remote sensing

MDPI

Article

Hyperspectral Image Classification via a Novel Spectral–Spatial 3D ConvLSTM-CNN

Ghulam Farooque [1], Liang Xiao [1,*], Jingxiang Yang [1] and Allah Bux Sargano [2]

[1] School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China; ghulam.farooque@njust.edu.cn (G.F.); yang123jx@njust.edu.cn (J.Y.)
[2] Department of Computer Science, COMSATS University Islamabad, Lahore 54000, Pakistan; allahbux@cuilahore.edu.pk
* Correspondence: xiaoliang@njust.edu.cn

Abstract: In recent years, deep learning-based models have produced encouraging results for hyperspectral image (HSI) classification. Specifically, Convolutional Long Short-Term Memory (ConvLSTM) has shown good performance for learning valuable features and modeling long-term dependencies in spectral data. However, it is less effective for learning spatial features, which is an integral part of hyperspectral images. Alternatively, convolutional neural networks (CNNs) can learn spatial features, but they possess limitations in handling long-term dependencies due to the local feature extraction in these networks. Considering these factors, this paper proposes an end-to-end Spectral-Spatial 3D ConvLSTM-CNN based Residual Network (SSCRN), which combines 3D ConvLSTM and 3D CNN for handling both spectral and spatial information, respectively. The contribution of the proposed network is twofold. Firstly, it addresses the long-term dependencies of spectral dimension using 3D ConvLSTM to capture the information related to various ground materials effectively. Secondly, it learns the discriminative spatial features using 3D CNN by employing the concept of the residual blocks to accelerate the training process and alleviate the overfitting. In addition, SSCRN uses batch normalization and dropout to regularize the network for smooth learning. The proposed framework is evaluated on three benchmark datasets widely used by the research community. The results confirm that SSCRN outperforms state-of-the-art methods with an overall accuracy of 99.17%, 99.67%, and 99.31% over Indian Pines, Salinas, and Pavia University datasets, respectively. Moreover, it is worth mentioning that these excellent results were achieved with comparatively fewer epochs, which also confirms the fast learning capabilities of the SSCRN.

Keywords: 3D ConvLSTM; hyperspectral image classification; 3D CNN; spectral-spatial feature extraction; residual network; deep learning

Citation: Farooque, G.; Xiao, L.; Yang, J.; Sargano, A.B. Hyperspectral Image Classification via a Novel Spectral–Spatial 3D ConvLSTM-CNN. *Remote Sens.* **2021**, *13*, 4348. https://doi.org/10.3390/rs13214348

Academic Editor: Edoardo Pasolli

Received: 30 August 2021
Accepted: 20 October 2021
Published: 29 October 2021

1. Introduction

It is now possible to acquire hyperspectral images (HSIs) with numerous contiguous spectral bands due to the advancement of imaging technology and hyperspectral sensors [1]. A hyperspectral image includes hundreds of small spectral bands with 2D spatial information of different land covers. The abundant spectral information enables HSIs to be successfully applied in various research fields, such as change detection [2], urban planning [3], precision agriculture [4], geology and mineral resources [5,6], national defense [7], and environment monitoring [8]. For these research domains, an important step is a robust and accurate image classification, which aims at identifying the unique category of each pixel. The HSI is a 3D data cube that combines spectral and spatial information, where the spatial resolution is shallow compared to the spectral resolution. It can only provide fewer details on the geometric relationship between image pixels. Therefore, spectral information is more important to identify the variety of ground materials accurately [9]. As an important research topic, HSI classification has received a great deal of attention, and several techniques have been proposed over the last few decades. However, due to

the complex nature of HSIs and the scarcity of labeled samples, it remains a challenging task [10].

Early classification techniques were mainly based on two steps: traditional handcrafted feature engineering followed by generic trainable classifiers [9]. Firstly, the most representative features were obtained by utilizing handcrafted feature descriptors and reducing the dimensionality of HSI [11]. Secondly, the traditional machine learning classifiers were trained through a non-linear transformation using the extracted features [12]. Support vector machines (SVM) and random forest (RF) are typical examples of these classifiers [13]. In this direction, a number of spatial feature descriptors such as local binary patterns (LBP) [14], Markov random fields (MRFs) [15], extended morphological profile (EMP) [16], spatial filtering [17], and 3D Gabor features [18] were employed to achieve suitable feature representation. However, due to the separation of feature extraction and classification processes, the adaptability between features and classifiers was not optimal [19,20]. Moreover, these methods had a limited capability to represent spectral and spatial information together due to the curse of dimensionality [10]. Consequently, the need for more robust feature extraction and classification methods was prevalent for better performance.

In recent years, deep learning-based approaches have superseded the traditional handcrafted feature-based methods for HSI classification. They have attracted a great deal of attention for employing an end-to-end strategy for feature extraction and classification that reduces the chances of information loss during the pre-processing of of HSI and improves the classification results [10,21,22]. Beside HSI classification, CNN-based techniques have also been employed for several tasks such as speech emotion recognition [23], human activity recognition [24], coin recognition [25], and music classification [26]. Initial deep learning-based methods employed 1D networks such as stacked autoencoder (SAE) [27], deep belief networks (BDNs) [28], recurrent neural networks (RNNs) [29], and CNNs [30]. These methods achieved better accuracy than their predecessor handcrafted approaches; however, due to 1D input requirements, these models suffered from spatial information loss.

Consequently, several 2D CNN-based methods were introduced for HSI classification [31]. In contrast to the previous methods, image patches were considered as inputs, and a weight-sharing mechanism was introduced to acquire the spatial and spectral information for classification. Along these lines, a CNN model with three convolutions and one fully connected layer was proposed by Zhao et al. [32]. Due to its simpler structure, this model learned spatial features but could not use spectral information. Another deep feature extraction approach based on the Siamese CNN that included a margin ranking loss function to ensure high interclass variability and low intraclass variability was reported in [33]. One of the significant challenges in the CNN-based model is learning the spectral features and combining them with spatial features. In this direction, Lee et al. [31] presented a deep contextual CNN technique to predict the label of each pixel by utilizing local spectral–spatial features. The spectral and spatial features were extracted from multi-scale filters using 2D CNN and then merged to produce a combined spectral–spatial feature map. However, 2D CNN-based techniques cannot effectively use spectral and spatial information at the same time, and chances of information loss are high during the feature learning process [34]. Another method [35] proposed a hybrid architecture based on 1D and 2D CNN layers to learn spectral and spatial information, respectively.

To address the limitations of 2D CNN based networks, 3D CNN models were adopted to obtain spatial–spectral features directly from raw HSI [36]. In this direction, a 3D CNN model was proposed in [37], where 3D image patches were considered as input to learn the spatial–spectral information from HSI. Li et al. [38] proposed a variation to this model to obtain spectral–spatial features simultaneously using spatial filtering to take full benefit of structural properties of 3D HSI data. However, 3D CNN approaches have many parameters and quickly lead to overfitting, particularly when training samples are limited. Zhong et al. [39] developed spectral and spatial residual blocks to alleviate the declining accuracy due to overfitting and learned the discriminative features from spatial

contexts and spectral signatures. As a variation to this method, Song et al. [40] introduced a deep feature fusion network based on residual learning to smooth the training of the deep model. Further, a densely connected 3D CNN (3D-DensNet) [41] was proposed to learn more robust spatial–spectral features. Lui et al. [42] introduced a content-guided CNN (CGCNN), which adaptively adjusts its kernel shape according to the spatial distribution of land covers. Another heterogeneous model based on CNN and graph convolutional network (GCN) was proposed in [43] to learn complementary spectral–spatial features at pixel and super-pixel levels. Other methods extended the 3D CNN by employing the concepts of attention mechanism [44], multi-scale convolution [45], and active learning [46] to improve classification accuracy. The major concern with the 3D CNN-based method is the degradation of accuracy due to very deep models as architectures become gradually complex with the growing number of hyper-parameters. Moreover, CNN-based models cannot handle long-term dependencies in spectral data [47].

Long Short-Term Memory (LSTM) [48] has proved its stability and strength in handling long-term dependencies in numerous computer vision tasks such as HSI classification [49] and speech emotion recognition [50]. Since HSIs are intensively sampled from the whole spectrum, dependencies between various spectral bands are expected. In this direction, Ienco et al. [48] proposed an LSTM-based model to perform classification tasks over multi-temporal satellite images. Further, Zhou et al. [49] introduced a spectral–spatial LSTMs (SSLSTMs) consisting of two independent LSTMs: spectral LSTM (SeLSTM) and spatial LSTM (SaLSTM). However, LSTM is limited in dealing with spectral–spatial and spatial–temporal data because it requires converting the input data into a 1D form, which causes a loss of essential spatial information. To overcome these limitations, Shi et al. [51] replaced the 1D data operation of each gate in LSTM with multi-dimensional processing and introduced convolutional LSTM (ConvLSTM). In ConvLSTM, 2D and 3D convolution filters can be used to construct the ConvLSTM2D and ConvLSTM3D, respectively. Motivated by this, Liu et al. [52] developed a model for obtaining spatial–spectral characteristics for HSI classification based on Bidirectional-Convolutional LSTM. Further, a spatial–spectral ConvLSTM2D neural network (SSCL2DNN) was introduced in [53] to manage the long-term dependencies and to obtain more discriminative features. Furthermore, ConvLSTM3D was used to simulate spatial and spectral information more efficiently.

Although deep learning-based approaches demonstrated encouraging performance for HSI classification, there are still several challenges that need to be addressed to achieve excellent performance. These include but are not limited to an imbalance between the high dimensionality of the data and the scarcity of training samples, the existence of mixed pixels in the data, and the integration of spectral and spatial information [54]. In addition to this, due to the high-dimensionality of hyperspectral data, techniques developed for low-dimensional spaces are not effective for HSI analysis. One possible solution could be to reduce the dimensionality of HSI data and then apply the classification techniques. However, this may cause a loss of crucial information [55]. These challenges are critical for the development of robust and efficient techniques for this purpose.

To overcome the abovementioned challenges, and motivated by the CNN-based residual learning and advanced ConvLSTM models, a novel spectral–spatial 3D ConvLSTM-CNN based residual network for HSI classification is introduced. The proposed SSCRN has two key modules: a 3D ConvLSTM spectral module and a 3D CNN spatial residual module. The 3D ConvLSTM module is directly applied to the original HSI to avoid losing crucial information and learn more discriminative spectral features. The 3D CNN spatial residual module is adapted to learn robust spatial information. The major contributions of our proposed SSCRN framework are as follows:

- The proposed framework SSCRN can learn both spatial and spectral feature representations jointly, without using any dimensionality reduction technique. The 3D ConvLSTM is exploited to learn the robust spectral feature representations, and the 3D CNN residual network is used to learn spatial features from HSI. This combination yields excellent performance.

- To the best of the authors' knowledge, this is the first time that 3D ConvLSTM and 3D CNN networks with skip connections are combined to build an end-to-end framework for HSI classification. This framework adopts residual connections to accelerate the training, mitigate the decreasing accuracy phenomenon, and improve the classification accuracy.
- The performance of the proposed framework is evaluated on three challenging benchmark datasets. The results confirm that SSCRN outperforms existing methods with limited labeled training samples.

The rest of the paper is organized as follows. Section 2 presents the background related to the core building block of the proposed model; Section 3 presents the proposed model; experimental results are given in Section 4; Section 5 offers the discussion; and finally, the paper is concluded in Section 6.

2. Background

In this section, a brief overview of CNN, LSTM, and ConvLSTM are presented; these models are the core building blocks of the proposed framework.

2.1. CNN

Typically, deep CNN includes three types of layers: convolutional, pooling, and fully connected layers. In convolutional layers, a set of learnable filters is convolved over the image to detect specific features and patterns in the image. Then, pooling layers reduce the number of parameters learned by convolutional layers by reducing the size of feature maps. The feature maps obtained from the pooling layers are transformed into feature vectors for classification using fully connected layers. The hyperspectral image is a cube consisting of multiple channels. For example, X is an input cube with a dimension of $w \times h \times s$, where $w \times h$ is the spatial size of the image and s is the number of channels. Then, outputs are obtained by convolving filters over the entire image, and bias terms are added to these outputs. Finally, an activation function is applied to generate activations as shown in Equations (1) and (2). These activations from layer 1 act as the input for the layer 2, and so on.

$$Z^{[l]} = W^{[l]} * a^{[l-1]} + b^{[l]} \tag{1}$$

$$a^{[l]} = g(z^{[l]}) \tag{2}$$

in this equation, $Z^{[l]}$ is the output of the current convolution layer, $W^{([l])}$ are weights of the current layer, $a^{[l-1]}$ is the activation of the previous layer, and $b^{([l])}$ is the bias of the current layer, whereas $a^{([l])}$ is the activation of the current layer, and g is the activation function.

2.2. LSTM

The recurrent neural network is a sequential deep learning model that effectively handles long-term dependencies in sequential data. The sequential data have sequences of timesteps, and the computation of the current timestep depends on the previous timestep. However, the major limitation of the RNN is its incapability to handle the problem of vanishing and exploding gradients. LSTM [56] was developed to solve this issue, which replaced a recurrent hidden node through a memory cell, functioning as an accumulator of state information. Various self-parametrized controlling gates can retrieve, update, and clear data from this cell. One of the primary advantages of employing gates and the memory cell is that they allow for the regulation of information flow, such that the gradient may travel over multiple timesteps without exploding or vanishing. The LSTM

unit is composed of four essential sub-units—the input gate i_t, output gate o_t, forget gate f_t, and memory cell c_t—which are computed as follows:

$$\begin{aligned} i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci} \circ c_{t-1} + b_i) \\ f_t &= \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf} \circ c_{t-1} + b_f\right) \\ c_t &= f_t \circ c_{t-1} + i_t \circ \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\ o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co} \circ c_t + b_o) \\ h_t &= o_t \circ \tanh(c_t), \end{aligned} \tag{3}$$

where *tanh* and "σ" are activation functions, x_t, and c_{t-1}, h_{t-1} denote the input of the current cell, state, and output of the cell, respectively. b_f, b_i, b_c, b_o are bias terms. The weight matrices are W_{xi}, W_{xf}, and W_{xo}, indicating the weights of the input, forget, and output gate, respectively, and "$\circ$" is a dot product.

2.3. ConvLSTM

The major shortcoming of LSTM is its incapability to handle spatio-temporal data effectively due to its fully connected architecture, which provides input-to-state and state-to-state transitions. To address this issue, ConvLSTM, an extended version of LSTM, was proposed [51]. In ConvLSTM, input-to-state and state-to-state transitions are performed using convolutional structures. The matrix multiplication is replaced with the convolution operation at each gate in the LSTM cell. There are two major variations of convolutional LSTM—i.e., ConvLSTM2D and ConvLSTM3D—which have different convolution structures and can be employed to model long-range dependencies in the spectral and time domains. The inner structure of ConvLSTM is illustrated in Figure 1. Where $X_1, \ldots, X_t$ inputs, $C_1, \ldots, C_t$ cell outputs, $H_1, \ldots, H_t$ hidden states and i_t, f_t, o_t gates of the ConvLSTM are 3D tensors. Moreover, the mathematical formulation of ConvLSTM is similar to LSTM as described in Equation (3), except for the convolution operation which needs to be added in the case of ConvLSTM.

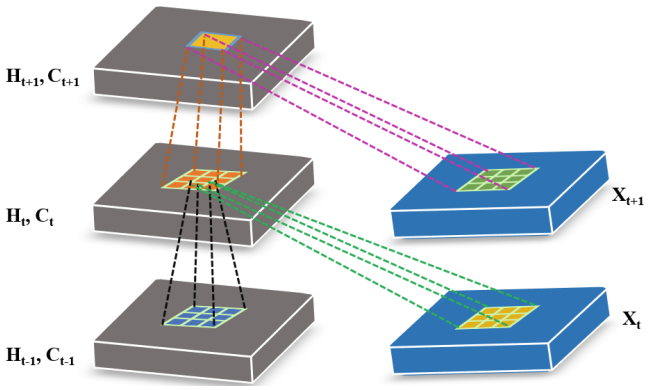


Figure 1. The inner structure of ConvLSTM.

ConvLSTM includes three gates that complete data processing and transmission, making it more convenient to use spectral information of HSI. In contrast to CNN, which uses a sliding window to extract spatial information, the ConvLSTM implements the intra-layer data processing in addition to inter-layer processing. This is another significant difference from the CNN [53]. The unique design allows ConvLSTM to extract more powerful feature representations from sequential data.

3. Proposed Methodology

This section presents the proposed spectral–spatial 3D ConvLSTM-CNN based Residual Network (SSCRN) for HSI classification. The SSCRN takes into account both spectral and spatial domains and the universality of the HSIs. Precisely, the proposed architecture consists of two modules; i.e., 3D ConvLSTM and 3D CNN. The 3D ConvLSTM plays a

role in learning robust spectral features, while 3D CNN aims to learn rich spatial features. The proposed architecture is shown in Figure 2 and explained in the subsequent sections.

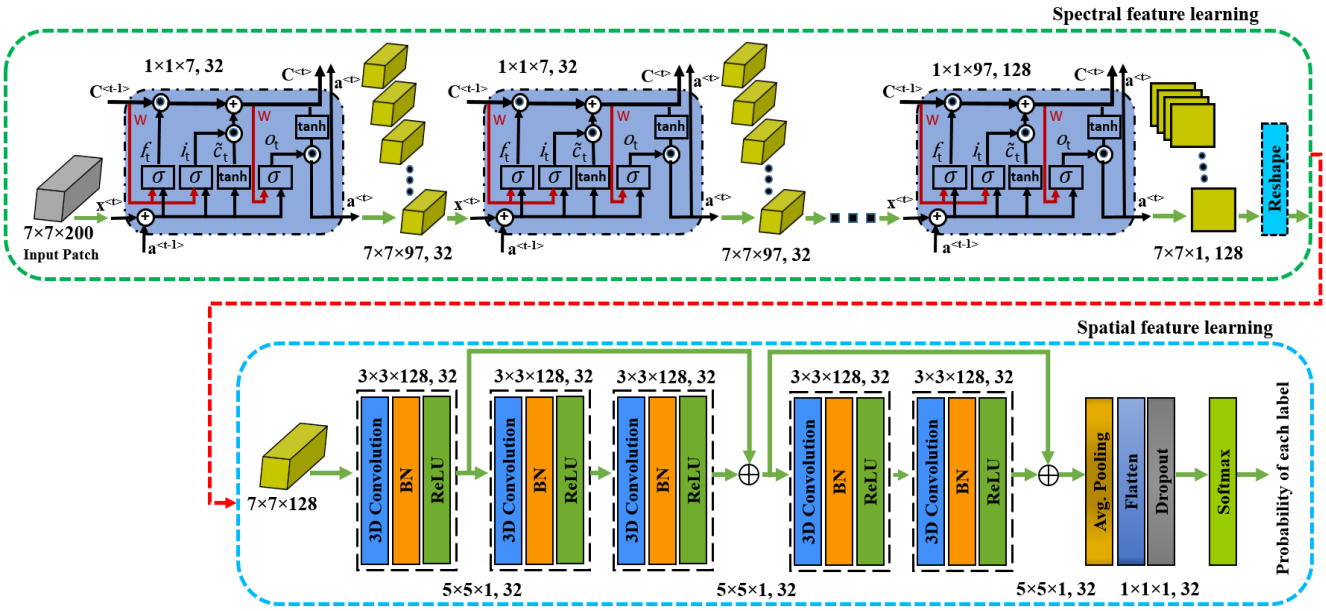


Figure 2. The proposed spectral–spatial 3D ConvLSTM-CNN-based Residual Network (SSCRN). An HSI image is decomposed into a sequence of patches used as input to the ConvLSTM cell. The <t> represents the output/activation of the current cell/layer of the ConvLSTM , while <t−1> represents the output/activation of the previous layer.

3.1. 3D ConvLSTM Spectral Module

HSIs have many spectral bands, and some earlier methods [40] applied unsupervised principal component analysis (PCA) to acquire spectral features but could not obtain good results due to insufficient discriminative features [57]. On the other hand, 2D ConvLSTM can perform reasonably well for addressing the issue of long-term dependencies; however, it cannot handle 3D data cubes properly. Moreover, 2D models do not preserve the intrinsic structure of the 3D cubes when taking each band as an input for the corresponding memory cell [53]. This motivates us to employ 3D ConvLSTM for discriminative spectral feature extraction and preserve the intrinsic structure of the data. Generally, a 3D ConvLSTM is considered the extension of 2D ConvLSTM: it has three gates where 3D data $w \times h \times s$ is taken as input for each memory cell of the 3D ConvLSTM. Moreover, the convolutional kernel in 3D ConvLSTM is of shape $f1 \times f2 \times d$, where $f1$, $f2$ and d are the kernel size and depth of the convolutional filter, respectively. The structure of the 3D ConvLSTM layer is shown in Figure 3. The equations of 3D ConvLSTM cell are written as

$$\begin{aligned}
\tilde{C}^{<t>} &= tanh\left(\left(W_c * a^{<t-1>} + W_c * x^{<t>}\right) + b_c\right) \\
i^{<t>} &= \sigma\left(\left(W_i * a^{<t-1>} + W_i * x^{<t>} + W_i \circ C^{<t-1>}\right) + b_i\right) \\
f^{<t>} &= \sigma\left(\left(W_f * a^{<t-1>} + W_f * x^{<t>} + W_f \circ C^{<t-1>}\right) + b_f\right) \\
o^{<t>} &= \sigma\left(\left(W_o * a^{<t-1>} + W_o * x^{<t>} + W_o \circ C^{<t>}\right) + b_o\right) \\
C^{<t>} &= i^{<t>} \circ \tilde{C}^{<t>} + f^{<t>} \circ C^{<t-1>} \\
a^{<t>} &= o^{<t>} \circ \tanh\left(C^{<t>}\right),
\end{aligned} \tag{4}$$

where $i^{<t>}$, $f^{<t>}$, $o^{<t>}$, and $C^{<t>}$ are the input, forget, output gate, and memory cell, respectively. $x^{<t>}$ and $a^{<t>}$ are the input and output of the current memory cell, while $C^{<t-1>}$ and $a^{<t-1>}$ are the state and output of the previous memory cell. $\tilde{C}^{<t>}$ is a candidate for replacing the memory cell. "σ" and $tanh$ are activation functions, "$*$" is a convolution operator, and "$\circ$" is a dot product, and b_f, b_i, b_c, and b_o are bias terms, as introduced in [51,58].

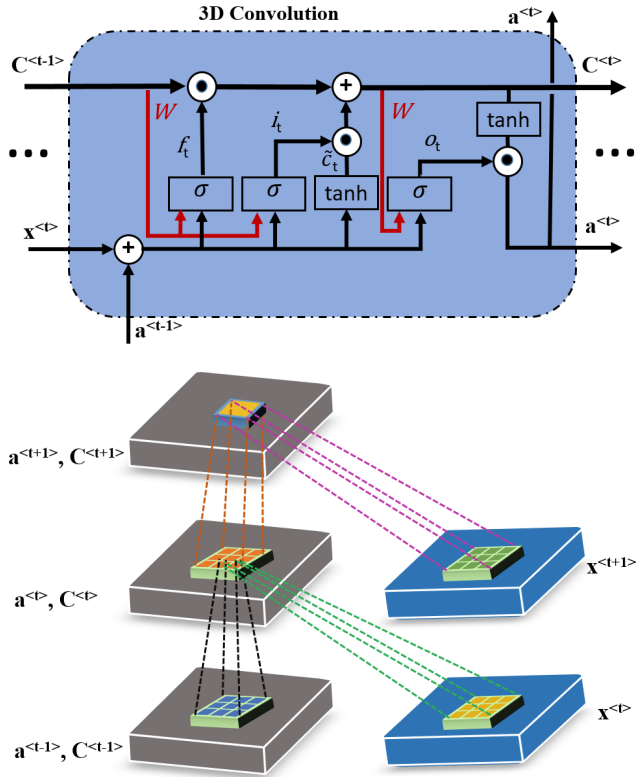


Figure 3. The internal structure of 3D ConvLSTM.

More specifically, the input $X^{<t>}$, the state $C^{<t-1>}$ and $C^{<t>}$, the output $a^{<t-1>}$ and $a^{<t>}$, and gate units $i^{<t>}$, $f^{<t>}$, and $o^{<t>}$ are 4D tensors with three spectral and two spatial dimensions, and the convolutional filters W_i, W_f, and W_o are 3D tensors. "$*$" is a 3D convolution between 4D input or output and 3D convolution filters. $X^{<t>} \in R^{\tau_t \times w_t \times h_t \times s_t}$ is the input of the 3D ConvLSTM cell, which is the tth component in a sequence decomposed from the input of the 3D ConvLSTM according to the dimension $time-step$, and $W_i \in R^{f_1 \times f_2 \times d}$, where τ_t, w_t, h_t, s_t, f_1, f_2, and d are the dimension $time-step$, width, height, number of the spectral band, kernel size and depth, respectively. Hence, the 3D convolution of $X^{<t>}$ and W_i can be defined as $W_i * X^{<t>}$. The output can be defined as $output^{\tau_t l_x l_y l_z}$ by $X^{<t>}$ at position (l_x, l_y, l_z) in the input gate described as follows in Equation (5), and the value of τ is fixed as 1.

$$Output^{\tau_t\ l_x\ l_y\ l_z} = \sum_{i1}^{f1} \sum_{m1}^{f2} \sum_{n1}^{d} W_i^{jpq}\ X^{<t>\ \tau_t (l_x+i)(l_y+m)(l_z+n)} \tag{5}$$

As represented above, the proposed spectral module can obtain the spectral features directly from HSI cubes. The unique structure of this module makes it an appropriate architecture for learning discriminative spectral features and preserving the intrinsic structure of data. This module is composed of four 3D ConvLSTM layers. In the first layer, a data patch of size $7 \times 7 \times 200$ is taken as an input. Then, 32 convolutional kernels of size $1 \times 1 \times 7$ with a stride of $(1, 1, 2)$ are convolved over the HSI patch to generate 32 feature columns with a size of $7 \times 7 \times 97$ as shown in Figure 2. Here, the convolution operation is performed only on the spectral channel, which also mitigates the redundant spectral information. The activation function for this layer is tanh, which is considered to be one of the most appropriate activations for LSTM layers with batch normalization. Next, the output of the first layer is passed to the second layer by applying the "same" padding to keep the input and output of the same size, while the rest of the settings are the same as the first layer. The output of the second layer is passed to the third layer, where the same settings are applied as layer 2. The last layer of the spectral module uses 128 convolution kernels of size $1 \times 1 \times 97$ in the Indian pines dataset, while for Salinas and the Pavia University, it becomes

$1 \times 1 \times 99$ and $1 \times 1 \times 49$, respectively. Finally, the output of the spectral module becomes the input for the spatial module for training the framework in an end-to-end fashion.

3.2. Deformable Process

The deformation process is required to convert the output of the spectral module into a 3D cube, making it suitable to be used as an input to the 3D CNN spatial module. In the spectral module discussed in Section 3.1, spectral feature maps are obtained with a size of $7 \times 7 \times 1,128$. Then, these feature maps are reshaped into a $7 \times 7 \times 128$ cube to be used as an input to the spatial module, as depicted in Figure 4. The spatial module processes it further and learns the discriminative and robust spatial features for classification. This approach enables the effective combination of spectral and spatial information through an end-to-end framework, as shown in Figure 2.

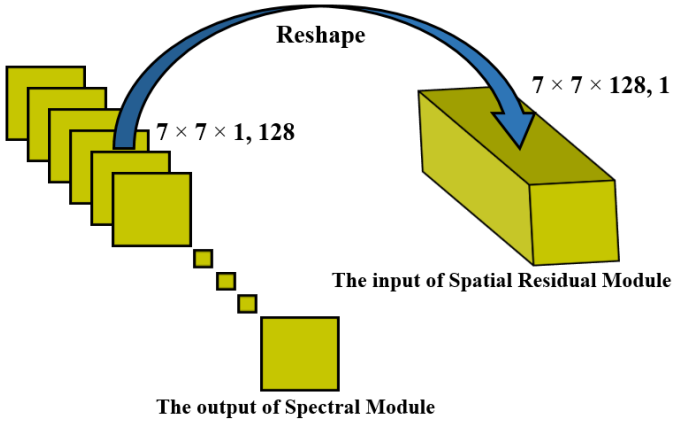


Figure 4. Overview of deformable process.

3.3. Three-Dimensional CNN Spatial Residual Module

In this module, the 3D CNN layers are adopted as the basic building blocks of SSCRN, and batch normalization (BN) is applied after each layer. The BN makes the training process more efficient and smoother. A 3D CNN layer has n^k input feature cubes of size $v^k \times v^k \times d^k$ with n^{k+1} filters of size $b^{k+1} \times b^{k+1} \times m^{k+1}$, and a stride of (s_1, s_1, s_2) for the convolutional operation. Then, this layer generates an output of n^{k+1} feature cubes of size $v^{k+1} \times v^{k+1} \times d^{k+1}$, where the spatial width $V^{k+1} = (1 + (V^k - b^{k+1})/s_1)$. The 3D CNN layer with BN can be represented as follows:

$$Z_i^{k+1} = REL\left(\sum_{l=1}^{n^k} \hat{Z}_l^k * W_i^{k+1} + b_i^{k+1}\right) \tag{6}$$

$$\hat{Z}^k = \frac{Z^k - E\left(Z^k\right)}{Std(Z^k)} \tag{7}$$

where $Z_l^k \in R^{v \times v \times d}$ is the lth input feature vector of $(k+1)$th layer, $\hat{Z}^k$ is the normalization result of batch feature Z^k in the kth layer, and Std(.) and $E(.)$ represent the standard deviation and expectation of the input feature vector, respectively. b_i^{k+1} and W_i^{k+1} denote the bias and weights of the ith filter in the $(k+1)$th layer, "$*$" represents the 3D convolution operator, and REL(.) is the activation function which transforms the negative numbers to zero.

In the case of a large CNN, the accuracy begins to decrease after a few layers [39]. This issue may be addressed by creating residual blocks by introducing shortcut connections between layers [47]. In this direction, two residual blocks are meant to obtain spatial information, where each block is composed of two convolutional layers, as illustrated in Figure 5.

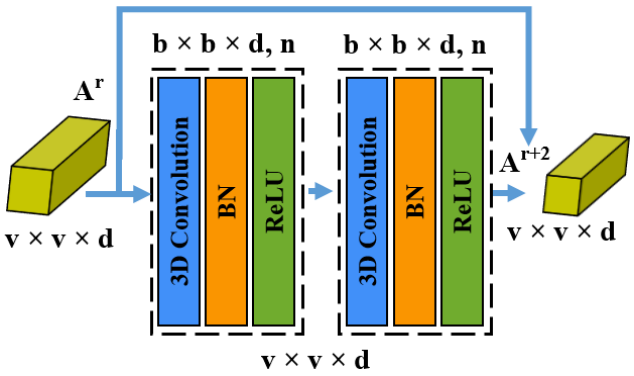


Figure 5. The spatial residual block consists of two successive 3D CNN layers, and a skip connection to add input feature maps A^r directly to the output feature maps A^{r+2}.

There are five 3D convolutional layers, including an initial layer followed by two residual blocks; each layer is supported by batch normalization and the rectified linear unit (ReLU). This formulation allows gradients in higher layers to propagate back to lower layers to regularize and smooth the training process. The first convolutional layer receives an input from the spectral module discussed in Section 3.1, which has an input size of $7 \times 7 \times 128$. Then, 32 kernels of size $3 \times 3 \times q$ are applied. Here, q is the number of channels of this layer with a stride of $(1, 1, 1)$, resulting in an output of $5 \times 5 \times 1, 32$ feature maps. Then, two residual blocks, each with two convolutional layers, are used to learn spatial representations using 32 kernels of size $3 \times 3 \times q$ at each layer. Here, the convolution operations are performed on spatial and spectral dimensions to carry on the complete information. After these residual blocks, an average pooling is applied to transform the obtained $5 \times 5 \times 1, 32$ spectral–spatial feature volume to a $1 \times 1 \times 1, 32$ feature vector. In addition to this, a dropout regularizer is also applied to address the overfitting problem. Subsequently, a fully connected (FC) layer receives input from the pooling layer and formulates a feature vector for classification. To validate the output produced by the model, a categorical cross-entropy loss is used, as shown in Equation (8). Finally, softmax layer activation is employed for the multi-class classification of different land cover categories.

$$Loss(Y, \hat{Y}) = -\sum(Y.\, log(\hat{Y})) \tag{8}$$

where Y is the ground truth and $\hat{Y}$ is the corresponding predicted value by our proposed model.

4. Experimentations and Results Analysis

The proposed SSCRN is implemented in Keras and Tensorflow deep learning frameworks using the Python language. The results are generated on a Lenovo Legion Y7000 Intel Core i7-9750H gaming machine equipped with Nvidia GeForce RTX2060-6G and 32G memory. The proposed SSCRN model is evaluated using three evaluation metrics—i.e., overall accuracy (OA), average accuracy (AA), and kappa coefficient (k)—while the higher values of OA, AA, and $kappa$ are considered better. Let $V \in R^{n \times n}$ represent the error matrix classification results, where n indicates the number of land-cover categories and the value of V in position (i, j) represents the number of ith category samples classified to the jth category. The formulas for these metrics are shown in Equations (9)–(11).

$$OA = sum(diag(V)/sum(V)) \tag{9}$$

$$AA = mean(diag(V)./sum(V, 2)) \tag{10}$$

$$Kappa = \frac{OA - (sum(V, 1)sum(V, 2))/sum(V)^2}{1 - (sum(V, 1)sum(V, 2))/sum(V)^2} \tag{11}$$

where diag$(V) \in R^{n\times 1}$ is a vector of diagonal values of V, $sum(.) \in R^1$ is the sum of values, $sum(.,1) \in R^{n\times 1}$ is the vector of the column-wise sum of all values, $sum(.,2) \in R^{n\times 1}$ is the vector which shows the row-wise sum of all values, $mean(.) \in R^1$ is the mean of all values, and ./ shows the element-wise division.

4.1. Experimental Data Sets

The performance of SSCRN is evaluated on three well-known benchmark datasets: Indian Pines, Salinas, and Pavia University. The Indian Pines (IP) dataset was acquired by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor over diverse agricultural and forest areas. In this dataset, the spatial size of the scene is 145 × 145 pixels, and the spatial resolution is 20 m per pixel (mpp). It contains 224 spectral bands across a wavelength ranging from 400 to 2500. In this experimentation, 200 bands are utilized after removing 24 noisy and zero-value bands. The land cover scene consists of 16 classes with 10,249 labeled pixels, where each class has different numbers of samples ranging from 20 to 2455. Each class's samples are divided into training, validation, and test sets, as presented in Table 1.

Table 1. Train, validation, and test split of IP dataset.

No.	Class Name	Train	Validation	Test	Total
1	Alfalfa	5	5	36	46
2	Corn-n	143	143	1142	1428
3	Corn-m	83	83	664	830
4	Corn	24	24	189	237
5	Grass-p	49	49	385	483
6	Grass-t	73	73	584	730
7	Grass-p-m	3	3	22	28
8	Hay-w	48	48	382	478
9	Oats	2	2	16	20
10	Soybean-n	98	98	776	972
11	Soybean-m	246	246	1963	2455
12	Soybean-c	60	60	473	593
13	Wheat	21	21	163	205
14	Woods	127	127	1011	1265
15	Buildings-g-t-d	39	39	308	386
16	Stone-s-t	10	10	73	93
	Total	1031	1031	8187	10,249

The Salinas (SA) is another important and well-known dataset used for hyperspectral image classification. An AVIRIS sensor was used to collect this dataset with 224 spectral bands over the Salinas Valley. Out of 224 spectral bands, 204 bands are utilized. This dataset has a spatial size of 512 × 217 pixels with a spatial resolution of 3.7 mpp with 16 different land cover classes. The training, validation, and test split of samples is shown in Table 2.

The Pavia University (PU) dataset was recorded by the reflective optics system imaging spectrometer (ROSIS) sensor over the University of Pavia. This dataset has 115 spectral bands; 103 bands are utilized after discarding noisy bands. The spatial size of this dataset is 610 × 340. There are nine classes with different numbers of samples, as listed in Table 3.

Table 2. Train, validation, and test split of SA dataset.

No.	Class Name	Train	Validation	Test	Total
1	Brocoli_g_w_1	101	101	1807	2009
2	Brocoli_g_w_2	187	187	3352	3726
3	Fallow	99	99	1778	1976
4	Fallow_r_p	70	70	1254	1394
5	Fallow_s	134	134	2410	2678
6	Stubble	198	198	3563	3959
7	Celery	179	179	3221	3579
8	Grapes_u	564	564	10,143	11,271
9	Soil_v_d	311	311	5581	6203
10	Corn_s_g_w	164	164	2950	3278
11	Lettuce_r_4wk	54	54	960	1068
12	Lettuce_r_5wk	97	97	1733	1927
13	Lettuce_r_6wk	46	46	824	916
14	Lettuce_r_7wk	54	54	962	1070
15	Vinyard_u	364	364	6540	7268
16	Vinyard_v_t	91	91	1625	1807
	Total	2713	2713	48,703	54,129

Table 3. Train, validation and test split of PU dataset.

No.	Class Name	Train	Validation	Test	Total
1	Asphalt	332	332	5967	6631
2	Meadows	933	933	16,783	18,649
3	Gravel	105	105	1889	2099
4	Trees	154	154	2756	3064
5	Painted metal sheets	68	68	1209	1345
6	Bare Soil	252	252	4525	5029
7	Bitumen	67	67	1196	1330
8	Self-blocking Bricks	185	185	3312	3682
9	Shadows	48	48	851	947
	Total	2144	2144	38,488	42,776

4.2. Experimental Settings

In the experimental setup, dataset division is considered an essential factor for the system's performance. For this purpose, a train/validation/test split strategy is selected to evaluate the proposed model. In the case of SA and PU datasets, 5%, 5%, and 90% split ratios are used for training, validation, and test, respectively, while for the IP dataset, the ratio was 10%:10%:80%. Further, the detailed information is given in Tables 1–3. Another important factor is the proper selection and tuning of hyperparameters of the model. Although all parameters are not equally important, appropriate values of parameters always lead to a balanced model. The important hyperparameters used in the proposed model are batch size, learning rate, regularizers, optimizers, and number of epochs. There are two well-known strategies for tuning parameters: grid search and random search. A random search is more suitable for deep learning models due to its capability to explore greater numbers of relevant parameter values. Therefore, a random search strategy has been adopted to find appropriate values for the hyperparameters used in the model. The appropriate value for the batch size is 32 for IP and 64 for SA and PU datasets.

In this way, among different optimizers, Adam [59] was selected for optimizing the model based on the tuning process. During optimization, the learning rate is considered one of the most important hyperparameters; its appropriate value plays an important role in the model's convergence. Consequently, optimal learning rates for IP, SA, and PU datasets are 0.0003, 0.0001, and 0.0001, respectively. Moreover, batch normalization and dropout are used during the training process to avoid overfitting. The proposed model was also assessed with and without regularization methods. The results confirm that

regularizing methods help to achieve better performance, as shown in Table 4. The network and parameter settings for the proposed SSCRN framework are summarized in Table 5. Moreover, the detailed comparison for hyperparameters of the proposed method against state-of-the-art techniques is presented in Table 6.

Table 4. OA accuracy of SSCRN with different regularizers.

Datasets	Dropout	BN	Dropout & BN
IP	96.90	98.69	99.17
SA	98.42	99.15	99.67
PU	97.97	98.81	99.31

Table 5. Network topology of the proposed SSCRN model.

Layer Name	Output Shape	Kernel Size	No. of Convolutional Kernel	Stride	Padding
ConvLSTM3D	7 × 7 × 97 × 32	1 × 1 × 7	32	1 × 1 × 2	Valid
ConvLSTM3D	7 × 7 × 97 × 32	1 × 1 × 7	32	1 × 1 × 1	Same
ConvLSTM3D	7 × 7 × 97 × 32	1 × 1 × 7	32	1 × 1 × 1	Same
ConvLSTM3D	7 × 7 × 1 × 128	1 × 1 × 97	128	1 × 1 × 1	N/A
Reshape	7 × 7 × 128 × 1	N/A	N/A	N/A	N/A
Conv3D	5 × 5 × 1 × 32	3 × 3 × 128	32	1 × 1 × 1	N/A
Conv3D	5 × 5 × 1 × 32	3 × 3 × 128	32	1 × 1 × 1	Same
Conv3D	5 × 5 × 1 × 32	3 × 3 × 128	32	1 × 1 × 1	Same
Skip connection					
Conv3D	5 × 5 × 1 × 32	3 × 3 × 128	32	1 × 1 × 1	Same
Conv3D	5 × 5 × 1 × 32	3 × 3 × 128	32	1 × 1 × 1	Same
Skip connection					
AveragePooling3D	1 × 1 × 1 × 32	N/A	N/A	1 × 1 × 1	N/A
Flatten	32	N/A	N/A	N/A	N/A
Dropout	32	0.25	N/A	N/A	N/A
Dense (Output)	N	N/A	N/A	N/A	N/A

Table 6. Comparison of hyperparameters with state-of-the-art methods.

Hyper-Parameter	3D-CNN	BASSNet	2D-3D CNN	SS3FC	ADR	FFDN-SY	TAP-Net	SSCRN
optimizer	SGD	Adam	RMSprop	Adam	-	Adam	Adam	Adam
Learning-rate	0.001, 0.0001	0.0005	0.001	0.01	0.0005, 0.001	0.01	0.01	0.0003, 0.0001
Batch size	10	100	100	-	-	200	32	32, 64
Dropout	0.5	0.5	0.3, 0.8	-	-	-	-	0.25
Activation	ReLU	ReLU	ReLU	ReLU	ReLU	ReLU	ReLU	tanh, ReLU
Iterations	100K	1000	5000	40-80	150–300	9000	-	300
Loss-function	-	Cross-entropy	-	Focal loss	softmax	-	Focal loss	Categorical cross-entropy

4.3. Classification Results

The proposed model is compared with several state-of-the-art models to prove its efficacy, including SVM [60,61], 3D-CNN [31], BASSNet [62], 2D–3D CNN [63], SS3FCN [10], ADR-3D-CNN [64], FCLFN [65], FFDN-SY [66], and TAP-Net [67]. It is worth noting that the proposed model achieved superior performance with a smaller or equivalent amount of training data compared to other state-of-the-art techniques except [65], as presented in Tables 7–9.

Table 7. Classification results (%) of different methods on the IP dataset. Bold indicates the best accuracy.

Class	SVM	3D-CNN	BASSNet	2D–3D CNN	SS3FC	ADR	FCLFN	FFDN-SY	TAP-Net	SSCRN
1	-	-	-	**100.0**	40.40	97.96	95.12	-	70.98	**100.0**
2	72.17	90.10	96.09	98.36	77.89	97.21	**99.38**	95.09	76.54	98.17
3	67.11	97.10	98.25	97.80	60.74	97.47	**100.0**	98.98	75.62	99.40
4	-	-	-	97.20	11.80	99.53	**100.0**	-	46.83	**100.0**
5	91.07	**100.0**	**100.0**	99.30	67.50	98.88	95.16	99.56	69.78	98.00
6	94.14	-	99.24	99.07	91.95	98.51	99.24	**99.67**	94.77	99.50
7	-	-	-	**100.0**	20.14	95.24	72.10	-	80.40	**100.0**
8	98.64	**100.0**	**100.0**	99.83	81.71	97.73	99.53	99.89	98.95	**100.0**
9	-	-	-	92.72	31.67	94.44	88.89	-	70.03	**100.0**
10	73.65	95.90	94.82	97.34	78.15	97.24	**99.54**	97.98	84.59	98.43
11	86.23	87.10	94.41	98.23	69.32	97.70	98.64	94.20	80.39	**99.53**
12	59.43	96.40	97.46	97.66	40.81	97.83	92.68	99.53	76.84	**99.58**
13	-	-	-	99.32	93.43	95.81	**100.0**	-	97.13	**100.0**
14	97.69	99.40	99.90	99.01	91.77	99.83	**99.91**	99.25	94.83	99.50
15	-	-	-	98.60	37.93	96.49	97.41	-	51.70	**99.34**
16	-	-	-	92.59	75.19	96.47	**97.59**	-	92.27	97.29
OA	82.58	93.61	96.77	98.33	71.47	97.89	98.56	96.96	81.35	**99.17**
AA	82.46	/	/	/	60.65	97.39	95.94	98.24	78.85	**99.29**
k	79.42	/	/	/	/	98.72	98.36	96.36	0.787	**99.05**

Table 8. Classification results (%) of different methods on the SA dataset. Bold indicates the best accuracy.

Class	SVM	3D-CNN	BASSNet	2D–3D CNN	SS3FC	ADR	FCLFN	FFDN-SY	TAP-Net	SSCRN
1	82.64	**100.0**	**100.0**	99.81	92.36	96.73	98.54	99.96	98.73	99.94
2	86.31	**100.0**	99.97	99.65	92.58	98.50	99.97	99.96	99.71	**100.0**
3	98.15	**100.0**	**100.0**	99.75	66.35	96.06	95.71	99.61	91.29	**100.0**
4	96.51	99.30	99.66	99.37	98.13	98.80	95.51	**99.88**	98.78	98.88
5	97.63	98.50	99.59	98.68	95.63	97.88	95.06	99.80	96.27	**100.0**
6	98.96	**100.0**	**100.0**	99.99	99.30	98.87	99.97	99.77	99.26	**100.0**
7	98.03	99.80	99.91	99.88	99.43	96.58	99.97	99.80	99.35	**100.0**
8	95.34	83.40	90.11	98.05	69.27	98.61	**99.71**	93.77	84.76	99.42
9	90.45	99.60	99.73	99.80	99.67	98.92	**100.0**	99.75	98.13	99.92
10	82.54	94.60	97.46	**99.86**	84.07	98.30	99.66	99.43	88.56	99.69
11	83.21	99.30	99.08	98.67	85.31	98.96	94.80	**99.98**	84.59	99.36
12	82.14	**100.0**	**100.0**	99.92	97.98	99.71	99.53	99.95	99.02	99.88
13	84.56	**100.0**	99.44	99.89	98.45	98.78	98.79	99.85	98.07	**100.0**
14	86.57	**100.0**	**100.0**	99.40	87.32	98.96	95.28	99.90	94.59	99.16
15	92.93	**100.0**	83.94	97.76	52.31	98.01	98.08	96.63	69.09	99.17
16	-	98.00	99.38	99.88	59.97	98.77	99.16	99.86	90.71	**100.0**
OA	94.82	95.07	95.36	99.07	81.32	98.29	98.59	98.04	90.31	**99.67**
AA	/	/	/	/	86.13	98.28	98.11	99.24	93.18	**99.71**
k	/	/	/	/	/	98.16	98.69	97.81	0.881	**99.64**

The first experiment was performed on the IP dataset. Some classes have few samples in this dataset, making it challenging for classification due to the imbalanced class problem. Therefore, existing methods have either excluded these classes from their experiments or have shown poor performance, as evident from Table 7. Despite the fact, SSCRN demonstrated 100% accurate results for these classes, which are "Oats", "Grass-p-m", and "Alfafa". Moreover, across all evaluation measures—i.e., OA, AA, and kappa coefficient—the SSCRN performed better than the compared methods with an OA of 99.17%, as presented in Table 7.

The second experiment was performed on the SA dataset. The SSCRN achieved an accuracy of 100% in the "Brocoli_g_w_2", "Fallow", "Fallow_s", "Stubble", "Celery", "Lettuce_r_6wk", and "Vinyard_v_t" categories. Moreover, across all evaluation measures—i.e., OA, AA, and kappa coefficient—the SSCRN performed better than state-of-the-art methods with an overall accuracy of 99.67%, as shown in Table 8.

The third experiment was conducted on the PU dataset. Compared with the IP dataset, this dataset has a relatively large number of training samples for each category with nine classes, while the IP has 16 classes. Due to sufficient training samples, many

deep learning models have achieved better performance on this dataset. However, due to interfering pixels, extracting discriminative features is still a challenging task for this dataset. The SSCRN produced excellent results on this dataset, as shown in Table 9.

Table 9. Classification results (%) of different methods on the PU dataset. Bold indicates the best accuracy.

Class	SVM	3D-CNN	BASSNet	2D–3D CNN	SS3FC	ADR	FCLFN	FFDN-SY	TAP-Net	SSCRN
1	93.84	94.60	97.71	99.42	97.48	98.01	97.03	98.24	95.67	**99.88**
2	95.88	96.00	97.93	99.93	90.86	98.20	**100**	98.90	97.61	99.92
3	72.80	95.50	94.95	98.69	58.75	98.15	95.14	98.07	73.08	**98.71**
4	88.23	95.90	97.80	**99.88**	84.81	99.23	88.49	97.82	94.23	99.52
5	98.05	**100.0**	**100.0**	99.97	94.82	99.31	99.18	99.93	99.48	**100.0**
6	84.51	94.10	96.60	99.45	23.59	99.41	**99.46**	**99.46**	84.17	96.58
7	82.70	97.50	98.14	99.47	61.61	98.92	95.89	99.79	59.92	**100.0**
8	88.37	88.80	95.46	97.89	88.84	98.08	**100.0**	98.52	83.60	98.61
9	99.56	99.50	**100.0**	99.96	88.68	98.06	96.20	99.69	99.33	**100.0**
OA	91.64	95.97	97.48	**99.54**	79.89	98.45	98.17	98.78	91.64	99.31
AA	89.33	/	/	/	76.60	98.60	96.80	98.93	87.45	**99.24**
k	88.88	/	/	/	/	98.53	97.58	98.36	0.892	**99.09**

For better understanding, the visual classification results for each dataset are presented in Figures 6–8 with false color images and their corresponding ground-truth maps. These maps also confirm the outstanding performance of SSCRN on the three HSI datasets discussed above. In particular, the SSCRN performed exceptionally well at the edges of land-cover areas and alleviated the effect of interfering pixels. In addition to this, it also suppressed spectral variability and produced smooth visual maps.

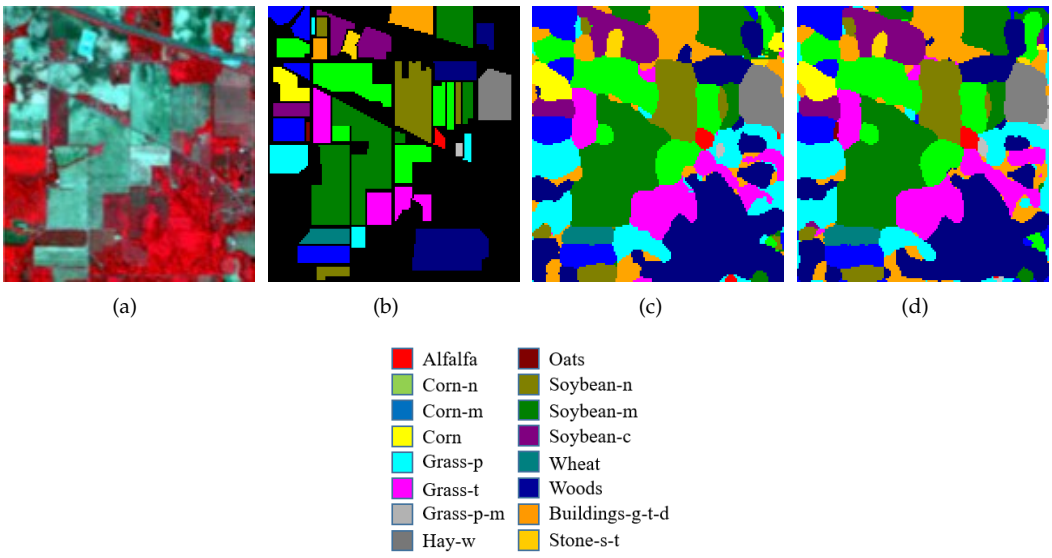


Figure 6. Visual classification results on the IP dataset. (**a**) A three-band color composite image, (**b**) the ground-truth image. Classification results of input image patch size (**c**) 7×7, (**d**) 11×11.

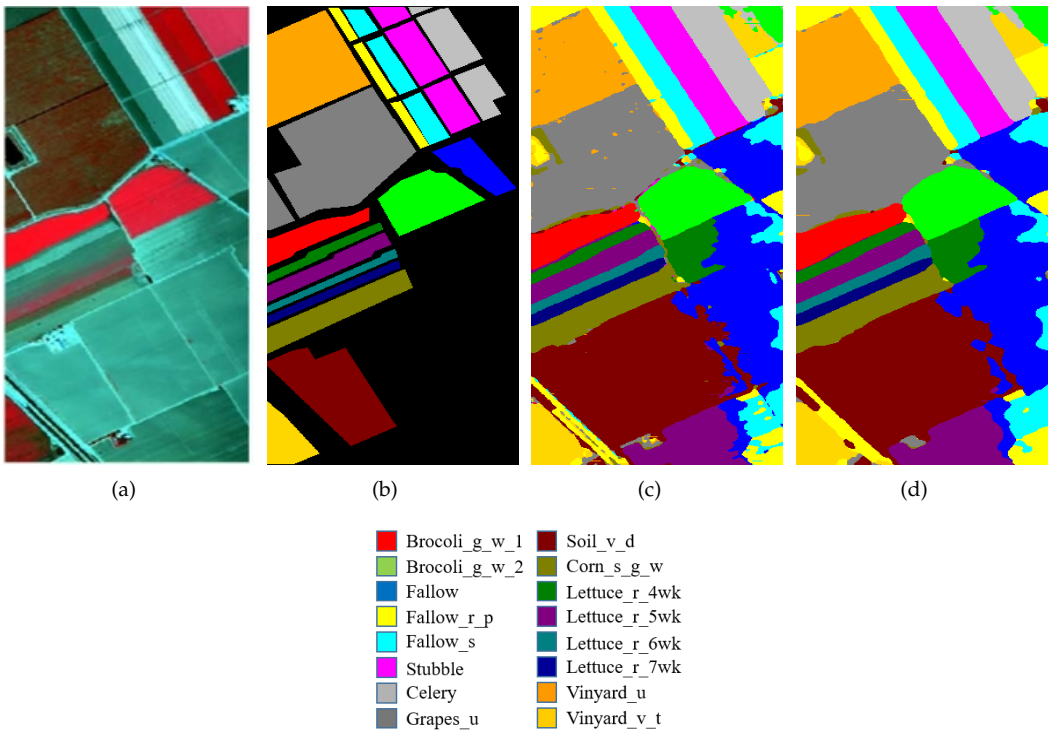


Figure 7. Visual classification results on the SA. (**a**) A three-band color composite image, (**b**) the ground-truth image. Classification results of input image patch size (**c**) 7×7, (**d**) 11×11.

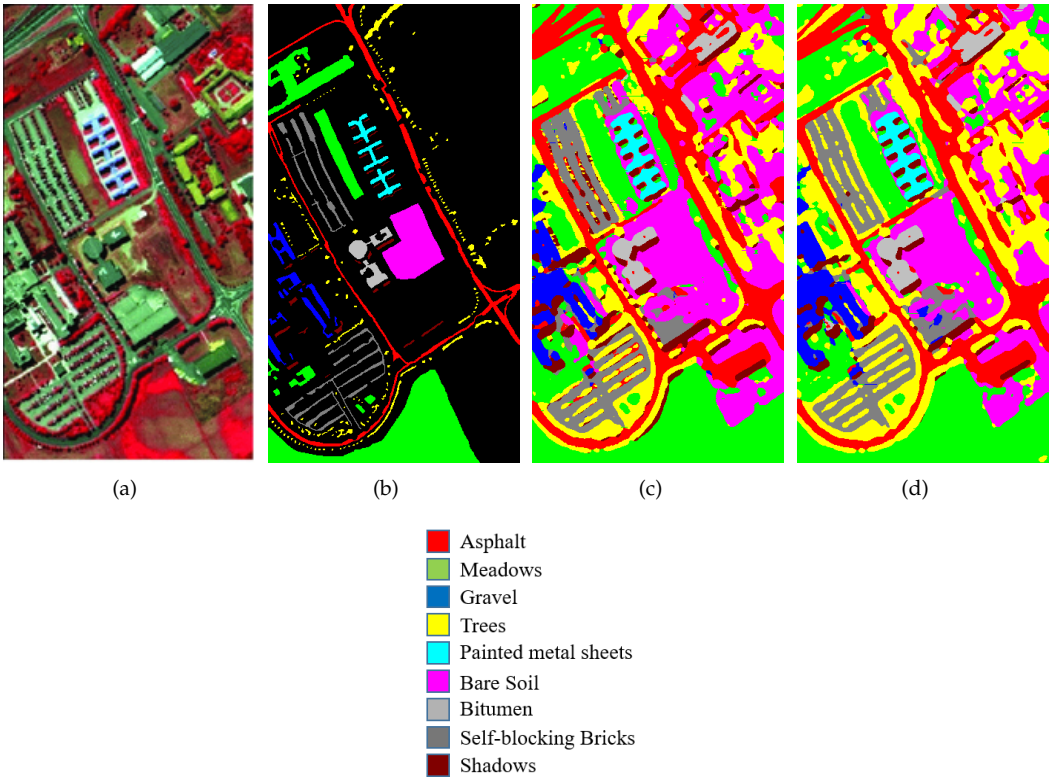


Figure 8. Visual classification results on the PU dataset. (**a**) A three-band color composite image, (**b**) the ground-truth image. Classification results of input image patch size (**c**) 7×7, (**d**) 11×11.

4.4. Impact of Training Ratio

The robustness and generalizability of the SSCRN were evaluated with a varying number of training samples from each dataset. In this regard, 5%, 10%, and 15% of the total samples were used from the IP dataset, and 2%, 5%, and 10% for the SA and PU datasets. The OA of SSCRN using different numbers of training samples is reported in Figure 9, which shows that a larger number of samples leads to higher accuracy.

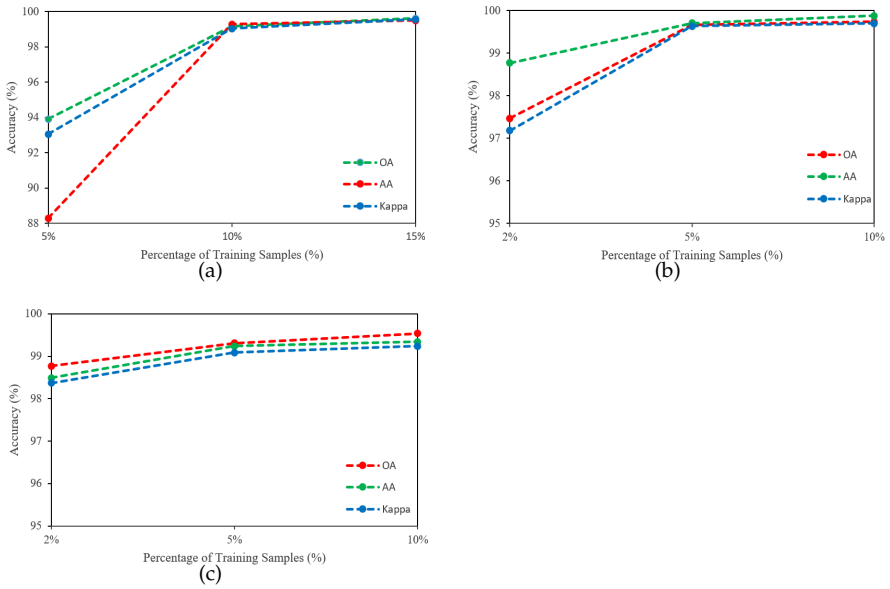


Figure 9. Accuracy (%) of SSCRN with different training percentages on three datasets: (**a**) IP, (**b**) SA, (**c**) PU.

4.5. Impact of Spatial Size of the Input Image Patches

The proposed model was also evaluated with different input patch sizes to assess the impact of input patches and find their optimal size. In this regard, the patch sizes of 5×5, 7×7, 9×9, and 11×11 were employed. It has been learned that a larger input patch size results in higher classification accuracy, which is quite reasonable because a larger spatial size contains more information. The larger patch size provides higher accuracy but at the cost of computational complexity. In this regard, we selected a spatial size of 7×7, which provides a good balance between high performance and computational complexity. The results with different patch sizes are presented in Figure 10 and Table 10.

Table 10. Accuracy (%) of SSCRN under different spatial sizes of the input image patches on different datasets. Bold indicates the best accuracy.

Spatial Input Size	IP (*OA*)	IP (*AA*)	IP (*k*)	SA (*OA*)	SA (*AA*)	SA (*k*)	PU (*OA*)	PU (*AA*)	PU (*k*)
5×5	97.55	98.85	97.78	97.45	98.40	97.16	99.04	98.28	98.72
7×7	99.17	**99.29**	99.05	99.67	99.71	99.64	99.31	99.24	99.09
9×9	99.21	99.20	98.83	**99.81**	**99.89**	**99.79**	**99.88**	**99.89**	**99.84**
11×11	**99.35**	99.28	**99.13**	99.75	99.78	99.65	99.82	99.60	99.76

4.6. Impact of the Number of Convolution Kernels

The number of convolution kernels in a network determines its computational complexity and representation capacity. As illustrated in Figure 2, the SSCRN employs the same number of kernels in both the spectral and spatial modules. In order to evaluate the effect of varying numbers of kernels on the accuracy, different numbers of kernels such as 16, 24, 32, and 40 are applied in each layer, as shown in Figure 11. The highest accuracy is achieved using 32 kernels in each layer in the case of all three datasets. Figure 11a represents the

detailed classification results on the IP dataset in terms of OA, AA, and kappa. Likewise, Figure 11b shows the detailed classification results of the SA dataset, and Figure 11c illustrates the classification results on the PU dataset.

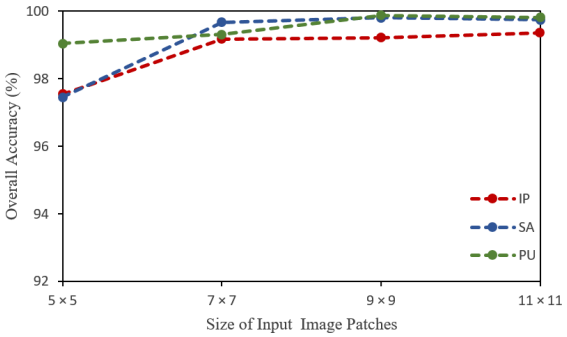
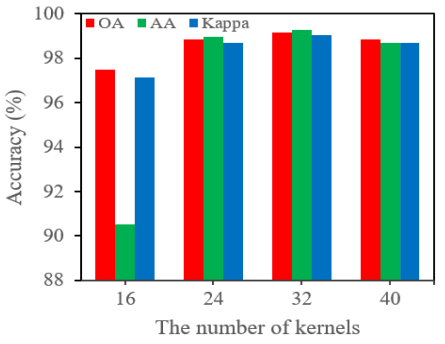
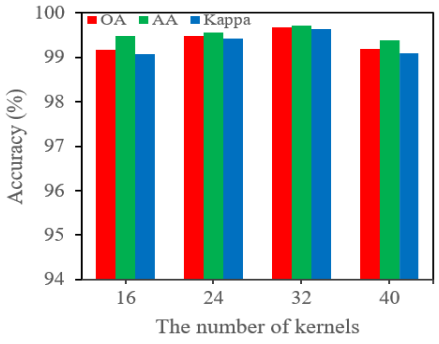


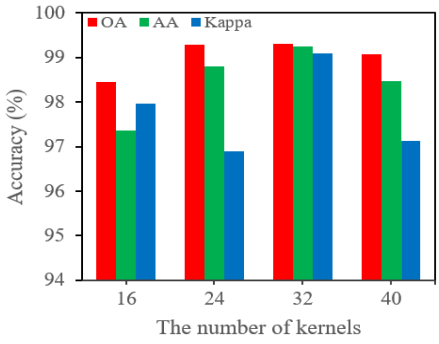
Figure 10. Overall accuracy (%) along with varying sizes of input image patches on IP, SA, and PU datasets.



(a)



(b)



(c)

Figure 11. Accuracy (%) of SSCRN with different numbers of kernels on three datasets: (**a**) IP, (**b**) SA, (**c**) PU.

4.7. Ablation Study

Although the SSCRN in its current form has proved to be effective for discriminative feature learning and HSI classification, an ablation study was performed to confirm the proposed model's robustness and generalization ability with different variations. For this purpose, the order of spectral and spatial modules was swapped. As a result of this change, the model first learned the spatial features using 3D CNN with residual blocks and then spectral features by employing the 3D ConvLSTM. The results confirm that the model also performed well with this arrangement of modules, but the accuracy was slightly decreased compared to its original version, as shown in Table 11. Moreover, it can also be concluded that arranging modules in a spectral–spatial fashion is superior to its spatial–spectral counterpart.

Table 11. Ablation study of swapping the sequence of spectral and spatial modules of SSCRN.

Sequence	Dataset	OA	AA	k
Spectral–Spatial	IP	99.17	99.29	99.05
	SA	99.67	99.71	99.64
	PU	99.31	99.24	99.09
Spatial–Spectral	IP	98.97	96.82	98.40
	SA	99.01	99.14	98.56
	PU	99.06	99.03	98.37

5. Discussion

Compared to traditional state-of-the-art approaches, deep learning-based techniques offer several benefits: automatic feature extraction from HSI data, hierarchical nonlinear transformation, objective functions that focus exclusively on classification rather than two independent steps, and the efficient use of computational resources such as GPU [39]. The proposed SSCRN harnesses the potential of 3D ConvLSTM and 3D CNN, which are considered excellent frameworks for different computer vision tasks, including HSI classification. However, the proposed SSCRN is significantly different from the existing techniques. There are four significant differences between SSCRN and other deep learning models. First, the SSCRN separates spectral and spatial features into two consecutive modules: 3D ConvLSTM and 3D CNN. The 3D ConvLSTM module aims to learn the robust spectral feature representations, and the 3D CNN learns spatial features. This allows for better discriminative features to be retrieved consecutively and reduces the chance of information loss. Second, the SSCRN employs residual connections to ensure that the network can work more deeply to enhance classification accuracy and avoid overfitting. Third, utilizing a BN operation at each layer, the model's fast learning capability can be ensured, and the model converges in fewer epochs. Fourth, the proposed SSCRN model achieved high classification accuracy, especially for the classes with few training samples. It is worth noting that, in this study, data augmentation [37] was not employed to increase the number of training samples; still, the model achieved excellent performance, which confirms the robustness of the SSCRN.

It has been learned that three key factors influence the performance of supervised deep learning models in terms of HSI classifications: (1) the spatial size of the input image patch, (2) the number of training samples, and (3) the representation ability of the proposed model. A greater number of training samples and a larger patch size lead to higher classification accuracy. Hence, the same number of training samples must be used for a fair comparison. For further clarification, the performance of SSCRN with different numbers of training samples and patch sizes were reported for three datasets, as shown in Figures 9 and 10. It has been observed that the performance in terms of OA declines with a reduction in the number of training samples and the patch size.

6. Conclusions

This paper presented a novel supervised deep learning framework consisting of two modules—i.e., a 3D ConvLSTM followed by a 3D CNN with residual blocks—to learn discriminative spectral–spatial representations for HSI classification. The input of the 3D ConvLSTM module was a sequence of local data patches fed into each memory cell to learn effective features and model long-term dependencies in the spectral domain. The output of the spectral module was converted into a 3D cube by applying a special transformation process, making it suitable for the 3D CNN spatial module. The 3D CNN spatial module was designed to extract robust features in the spatial context. This module employed the concept of skip connections similar to residual blocks to accelrate the training process and avoid overfitting. Then, fused spectral and spatial information was used for final classification. The proposed method achieved excellent results on three benchmark datasets, with an overall accuracy of 99.17%, 99.67%, 99.31% and the average accuracy of 99.29%, 99.71%, and 99.24% over Indian Pines, Salinas, and Pavia University datasets, respectively. The major advantage of the SSCRN is that it can be generalized to other remote sensing problems due to its robust design and in-depth feature learning capacity. However, it has been observed that the performance in terms of OA declines with a reduction in the number of training samples and the input image patch size. One of the possible extension of this work could be to make it computationally lean and achieve better accuracy with a reduced number of training samples and smaller patch size.

Author Contributions: G.F. and A.B.S. conceived and designed the experiments; G.F. performed the experiments. A.B.S. and G.F. analyzed the results; G.F. wrote the paper; A.B.S. improved the manuscript grammatically; J.Y. revised the manuscript; L.X. supervised and provided insightful advice to this study and revised the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China (Grants No. 61871226, 61571230), the Jiangsu Provincial Social Developing Project (Grant No. BE2018727).

Data Availability Statement: The HSI datasets used in this study are freely available at http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes (accessed on 30 August 2021).

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

References

1. Khan, M.J.; Khan, H.S.; Yousaf, A.; Khurshid, K.; Abbas, A. Modern trends in hyperspectral image analysis: A review. *IEEE Access* **2018**, *6*, 14118–14129. [CrossRef]
2. Wang, Q.; Yuan, Z.; Du, Q.; Li, X. GETNET: A general end-to-end 2-D CNN framework for hyperspectral image change detection. *IEEE Trans. Geosci. Remote Sens.* **2018**, *57*, 3–13. [CrossRef]
3. Sun, H.; Zheng, X.; Lu, X.; Wu, S. Spectral-Spatial Attention Network for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2019**, *58*, 3232–3245. [CrossRef]
4. Maes, W.H.; Steppe, K. Perspectives for remote sensing with unmanned aerial vehicles in precision agriculture. *Trends Plant Sci.* **2019**, *24*, 152–164. [CrossRef] [PubMed]
5. Tan, Y.; Lu, L.; Bruzzone, L.; Guan, R.; Chang, Z.; Yang, C. Hyperspectral Band Selection for Lithologic Discrimination and Geological Mapping. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 471–486. [CrossRef]
6. Yousefi, B.; Castanedo, C.I.; Bédard, É.; Beaudoin, G.; Maldague, X.P. Mineral identification in LWIR hyperspectral imagery applying sparse-based clustering. *Quant. Infrared Thermogr. J.* **2019**, *16*, 147–162. [CrossRef]
7. Shimoni, M.; Haelterman, R.; Perneel, C. Hypersectral Imaging for Military and Security Applications: Combining Myriad Processing and Sensing Techniques. *IEEE Geosci. Remote Sens. Mag.* **2019**, *7*, 101–117. [CrossRef]
8. Stuart, M.B.; McGonigle, A.J.; Willmott, J.R. Hyperspectral Imaging in Environmental Monitoring: A Review of Recent Developments and Technological Advances in Compact Field Deployable Systems. *Sensors* **2019**, *19*, 3071. [CrossRef] [PubMed]
9. Wu, P.; Cui, Z.; Gan, Z.; Liu, F. Three-Dimensional ResNeXt Network Using Feature Fusion and Label Smoothing for Hyperspectral Image Classification. *Sensors* **2020**, *20*, 1652. [CrossRef] [PubMed]
10. Zou, L.; Zhu, X.; Wu, C.; Liu, Y.; Qu, L. Spectral–Spatial Exploration for Hyperspectral Image Classification via the Fusion of Fully Convolutional Networks. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 659–674. [CrossRef]
11. Gan, Y.; Luo, F.; Liu, J.; Lei, B.; Zhang, T.; Liu, K. Feature extraction based multi-structure manifold embedding for hyperspectral remote sensing image classification. *IEEE Access* **2017**, *5*, 25069–25080. [CrossRef]

12. Fauvel, M.; Tarabalka, Y.; Benediktsson, J.A.; Chanussot, J.; Tilton, J.C. Advances in spectral-spatial classification of hyperspectral images. *Proc. IEEE* **2012**, *101*, 652–675. [CrossRef]
13. Belgiu, M.; Drăguţ, L. Random forest in remote sensing: A review of applications and future directions. *ISPRS J. Photogramm. Remote Sens.* **2016**, *114*, 24–31. [CrossRef]
14. Jia, S.; Hu, J.; Zhu, J.; Jia, X.; Li, Q. Three-dimensional local binary patterns for hyperspectral imagery classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 2399–2413. [CrossRef]
15. Xu, Y.; Wu, Z.; Wei, Z. Markov random field with homogeneous areas priors for hyperspectral image classification. In Proceedings of the 2014 IEEE Geoscience and Remote Sensing Symposium, Quebec City, QC, Canada, 13–18 July 2014; pp. 3426–3429.
16. Quesada-Barriuso, P.; Argüello, F.; Heras, D.B. Spectral–Spatial classification of hyperspectral images using wavelets and extended morphological profiles. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 1177–1185. [CrossRef]
17. He, L.; Chen, X. A three-dimensional filtering method for spectral-spatial hyperspectral image classification. In Proceedings of the 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Beijing, China, 10–15 July 2016; pp. 2746–2748.
18. Bau, T.C.; Sarkar, S.; Healey, G. Hyperspectral region classification using a three-dimensional Gabor filterbank. *IEEE Trans. Geosci. Remote Sens.* **2010**, *48*, 3457–3464. [CrossRef]
19. Yin, J.; Li, S.; Zhu, H.; Luo, X. Hyperspectral image classification using CapsNet with well-initialized shallow layers. *IEEE Geosci. Remote Sens. Lett.* **2019**, *16*, 1095–1099. [CrossRef]
20. Zhang, L.; Zhang, L.; Du, B. Deep learning for remote sensing data: A technical tutorial on the state of the art. *IEEE Geosci. Remote Sens. Mag.* **2016**, *4*, 22–40. [CrossRef]
21. Ma, A.; Filippi, A.M.; Wang, Z.; Yin, Z. Hyperspectral image classification using similarity measurements-based deep recurrent neural networks. *Remote Sens.* **2019**, *11*, 194. [CrossRef]
22. Zhang, M.; Li, W.; Du, Q.; Gao, L.; Zhang, B. Feature extraction for classification of hyperspectral and LiDAR data using patch-to-patch CNN. *IEEE Trans. Cybern.* **2018**, *50*, 100–111. [CrossRef] [PubMed]
23. Mustaqeem; Kwon, S. 1D-CNN: Speech Emotion Recognition System Using a Stacked Network with Dilated CNN Features. *CMC-Comput. Mater. Contin.* **2021**, *67*, 4039–4059. [CrossRef]
24. Sargano, A.B.; Wang, X.; Angelov, P.; Habib, Z. Human action recognition using transfer learning with deep representations. In Proceedings of the 2017 International joint conference on neural networks (IJCNN), Anchorage, AK, USA, 14–19 May 2017; pp. 463–469.
25. Farooque, G.; Sargano, A.B.; Shafi, I.; Ali, W. Coin recognition with reduced feature set sift algorithm using neural network. In Proceedings of the 2016 International Conference on Frontiers of Information Technology (FIT), Islamabad, Pakistan, 19–21 December 2016; pp. 93–98.
26. Ashraf, M.; Geng, G.; Wang, X.; Ahmad, F.; Abid, F. A Globally Regularized Joint Neural Architecture for Music Classification. *IEEE Access* **2020**, *8*, 220980–220989. [CrossRef]
27. 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]
28. 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]
29. 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]
30. Yang, J.; Zhao, Y.Q.; Chan, J.C.W. Learning and transferring deep joint spectral–spatial features for hyperspectral classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 4729–4742. [CrossRef]
31. Lee, H.; Kwon, H. Going deeper with contextual CNN for hyperspectral image classification. *IEEE Trans. Image Process.* **2017**, *26*, 4843–4855. [CrossRef] [PubMed]
32. 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]
33. Liu, B.; Yu, X.; Zhang, P.; Yu, A.; Fu, Q.; Wei, X. Supervised deep feature extraction for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *56*, 1909–1921. [CrossRef]
34. Feng, F.; Wang, S.; Wang, C.; Zhang, J. Learning Deep Hierarchical Spatial–Spectral Features for Hyperspectral Image Classification Based on Residual 3D-2D CNN. *Sensors* **2019**, *19*, 5276. [CrossRef] [PubMed]
35. Jia, S.; Zhao, B.; Tang, L.; Feng, F.; Wang, W. Spectral–spatial classification of hyperspectral remote sensing image based on capsule network. *J. Eng.* **2019**, *2019*, 7352–7355. [CrossRef]
36. Li, X.; Ding, M.; Pižurica, A. Deep Feature Fusion via Two-Stream Convolutional Neural Network for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2019**, *58*, 2615–2629. [CrossRef]
37. 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]
38. Li, Y.; Zhang, H.; Shen, Q. Spectral–spatial classification of hyperspectral imagery with 3D convolutional neural network. *Remote Sens.* **2017**, *9*, 67. [CrossRef]
39. 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.* **2017**, *56*, 847–858. [CrossRef]

40. Song, W.; Li, S.; Fang, L.; Lu, T. Hyperspectral image classification with deep feature fusion network. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 3173–3184. [CrossRef]
41. Zhang, C.; Li, G.; Du, S.; Tan, W.; Gao, F. Three-dimensional densely connected convolutional network for hyperspectral remote sensing image classification. *J. Appl. Remote Sens.* **2019**, *13*, 016519. [CrossRef]
42. Liu, Q.; Xiao, L.; Yang, J.; Chan, J.C.W. Content-guided convolutional neural network for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 6124–6137. [CrossRef]
43. Liu, Q.; Xiao, L.; Yang, J.; Wei, Z. CNN-enhanced graph convolutional network with pixel-and superpixel-level feature fusion for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2020**, *59*, 8657–8671. [CrossRef]
44. Fang, B.; Li, Y.; Zhang, H.; Chan, J.C.W. Hyperspectral images classification based on dense convolutional networks with spectral-wise attention mechanism. *Remote Sens.* **2019**, *11*, 159. [CrossRef]
45. Xu, Q.; Xiao, Y.; Wang, D.; Luo, B. CSA-MSO3DCNN: Multiscale Octave 3D CNN with Channel and Spatial Attention for Hyperspectral Image Classification. *Remote Sens.* **2020**, *12*, 188. [CrossRef]
46. Jamshidpour, N.; Aria, E.H.; Safari, A.; Homayouni, S. Adaptive Self-Learned Active Learning Framework for Hyperspectral Classification. In Proceedings of the 10th Workshop on Hyperspectral Imaging and Signal Processing: Evolution in Remote Sensing (WHISPERS), Amsterdam, The Netherlands, 24–26 September 2019; pp. 1–5.
47. 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.
48. Ienco, D.; Gaetano, R.; Dupaquier, C.; Maurel, P. Land cover classification via multitemporal spatial data by deep recurrent neural networks. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 1685–1689. [CrossRef]
49. Zhou, F.; Hang, R.; Liu, Q.; Yuan, X. Hyperspectral image classification using spectral-spatial LSTMs. *Neurocomputing* **2019**, *328*, 39–47. [CrossRef]
50. Kwon, S. CLSTM: Deep feature-based speech emotion recognition using the hierarchical ConvLSTM network. *Mathematics* **2020**, *8*, 2133.
51. 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.
52. 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. [CrossRef]
53. Hu, W.S.; Li, H.C.; Pan, L.; Li, W.; Tao, R.; Du, Q. Feature extraction and classification based on spatial-spectral convlstm neural network for hyperspectral images. *arXiv* **2019**, arXiv:1905.03577.
54. Bioucas-Dias, J.M.; Plaza, A.; Camps-Valls, G.; Scheunders, P.; Nasrabadi, N.; Chanussot, J. Hyperspectral remote sensing data analysis and future challenges. *IEEE Geosci. Remote Sens. Mag.* **2013**, *1*, 6–36. [CrossRef]
55. Liu, S.; Marinelli, D.; Bruzzone, L.; Bovolo, F. A review of change detection in multitemporal hyperspectral images: Current techniques, applications, and challenges. *IEEE Geosci. Remote Sens. Mag.* **2019**, *7*, 140–158. [CrossRef]
56. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
57. Mei, X.; Pan, E.; Ma, Y.; Dai, X.; Huang, J.; Fan, F.; Du, Q.; Zheng, H.; Ma, J. Spectral-spatial attention networks for hyperspectral image classification. *Remote Sens.* **2019**, *11*, 963. [CrossRef]
58. Hu, W.S.; Li, H.C.; Pan, L.; Li, W.; Tao, R.; Du, Q. Spatial–spectral feature extraction via deep ConvLSTM neural networks for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 4237–4250. [CrossRef]
59. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
60. Seifi Majdar, R.; Ghassemian, H. A probabilistic SVM approach for hyperspectral image classification using spectral and texture features. *Int. J. Remote Sens.* **2017**, *38*, 4265–4284. [CrossRef]
61. Plaza, J.; Plaza, A.J.; Barra, C. Multi-channel morphological profiles for classification of hyperspectral images using support vector machines. *Sensors* **2009**, *9*, 196–218. [CrossRef] [PubMed]
62. Santara, A.; Mani, K.; Hatwar, P.; Singh, A.; Garg, A.; Padia, K.; Mitra, P. BASS net: Band-adaptive spectral-spatial feature learning neural network for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 5293–5301. [CrossRef]
63. Yu, C.; Han, R.; Song, M.; Liu, C.; Chang, C.I. A simplified 2D-3D CNN architecture for hyperspectral image classification based on spatial–spectral fusion. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 2485–2501. [CrossRef]
64. Sellami, A.; Farah, M.; Farah, I.R.; Solaiman, B. Hyperspectral imagery classification based on semi-supervised 3-D deep neural network and adaptive band selection. *Expert Syst. Appl.* **2019**, *129*, 246–259. [CrossRef]
65. Zhao, G.; Liu, G.; Fang, L.; Tu, B.; Ghamisi, P. Multiple convolutional layers fusion framework for hyperspectral image classification. *Neurocomputing* **2019**, *339*, 149–160. [CrossRef]
66. Guo, H.; Liu, J.; Xiao, Z.; Xiao, L. Deep CNN-based hyperspectral image classification using discriminative multiple spatial-spectral feature fusion. *Remote Sens. Lett.* **2020**, *11*, 827–836. [CrossRef]
67. Qu, L.; Zhu, X.; Zheng, J.; Zou, L. Triple-Attention-Based Parallel Network for Hyperspectral Image Classification. *Remote Sens.* **2021**, *13*, 324. [CrossRef]