyperspectral imagery (HSI) classification tasks. These methods are able to automatically extract spectral and spatial information and to apply them in classification processes. However, a full extraction of both the spectral and spatial information are the main concerns of these methods. To address them, here, an innovative convolution-recurrent framework has been presented to build a 3-D spectral-spatial trainable model for HSI classification. This model contains two stages: (i) the 3-D CNNs, which produce shallow spectral-spatial features in a feed-forward procedure, and (ii) a CLSTM, which extracts more abstract and semantic features in a recurrent manner. The focus of the first stage is to analyze a high spectral dimension, while the concentration of the second one is to process the full content of spatial information. In this study, the performance of the PRCLSTM (the proposed model) has been assessed in the presence

of (i) a sufficient number of training samples, (ii) a limited number of training samples, and (iii) different input cubes sizes. From the results, it can be concluded that the PRCLSTM performance is promising, even with a limited number of training samples. Moreover, the recurrent analysis for extracting spatial information makes the classification model robust against the different input cubes sizes, indicating a high generalization capacity of the proposed model.

Author Contributions: M.S. (Majid Seydgar), A.A.N., and M.S. (Mehran Satari) conceptualized and developed the methodology. M.S. (Majid Seydgar) performed the experiments. W.L. and M.Z. verified the experiments and provided advice during the work. M.S. (Majid Seydgar) and A.A.N. wrote the manuscript. All the authors revised the manuscript.

Funding: This research received no external funding.

Acknowledgments: The authors would like to thank N. Iranpanah from the University of Isfahan for providing advice for the statistical analyses performed in this study. The same appreciation goes to A.M. Chegoonian from the University of Waterloo, who reviewed the paper and provided us his valuable comments.

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

References

1. Kruse, F.A.; Boardman, J.W.; Huntington, J.F. Comparison of airborne hyperspectral data and EO-1 Hyperion for mineral mapping. *IEEE Trans. Geosci. Remote Sens.* **2003**, *41*, 1388–1400.
2. Li, W.; Wu, G.; Du, Q. Transferred Deep Learning for Anomaly Detection in Hyperspectral Imagery. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 597–601. [CrossRef]
3. Melgani, F.; Bruzzone, L. Classification of hyperspectral remote sensing images with support vector machines. *IEEE Trans. Geosci. Remote Sens.* **2004**, *42*, 1778–1790. [CrossRef]
4. Zheng, X.; Yuan, Y.; Lu, X. Dimensionality reduction by spatial–spectral preservation in selected bands. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 5185–5197. [CrossRef]
5. Gao, F.; Wang, Q.; Dong, J.; Xu, Q. Spectral and Spatial Classification of Hyperspectral Images Based on Random Multi-Graphs. *Remote Sens.* **2018**, *10*, 1271. [CrossRef]
6. Kuching, S. The performance of maximum likelihood, spectral angle mapper, neural network and decision tree classifiers in hyperspectral image analysis. *J. Comput. Sci.* **2007**, *3*, 419–423.
7. Ham, J.; Chen, Y.; Crawford, M.M.; Ghosh, J. Investigation of the random forest framework for classification of hyperspectral data. *IEEE Trans. Geosci. Remote Sens.* **2005**, *43*, 492–501. [CrossRef]
8. Camps-Valls, G.; Gomez-Chova, L.; Muñoz-Marí, J.; Vila-Francés, J.; Calpe-Maravilla, J. Composite kernels for hyperspectral image classification. *IEEE Geosci. Remote Sens. Lett.* **2006**, *3*, 93–97. [CrossRef]
9. Fauvel, M.; Chanussot, J.; Benediktsson, J.A. A spatial–spectral kernel-based approach for the classification of remote-sensing images. *Pattern Recognit.* **2012**, *45*, 381–392. [CrossRef]
10. Pesaresi, M.; Benediktsson, J.A. A new approach for the morphological segmentation of high-resolution satellite imagery. *IEEE Trans. Geosci. Remote Sens.* **2001**, *39*, 309–320. [CrossRef]
11. 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]
12. Tarabalka, Y.; Fauvel, M.; Chanussot, J.; Benediktsson, J.A. SVM and MRF-based method for accurate classification of hyperspectral images. *IEEE Geosci. Remote Sens. Lett.* **2010**, *7*, 736–740. [CrossRef]
13. Ghamisi, P.; Maggiori, E.; Li, S.; Souza, R.; Tarablaka, Y.; Moser, G.; De Giorgi, A.; Fang, L.; Chen, Y.; Chi, M.; et al. New Frontiers in Spectral-Spatial Hyperspectral Image Classification: The Latest Advances Based on Mathematical Morphology, Markov Random Fields, Segmentation, Sparse Representation, and Deep Learning. *IEEE Geosci. Remote Sens. Mag.* **2018**, *6*, 10–43. [CrossRef]
14. Ma, X.; Fu, A.; Wang, J.; Wang, H.; Yin, B. Hyperspectral Image Classification Based on Deep Deconvolution Network With Skip Architecture. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 4781–4791. [CrossRef]
15. Shu, L.; McIsaac, K.; Osinski, G.R. Hyperspectral image classification with stacking spectral patches and convolutional neural networks. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 5975–5984. [CrossRef]
16. Zhang, M.; Li, W.; Du, Q. Diverse Region-Based CNN for Hyperspectral Image Classification. *IEEE Trans. Image Process.* **2018**, *27*, 2623–2634. [CrossRef]

17. 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]
18. Hu, W.; Huang, Y.; Wei, L.; Zhang, F.; Li, H. Deep convolutional neural networks for hyperspectral image classification. *J. Sens.* **2015**, *2015*, 258619. [CrossRef]
19. Yu, S.; Jia, S.; Xu, C. Convolutional neural networks for hyperspectral image classification. *Neurocomputing* **2017**, *219*, 88–98. [CrossRef]
20. Zhu, L.; Chen, Y.; Ghamisi, P.; Benediktsson, J.A. Generative Adversarial Networks for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 5046–5063. [CrossRef]
21. 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]
22. 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]
23. Li, W.; Wu, G.; Zhang, F.; Du, Q. Hyperspectral image classification using deep pixel-pair features. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 844–853. [CrossRef]
24. 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]
25. Xu, X.; Li, W.; Ran, Q.; Du, Q.; Gao, L.; Zhang, B. Multisource remote sensing data classification based on convolutional neural network. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 937–949. [CrossRef]
26. He, N.; Paoletti, M.E.; Fang, L.; Li, S.; Plaza, A.; Plaza, J. Feature Extraction With Multiscale Covariance Maps for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2018**, *99*, 1–15. [CrossRef]
27. Zhong, Z.; Li, J.; Luo, Z.; Chapman, M. Spectral–Spatial Residual Network for Hyperspectral Image Classification: A 3-D Deep Learning Framework. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 847–858. [CrossRef]
28. Liu, X.; Sun, Q.; Meng, Y.; Fu, M.; Bourennane, S. Hyperspectral Image Classification Based on Parameter-Optimized 3D-CNNs Combined with Transfer Learning and Virtual Samples. *Remote Sens.* **2018**, *10*, 1425. [CrossRef]
29. Paoletti, M.; Haut, J.; 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]
30. Mou, L.; Ghamisi, P.; Zhu, X.X. Deep recurrent neural networks for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 3639–3655. [CrossRef]
31. Wu, H.; Prasad, S. Convolutional recurrent neural networks forhyperspectral data classification. *Remote Sens.* **2017**, *9*, 298. [CrossRef]
32. Liu, Q.; Zhou, F.; Hang, R.; Yuan, X. Bidirectional-convolutional LSTM based spectral-spatial feature learning for hyperspectral image classification. *Remote Sens.* **2017**, *9*, 1330.
33. Ji, S.; Xu, W.; Yang, M.; Yu, K. 3D convolutional neural networks for human action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 221–231. [CrossRef] [PubMed]
34. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv* **2015**, arXiv:1502.03167.
35. Nair, V.; Hinton, G.E. Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th International Conference on Machine Learning (ICML-10), Haifa, Israel, 21–24 June 2010; pp. 807–814.
36. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
37. Graves, A.; Schmidhuber, J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Netw.* **2005**, *18*, 602–610. [CrossRef]
38. Sak, H.; Senior, A.; Beaufays, F. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In Proceedings of the Fifteenth Annual Conference of the International Speech Communication Association, Singapore, 14–18 September 2014.
39. Sundermeyer, M.; Schlüter, R.; Ney, H. LSTM neural networks for language modeling. In Proceedings of the Thirteenth Annual Conference of the International Speech Communication Association, Portland, OR, USA, 9–13 September 2012.

40. Vinyals, O.; Toshev, A.; Bengio, S.; Erhan, D. Show and tell: A neural image caption generator. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3156–3164.
41. Hochreiter, S. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* **1998**, *6*, 107–116. [CrossRef]
42. Pascanu, R.; Mikolov, T.; Bengio, Y. On the difficulty of training recurrent neural networks. In Proceedings of the International Conference on Machine Learning, Atlanta, GA, USA, 16–21 June 2013; pp. 1310–1318.
43. Bengio, Y.; Simard, P.; Frasconi, P. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Netw.* **1994**, *5*, 157–166. [CrossRef]
44. Xingjian, S.; Chen, Z.; Wang, H.; Yeung, D.Y.; Wong, W.K.; Woo, W.C. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; pp. 802–810.
45. Wu, H.; Prasad, S. Semi-supervised deep learning using pseudo labels for hyperspectral image classification. *IEEE Trans. Image Process.* **2018**, *27*, 1259–1270. [CrossRef]
46. Yang, X.; Ye, Y.; Li, X.; Lau, R.Y.; Zhang, X.; Huang, X. Hyperspectral Image Classification With Deep Learning Models. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 5408–5423. [CrossRef]
47. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
48. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 13–16 December 2015; pp. 1026–1034.
49. Tieleman, T.; Hinton, G. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Netw. Mach. Learn.* **2012**, *4*, 26–31.
50. Makantasis, K.; Karantzalos, K.; Doulamis, A.; Doulamis, N. Deep supervised learning for hyperspectral data classification through convolutional neural networks. In Proceedings of the 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Milan, Italy, 26–31 July 2015; pp. 4959–4962.
51. Li, Y.; Zhang, H.; Shen, Q. Spectral–spatial classification of hyperspectral imagery with 3D convolutional neural network. *Remote Sens.* **2017**, *9*, 67. [CrossRef]
52. Gibbons, J.D.; Chakraborti, S. *Nonparametric Statistical Inference*; Springer: Berlin, Germany, 2011.