Deep Pyramidal Residual Networks for Spectral–Spatial Hyperspectral Image Classification

Mercedes E. Paoletti, *Student Member, IEEE*, Juan Mario Haut, *Student Member, IEEE*, Ruben Fernandez-Beltran, Javier Plaza, *Senior Member, IEEE*, Antonio J. Plaza, *Fellow, IEEE*, and Filiberto Pla

***Abstract*— Convolutional neural networks (CNNs) exhibit good performance in image processing tasks, pointing themselves as the current state-of-the-art of deep learning methods. However, the intrinsic complexity of remotely sensed hyperspectral images still limits the performance of many CNN models. The high dimensionality of the HSI data, together with the underlying redundancy and noise, often makes the standard CNN approaches unable to generalize discriminative spectral–spatial features. Moreover, deeper CNN architectures also find challenges when additional layers are added, which hampers the network convergence and produces low classification accuracies. In order to mitigate these issues, this paper presents a new deep CNN architecture specially designed for the HSI data. Our new model pursues to improve the spectral–spatial features uncovered by the convolutional filters of the network. Specifically, the proposed residual-based approach gradually increases the feature map dimension at all convolutional layers, grouped in pyramidal bottleneck residual blocks, in order to involve more locations as the network depth increases while balancing the workload among all units, preserving the time complexity per layer. It can be seen as a pyramid, where the deeper the blocks, the more feature maps can be extracted. Therefore, the diversity of high-level spectral–spatial attributes can be gradually increased across layers to enhance the performance of the proposed network with the HSI data. Our experiments, conducted using four well-known HSI data sets and 10 different classification techniques, reveal that our newly developed HSI pyramidal residual model is able to provide competitive advantages (in terms of both classification accuracy and computational time) over the state-of-the-art HSI classification methods.**

***Index Terms*— Convolutional neural networks (CNNs), hyperspectral imaging (HSI), residual networks (ResNets).**

Manuscript received February 26, 2018; revised May 31, 2018 and July 17, 2018; accepted July 22, 2018. Date of publication August 24, 2018; date of current version January 21, 2019. This work supported in part by the Ministerio de Educación (Resolución de 26 de diciembre de 2014 y de 19 de noviembre de 2015, de la Secretaría de Estado de Educación, Formación Profesional y Universidades, por la que se convocan ayudas para la formación de profesorado universitario, de los subprogramas de Formación y de Movilidad incluidos en el Programa Estatal de Promoción del Talento y su Empleabilidad, en el marco del Plan Estatal de Investigación Científica y Técnica y de Innovación 2013–2016, in part by the Junta de Extremadura (decreto 297/2014, ayudas para la realización de actividades de investigación y desarrollo tecnológico, de divulgación y de transferencia de conocimiento por los Grupos de Investigación de Extremadura) under Grant GR15005, in part by the Generalitat Valenciana under Contract APOSTD/2017/007, and in part by the Spanish Ministry of Economy under Project ESP2016-79503-C2-2-P. *(Corresponding author: Mercedes E. Paoletti.)*

M. E. Paoletti, J. M. Haut, J. Plaza, and A. J. Plaza are with the Hyperspectral Computing Laboratory, Department of Technology of Computers and Communications, Escuela Politécnica, University of Extremadura, PC-10003 Cáceres, Spain (e-mail: mpaoletti@unex.es; juanmariohaut@unex.es; jplaza@unex.es; aplaza@unex.es).

R. Fernandez-Beltran and F. Pla are with the Institute of New Imaging Technologies, University Jaume I, 12071 Castellón, Spain (e-mail: rufernan@uji.es; pla@uji.es).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TGRS.2018.2860125

I. Introduction

HYPERSPECTRAL imaging (HSI) collects valuable information for monitoring the surface of Earth [1], thus addressing important remote sensing applications, including environmental management [2], agriculture [3], surveillance [4], and physics [5]. In general, HSI science aims at acquiring data using hundreds of (narrow) spectral bands in order to simultaneously provide detailed spectral and spatial information. Therefore, HSIs are particularly useful for providing highly precise material identification by analyzing discriminative spectral and spatial features over specific areas of interest [6].

In the literature, different kinds of unsupervised and supervised approaches have been proposed to classify the HSI data [7]. Unsupervised methods do not make use of the labeled data, so they do not need a supervised training phase, which makes them suitable when poor prior knowledge of the considered scenes is available. In this sense, unsupervised clustering methods such as K-means [8] are used. Recently, more sophisticated unsupervised methods have been developed to efficiently extract a proper set of features for remote sensing data classification and segmentation purposes. In this sense, information theory approaches are showing an increasing potential in remote sensing data management and analysis, because they pursue to uncover hidden data interactions and correlations, which eventually can be very useful to deal with the inherent complexity of HSI data. For instance, Marinoni and Gamba [9] present a new unsupervised feature extraction approach based on data-driven discovery for data classification, which exploits mutual information maximization in order to retrieve the most relevant features. Another relevant information theory-based approach is the one in [10], where the authors present an efficient classification framework that relies on an entropy-based feature selection together with a

Pareto optimality criteria in order to detect relevant HSI data patterns for classification purposes.

Although unsupervised methods only rely on the data itself to categorize the pixels in the scene, supervised models have shown to provide more accurate results by learning the data relations from a given training set containing ground-truth information [11]. Over the past years, a wide variety of supervised machine learning paradigms have been successfully applied to remotely sensed HSI classification [12]. Support vector machines (SVMs) and kernel-based methods [13], statistical procedures as principal component analysis (PCA) [14] or logistic regression [15], Bayesian models [16], random forest (RF) [17], and neural networks [18] are amongst the most popular approaches.

Nonetheless, the intrinsic complexity of hyperspectral imagery still makes many of these approaches unable to consistently provide satisfactory classification results, especially under challenging scenarios [1]. Note that the number of training samples in the HSI field is usually rather limited compared to the available number of spectral bands, and this fact typically results in an undercomplete training process which is prone to overfitting, i.e., the so-called Hughes phenomenon [19]. Additionally, spectral redundancy and noise are often present in HSI since contiguous bands tend to be highly correlated, and the physical limitations of the acquisition technology always introduce some sort of signal perturbations.

Several strategies have been adopted in the remote sensing field to mitigate these problems and, consequently, improve the resulting HSI classification accuracy. This includes feature extraction [20]–[22], band reduction [23]–[26], data augmentation [27], and active learning techniques [28]–[30], [31]. However, one of the most popular research lines to deal with the high complexity of the HSI domain is based on developing spectral–spatial classifiers [6], which can achieve better classification performance than pixelwise classifiers, since they take into account not only the information of the spectral signatures but also the spatial–contextual information. He *et al.* [32] resort to discriminative low-rank Gabor filtering which is shown to be particularly effective for spatial–spectral HSI classification. Approaches such as this often pursue a reduction of classification uncertainty by combining each pixel spectra with the size and shape of the corresponding structure to which it belongs; therefore, highly powerful models are usually required to effectively exploit the HSI spectral–spatial components [33], [34].

In this scenario, supervised deep learning models are attracting increased attention. Deep network-based approaches [35], [36] have been recently introduced to the hyperspectral community, showing a great potential in the field of remote sensing classification. The main idea behind deep learning is to extract higher abstract semantic features from the original data with a hierarchical representation method. In other words, the supervised deep learning approach may be considered as a nonlinear mapping from the feature space to the label space, achieving higher expressibility through a hierarchy of layers. Chen *et al.* [18] proposed a stacked autoencoder (SAE) to extract the high-level features for HSI classification using spectral–spatial information. Zhao *et al.* [37] also exploited a stacked sparse autoencoder to extract layerwise more abstract and deep-seated features from spectral feature sets, spatial feature sets, and spectral–spatial vectors, using RF for classification purposes. Li *et al.* [38] introduced the deep belief network (DBN) for spectral–spatial feature extraction and classification of hyperspectral images. Zhong *et al.* [39] introduced a diversity promoting prior into the pretraining (unsupervised) and fine-tuning procedure (supervised) of the DBN model in order to enhance the HSI classification performance. However, these models suffer from spatial information loss, because they require flat spatial HSI patches (in one dimension) to satisfy their input requirements, and may not effectively exploit the spatial information [40]. Ma *et al.* [41] tried to overcome these limitations by implementing a spatial updated deep autoencoder in order to exploit jointly spectral and spatial features from HSIs, replacing each feature with the weighted average computed from the surrounding samples. To further address this issue, Chen *et al.* [42] proposed the use of convolutional neural networks (CNNs) for HSI classification. Compared to SAE and DBN, the CNN model allows using spatial HSI patches as data input, providing a natural way to incorporate this kind of information and enhance the classification performance.

Several CNN-based models can be found in the literature for HSI classification using spectral–spatial features. Following the pixel-based approach, Mei *et al.* [43] presented a CNN model integrating spectral signatures and spatial context by preprocessing each pixel, i.e., calculating the mean of the pixel neighborhood and the mean and standard deviation per spectral band of this neighborhood. Li *et al.* [44] combined the CNN model with pixel pairs to learn discriminative features, using a majority voting strategy to obtain the final classification result. Other relevant approaches are proposed by Yang *et al.* [45] and Zhang *et al.* [46], where they proposed two different CNN models to separately extract spectral and spatial features (the last one merging PCA with CNN), combining them by a softmax regression classifier. Moreover, Zhao and Du [47] combined a spatial feature extraction process based on the CNN model with a spectral feature extraction process based on the balanced local discriminant embedding, stacking the obtained features and then performing a final classification step. Although these methods merge different kinds of techniques in addition to CNNs to separately extract spectral–spatial information, they do not take full advantage of the joint spectral–spatial correlation information. In contrast, the deep models in [48]–[50] can learn both the spatial and the spectral information, taking as input data 3-D blocks from the original hyperspectral image and calculating 3-D convolution kernels for each pixel together with its spatial neighborhood and the corresponding spectral information.

However, training very deep CNNs with HSI data is still difficult, due to the loss of information produced by the vanishing gradient problem [51], where gradients obtained by the activation outputs of each processing layer of the network tend to be smaller, making a poor propagation of activations and gradients and elongating the cost function. As a result, the accuracy of deep CNNs is saturated and then degrades rapidly. Recently,

advanced deep CNN schemes have been proposed to uncover highly discriminative spectral–spatial features pervading the HSI data. It is the case of the residual network (ResNet) [52], which defines a CNN extension based on processing blocks, called residual blocks [53] as fundamental structural elements to facilitate learning of deeper networks and enabling them to be substantially deeper. These residual blocks are modules with the same topology that perform a set of transformations whose outputs are aggregated by summation. In fact, ResNet can be interpreted as a large ensemble of much shallower networks [54], creating a much deeper architecture than its plain counterparts, ensuring a minimum loss of information by modeling each block closer to an identity mapping than to a zero mapping, and adding shortcut connections between each residual block so that they receive more detailed information rather than just abstract information. As a result, ResNet models [53], [55], [56] may outperform standard deep CNNs in HSI analysis and classification [48], [57].

In this paper, we propose a new ResNet model based on pyramidal bottleneck residual units to achieve fast and accurate HSI analysis and classification, using both spectral and spatial information. This new deep model is composed of several blocks of stacked convolutional layers, which have a diabolo (bottleneck) architecture in which the output layer is larger than the input layer. In this way, the number of spectral channels in the original HSI cube is increased step by step on each block, creating the illusion of a pyramid where, as the residual units are deeper, more feature maps can be extracted, allowing to learn more robust spectral–spatial representations from HSI cubes. However, these HSI pyramidal bottleneck residual units are still computationally expensive, which forces to adopt acceleration techniques to reduce execution time. In this sense, the proposed network has been accelerated using graphical processing units (GPUs). The obtained results (using four well-known hyperspectral data sets) show that the proposed model can outperform not only the spectral–spatial CNN but also the baseline HSI-ResNet classification results, extracting more discriminative spectral–spatial features without the need to use large amounts of training data, which may have great uncertainty.

The remainder of this paper is organized as follows. Section II describes the proposed method. Section III validates the proposed model by drawing comparisons with other state-of-the-art HSI classification approaches. Finally, Section IV concludes this paper with some remarks and hints at plausible future research lines.

II. Methodology

This section is structured as follows. First, we set notation and provide an overview of classic CNNs while highlighting the connections of our newly proposed model with the traditional CNN architecture. Then, we introduce the proposed hyperspectral pyramidal ResNet model.

A. Convolutional Neural Networks

Traditional neural networks (deep or shallow ones) are characterized by 1-D architectures composed of fully connected (FC) layers, e.g., multilayer perceptrons (MLPs), AEs, or DBNs, which can lead to the loss of HSI structural information, in particular the intrinsic 2-D data information contained in the spatial domain of the hyperspectral images, because of the vector-based feature alignment of each layer [58]. Instead of that, CNN models are able to automatically exploit not only spectral information but also the relevant spatial–contextual features and also spectral–spatial features, depending on their architecture. Moreover, CNNs employ local connections defined in each layer to deal with spectral–spatial dependencies via sharing weights, i.e., layers are applied over defined and small regions of the input data, obtaining an output volume composed of feature maps which will be the input of the next layer.

Let us suppose an hyperspectral image $\mathbf{X} \in \mathbb{R}^{N\times W\times H}$, where N, W, and H are the spectral bands, width, and height respectively. The pixel $\mathbf{x}_{i,j}$ of $\mathbf{X}$ (with $i = 1, 2, \ldots, W$ and $j = 1, 2, \ldots, H$) can be defined as the spectral vector $\mathbf{x}_{i,j} \in \mathbb{R}^N = [x_{i,j,1}, x_{i,j,2}, \ldots, x_{i,j,N}]$. Also, we can define a neighboring region $\mathbf{p}_{i,j} \in \mathbb{R}^{d\times d}$ around $\mathbf{x}_{i,j}$, composed of pixels from $(i - (d/2), j - (d/2))$ to $(i + (d/2), j - (d/2))$ and from $(i - (d/2), j + (d/2))$ to $(i + (d/2), j + (d/2))$. If $\mathbf{p}$ takes into account the spectral information, it can be defined as $\mathbf{p}_{i,j} \in \mathbb{R}^{N\times d\times d}$. Depending on the architecture of the CNN layers and the kind of data that they use as input (the pixel vector $\mathbf{x}_{i,j} \in \mathbb{R}^N$, the spatial region $\mathbf{p}_{i,j} \in \mathbb{R}^{d\times d}$, or the spectral–spatial region $\mathbf{p}_{i,j} \in \mathbb{R}^{N\times d\times d}$), we can classify CNNs into three categories.

1) Spectral-based classification approaches, also called 1-D-CNNs, which are conceptually simple and easier to understand and implement because these models follow the pixel vector-based approach of traditional networks, being the spectral feature $\mathbf{x}_{i,j} \in \mathbb{R}^N$ of the original HSI data directly deployed as the input vector. As a result, each 1-D-layer obtains an output composed of n feature vectors, being n the number of filters or kernels.
2) Spatial-based classification approaches, also called 2-D-CNNs, which are the most widely used for image analysis and categorization tasks. In these models, the HSI is normally preprocessed via PCA or similar dimension reduction methods (such as independent component analysis [59] or maximum noise fraction [60], and among others) in order to reduce the number of spectral bands, and neighboring regions $\mathbf{p}_{i,j} \in \mathbb{R}^{d\times d}$ are extracted from the original image in order to create the input patches that 2-D-CNN models process to extract the spatial feature representation. As a result, each 2-D-layer obtains an output made up of n feature maps.
3) Spectral–spatial classification approaches, also called 3-D-CNNs, make use of a 3-D-architecture to jointly extract spectral–spatial information. In this case, neighboring spatial–spectral regions $\mathbf{p}_{i,j} \in \mathbb{R}^{N\times d\times d}$ are extracted from the original image in order to create the input data blocks that feed the network.

The proposed method makes use of 2-D-CNN approaches, implementing 2-D layers. However, all the spectral bands will be used in order to create the input data blocks $\mathbf{p}_{i,j} \in \mathbb{R}^{N\times d\times d}$ instead of reducing the original spectral signatures using PCA.

This will allow us to extract not only spatial information but also spectral information, in a fast and integrated way, performing a full spectral–spatial feature extraction and further allowing 3-D processing. In particular, four kinds of CNN layers will be used by the proposed architecture.

1) *Convolution layers* (CONV) that perform a dot product between their weights and biases and small windows of the input volume data defined by a kernel $k \times k$, obtaining an output volume composed of n feature maps, where n is the number of kernels

$$\mathbf{p}_{l+1} = \phi(\mathbf{W}_l \cdot \mathbf{p}_l + \mathbf{b}_l) \tag{1}$$

where $\mathbf{p}_{l+1}$ is the output with n feature maps of the lth CONV layer, $\mathbf{W}_l$ is weight matrix defined by the filter bank with kernel size $k \times k$, and $\mathbf{b}_l$ of the ith CONV layer, $\mathbf{p}_l$ is the output feature maps of the $l-1$th CONV layer, and $\phi(\cdot)$ the nonlinear activation function.

2) *Batch normalization layers* (BATCH-NORM) reduce the covariance shift by means of which the hidden unit values shift around, allowing a more independent learning process in each layer. It regularizes and speeds up the training process, imposing a Gaussian distribution on each batch of feature maps

$$\mathrm{BN}(x) = \frac{x - \mathrm{mean}[x]}{\sqrt{\mathrm{Var}[x] + \epsilon}} \cdot \gamma + \beta \tag{2}$$

where γ and β are learnable parameter vectors, respectively, and ϵ is a parameter for numerical stability.

3) *Nonlinearity layers* embed a nonlinear function applied to each feature map's component in order to learn nonlinear representations. In this layer, the rectified linear unit (ReLU) [61], [62] has been implemented.

4) *Pooling layers* (POOL) reduce data variance and computation complexity, making the features location-invariant summarizing the output of multiple neurons in CONV layers through a pooling function, e.g., max-pool or average-pool.

B. Proposed Hyperspectral Deep Network for Spectral–Spatial Classification

We denote a hyperspectral data cube as $\mathbf{X} \in \mathbb{R}^{N \times W \times H}$, containing two spatial dimensions: the width W and height H, and one spectral dimension, the number of spectral bands or channels N. In order to exploit both sources of information, we present a learning framework based on very deep CNNs, with the aim of performing accurate spectral–spatial HSI classification, taking into account the spectral signature of each pixel $\mathbf{x}_{i,j} \in \mathbf{X}$ and its spatial neighborhood. However, training very deep CNNs becomes more difficult as depth increases due to the loss of information produced by the vanishing gradient problem [51], where the activation outputs of the network produce a poor propagation of activations and gradients, being gradients close to zero, which elongates the cost function that must be optimized and cannot sufficiently change the model weights at each iteration. This hampers the convergence of the network from the beginning, where accuracy first saturates and then degrades rapidly.

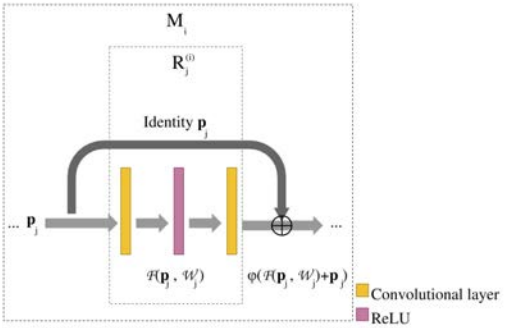


Fig. 1. Typical residual unit architecture $R_j^{(i)}$. $\mathcal{F}(\cdot) + \mathbf{p}_j$ is performed by the shortcut connection, with elementwise addition.

One of the most effective ways to solve the vanishing/exploding gradient problem is the use of a ResNet model [52], through a residual block-based [53] architecture. This model can be interpreted as a large ensemble of many grouped and shallower networks, similar to a matrioska. Let us consider a ResNet that is composed of M groups or modules. The ith module M_i, with $i = 1, 2, \ldots, M$, will be composed of $R^{(i)}$ residual units and the jth residual unit $R_j^{(i)}$ of M_i, with $j = 1, 2, \ldots, R^{(i)}$ composed of a few stacked layers, normally CONV layers stacked with ReLUs, and BATCH-NORM layers. In this architecture, two types of connections are given (see Fig. 1), the feedforward connection that connects layer-to-layer, i.e., each layer is connected with the previous one and the next one, and the skip or shortcut connection between each residual unit, i.e., a linear layer that connects the input of $R_j^{(i)}$ with its output, preserving information across layers. In this way, two operations are carried out related with these connections [see (3)], residual learning by feedforward connections and identity mapping by shortcut connections

$$\begin{aligned} \mathbf{y}_j &= h(\mathbf{p}_j) + \mathcal{F}(\mathbf{p}_j, \mathcal{W}_j) \\ \mathbf{p}_{j+1} &= \phi(\mathbf{y}_j) \end{aligned} \tag{3}$$

where $\mathbf{p}_j$ and $\mathbf{p}_{j+1}$ are the input and output feature maps of the jth residual unit, respectively, $\mathcal{W}_j = \{\mathbf{W}_l^{(j)} | 1 \le l \le L_j\}$ is the weight matrix of the L_j CONV layers associated with the jth residual unit, $\mathcal{F}(\cdot)$ is the residual function, $h(\mathbf{p}_j) = \mathbf{p}_j$ is the identity mapping, and $\phi(\cdot)$ is an activation function (normally a ReLU). The goal of the network is to learn the residual function $\mathcal{F}(\cdot)$ with respect to $h(\mathbf{p}_j) = \mathbf{p}_j$.

Also, in the ResNet, each $R_j^{(i)}$ shares the same topology, whose outputs are aggregated by summation and subject to two design rules: 1) for the same output feature map spatial size, the layers have the same number of filters n and 2) if the feature map size is halved, the number of filters n is doubled in order to preserve the time complexity per layer. The main idea behind this structure is that each residual unit is configured to perform the same recognition task as a single layer of the traditional CNN.

An interesting point of ResNets is the design of the residual blocks, depending on the size of the obtained feature maps of each CONV layer (as we can observe in Fig. 2 looking at the gray contours that indicate the size of each layer). As opposed

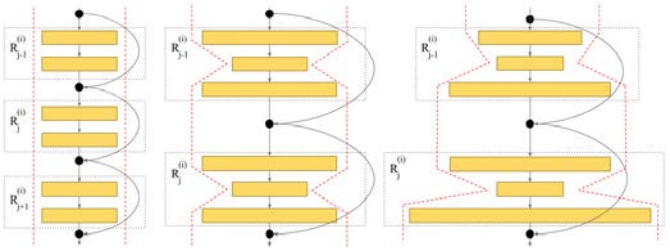


Fig. 2. Different residual unit architectures showing only CONV layers. (Left) Traditional residual units, where CONV layers have exactly the same topology. (Center) Bottleneck residual units, where feature maps are reduced and restored in depth for the input and output layers, maintaining the size between units. (Right) Pyramidal bottleneck residual units, where the number of channels of the CONV layers are gradually increased at every unit, resulting in progressively wider layers.

to traditional residual units, where each CONV layer shares the same topology, bottleneck residual units [52] have demonstrated to be more economical than the former, where the input and output CONV layers first reduce and then restore the depth dimension of the feature maps, allowing a faster execution of each residual unit. The pyramidal bottleneck residual unit [55] is a modification of the latter that outperforms the results of traditional residual units. This kind of units is characterized by a diabolo architecture, with the output layer being larger than the input layer (from the number of channels point of view), which imposes a processing on the identity mapping $h(\mathbf{p}_j) = \mathbf{p}_j$ because of the different depth sizes between the original input feature map $\mathbf{p}_j$ and the resulting feature maps of the residual function $\mathcal{F}(\mathbf{p}_j, \mathcal{W}_j)$. In order to solve this issue in a parameter-free way, pyramidal ResNets [55] implement a zero-padded shortcut, i.e., they add extra zero entries padded until reaching the increased dimension.

However, these residual units have been traditionally developed for only spatial feature extraction, in order to perform RGB image analysis and processing. Here we introduce, for the first time in the literature, a new residual unit inspired by pyramidal bottleneck residual units to perform the spectral–spatial classification of the HSI data. Fig. 3 provides a graphical illustration of our model architecture, which follows the same matrioska scheme of a ResNet. In this case, each module M_i is renamed as pyramidal module P_i, where the jth residual unit is implemented as a pyramidal bottleneck residual unit $B_j^{(i)}$. Also, this network implements zero-padded identity-mapping shortcut connections for each $B_j^{(i)}$, $h^*(\cdot)$.

Traditionally, CNNs are fed with a completely normalized image prior in order to perform classification. However, the HSI data typically exhibit land-cover classes that are highly mixed within the image $\mathbf{X} \in \mathbb{R}^{N \times W \times H}$, so each pixel $\mathbf{x}_{i,j} \in \mathbb{R}^N$ needs to be sent one by one to the network. In order to exploit spectral–spatial information, 3-D neighboring blocks around each $\mathbf{x}_{i,j}$ are extracted, denoted by $\mathbf{p}_{i,j} \in \mathbb{R}^{N \times d \times d}$, and sent to the model as input data, following a border mirroring method described in [50]. Moreover, the original HSI data $\mathbf{X}$ is normalized to zero mean and unit variance. Patches pass through five different modules, which compose the very deep neural network: one *input module* called C, three pyramidal modules called P_1, P_2, and P_3, and the final *output module*.

The input module C is made up of a CONV layer, with a kernel size $N \times k_1 \times k_1$ and a number of kernels n_1, followed by a BATCH-NORM layer. This module performs a first spectral–spatial feature extraction from the original input data, preparing its output feature maps for the rest of the network.

The next pyramidal modules, P_1, P_2, and P_3, are composed of three pyramidal bottleneck residual units each one, i.e., $B_1^{(i)}$, $B_2^{(i)}$, and $B_3^{(i)}$, with $i = \{1, 2, 3\}$. At this point, a new architecture for the pyramidal bottleneck residual units has been implemented in order to perform spectral–spatial HSI feature processing. As we can observe in Fig. 3, each $B_j^{(i)}$ is made up of several stacked layers, in particular three CONV layers, preceded by the corresponding BATCH-NORM layers, with a ReLU activation function at the end of the unit. Specifically, the distribution of the layers can be summarized as follows: BATCH-NORM$_1$ − CONV$_1$ − BATCH-NORM$_2$ − CONV$_2$ − BATCH-NORM$_3$ − CONV$_3$ − ReLU.

In order to exploit the spectral–spatial information contained in the HSI data, the lth CONV layer of the jth residual unit has been implemented with a filter bank defined by its own kernel size, $n_{l-1}^{(j)} \times k_l^{(j)} \times k_l^{(j)}$, and its own number of kernels, $n_l^{(j)}$. As a result, each CONV layer takes into account all the spectral information contained in its input feature maps, which is defined by the number of feature maps of the previous layer $n_{l-1}^{(j)}$, and processes the spatial information within a window over the feature maps defined by $k_l^{(j)} \times k_l^{(j)}$. In this way, each layer exploits both kinds of features spectral and spatial, computing its output feature maps via (1), with $n_l^{(j)}$ maps.

Moreover, following the implemented spectral–spatial pyramidal bottleneck residual block $B_j^{(i)}$, the output feature map can be obtained by reformulating (3) as follows:

$$\begin{aligned} \mathbf{y}_j^{(i)} &= h^*\left(\mathbf{p}_j^{(i)}\right) + \mathcal{F}\left(\mathbf{p}_j^{(i)}, \mathcal{W}_j^{(i)}\right) \\ \mathbf{p}_{j+1}^{(i)} &= \phi\left(\mathbf{y}_j^{(i)}\right) \\ &\quad \text{with } \mathcal{F}\left(\mathbf{p}_j^{(i)}, \mathcal{W}_j^{(i)}\right) \text{ equals to:} \\ \text{for } l \text{ in } L\text{: } \mathbf{p}_j^{(i)} &= \mathbf{W}_l^{(j)} \cdot \mathrm{BN}\left(\mathbf{p}_j^{(i)}\right) + \mathbf{b}_l^{(j)} \end{aligned} \tag{4}$$

where $\mathbf{p}_j^{(i)}$ and $\mathbf{p}_{j+1}^{(i)}$ are the input and output feature maps of the pyramidal bottleneck residual unit $B_j^{(i)}$, respectively, $h^*(\mathbf{p}_j^{(i)})$ is the zero-padded identity-mapping shortcut connection, $\mathcal{W}_j^{(i)}$ denotes all the weights and biases of each CONV layers associated with $B_j^{(i)}$, where L_j is the number of CONV layers, $\mathcal{F}(\mathbf{p}_j^{(i)}, \mathcal{W}_j^{(i)})$ is the dot product between the input feature maps and the CONV layers weights where $\mathcal{W}_j = \{\mathbf{W}_l^{(j)} | 1 \le l \le L_j\}$ being $\mathbf{W}_l^{(j)}$ and $\mathbf{b}_l^{(j)}$ the weight matrix and bias vector of the lth CONV layer, ϕ is the ReLU activation function, and BN$(\cdot)$ is the batch-normalization of the data. We must highlight that P_1 keeps the spatial feature size, setting the strides in all the CONV layers of each $B_j^{(1)}$ equal to $s = 1$. However, P_2 and P_3 implement two different mechanisms to perform downsampling over the input data. As we can see, in the first residual unit of both modules—$B_1^{(2)}$ and $B_1^{(3)}$—there is a CONV layer (in particular CONV$_2$) with stride equal to $s = 2$ and a downsampling layer added

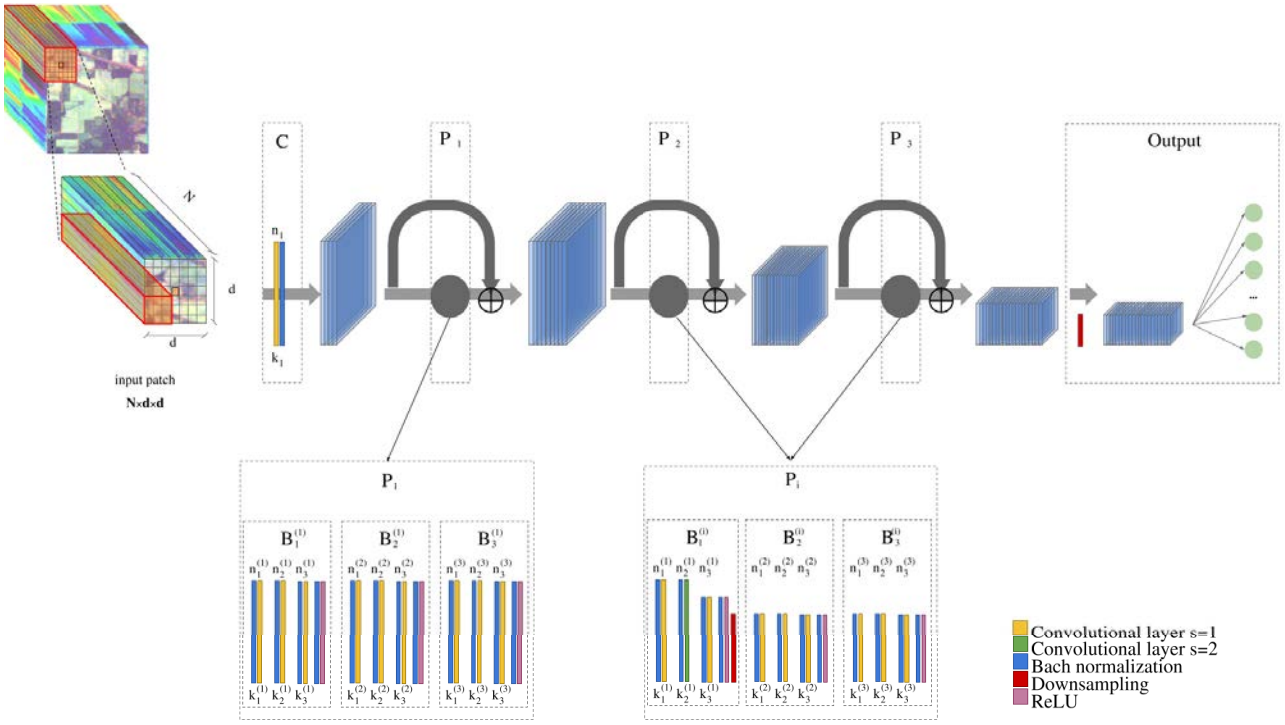


Fig. 3. Proposed hyperspectral pyramidal ResNet architecture model. The input block $\mathbf{p}_{i,j} \in \mathbb{R}^{N\times d\times d}$ is passed through five different modules that compose the hyperspectral pyramidal ResNet: C, P_1, P_2, P_3, and the output modules. C is composed of a CONV and a BATCH-NORM layers, while P_1, P_2, and P_3 modules, also called pyramidal modules, are composed of three pyramidal bottleneck residual units ($B_1^{(i)}$, $B_2^{(i)}$ and $B_3^{(i)}$, being $i = \{1, 2, 3\}$ the pyramid layer). These residual units are composed of three BATCH-NORM layers followed by their corresponding CONV layers and with a ReLU at the end of the unit. Instead of P_1, which maintains the spatial size, P_2 and P_3 reduce the data space adding strides equal to $s = 2$ (green CONV layer) and a downsampling layer. Finally, the output module is composed of a downsampling layer and an FC layer that performs the final classification. Each CONV layer has its own number of filters and kernel sizes, n_1 and k_1 for the first module and $n_l^{(j)}$ and $k_l^{(j)}$ for the pyramid layers (being $j = 1, 2, 3$ the jth residual unit $B_j^{(i)}$ and $l = 1, 2, 3$ the number of the lth CONV layer). The FC layer is composed of N_{class} neurons, being N_{class} the number of different land-cover classes in the original HSI data.

at the end of the unit. This last downsampling layer applies an average pooling over the input data in order to reduce data variance and extract low-level features from the spatial neighborhood, feeding those to the next layer. At this point, it is interesting to point that, instead of following the traditional two rules of residual units, the pyramidal ResNet approach has been adopted in order to calculate the depth at the end of each $B_j^{(i)}$, called $N_j^{(i)}$, attempting to gradually increase the depth of the feature map at each unit instead of doubling it in certain units, which allows to distribute the computational burden associated with the increase in the feature maps in an uniform way. In particular, (5) [55] has been adopted in order to linearly increase the depth of feature maps at each residual unit

$$N_j^{(i)} = \begin{cases} A, & \text{if } i = 1 \text{ and } j = 1 \\ \left\lfloor N_{j-1}^{(i)} + \dfrac{\alpha}{N^{(\text{net})}} \right\rfloor, & \text{otherwise} \end{cases} \tag{5}$$

where A is the initial depth of the input volume data, $N_j^{(i)}$ is the dimensionality of the feature map associated with the jth residual unit, $B_j^{(i)}$, which belongs to the ith module, P_i, and $N^{(\text{net})} = \sum_{i=1}^{P} B^{(i)}$ represents the total number of residual units, being P and $B^{(i)}$ the number of pyramid modules and the number of pyramidal bottleneck residual units per module, respectively.

Finally, the output feature maps of the last pyramidal module P_3 are downsampled one last time with average pooling, and reshaped into a vector in order to feed an FC layer that is added at the end of the network in order to perform the final classification task. On the other hand, the neural model has been optimized using the stochastic gradient descent method, with 200 epochs in the comparative experiments and a variable learning rate, with LR $= 0.1$ from epochs 1 to 149 and LR $= 0.01$ from epochs 150 to 200.

Table I summarizes the proposed architecture by stating the value of each of the kernel sizes and the number of filters employed in each CONV layer. The number of kernels $n_l^{(j)}$ of each CONV layer depends on the initial selected A and α values, being A the number of spectral bands (N in our case) and $\alpha = 50$.

III. Experiments

A. Hyperspectral Data Sets

Four well-known hyperspectral data sets have been considered in the experimental part of the work: Indian Pines (IP), University of Pavia (UP), Salinas Valley (SV), and Kennedy Space Center (KSC). Table II shows a brief summary of the considered HSI images, including the number of samples per class, as well as the available ground-truth information.

TABLE I

PROPOSED NETWORK TOPOLOGY. AVERAGE POOLING HAS A KERNEL OF 2×2 WITH STRIDE 2, AND FC LAYER HAS N_{class} NEURONS, BEING N_{class} THE NUMBER OF CLASSES OF EACH DATA SET

Module ID C/P_i	**Unit ID** $B_j^{(i)}$	**CONV ID** $C_l^{(j)}$	**Kernel size** $k_l^{(j)} \times k_l^{(j)}$	**Stride**
Input module				
C	–	–	3×3	1
Pyramidal modules				
P_1	$B_1^{(1)}$	$C_1^{(1)}$	1×1	1
		$C_2^{(1)}$	7×7	1
		$C_3^{(1)}$	1×1	1
	$B_2^{(1)}$	$C_1^{(2)}$	1×1	1
		$C_2^{(2)}$	7×7	1
		$C_3^{(1)}$	1×1	1
	$B_3^{(1)}$	$C_1^{(3)}$	1×1	1
		$C_2^{(3)}$	7×7	1
		$C_3^{(3)}$	1×1	1
P_2	$B_1^{(2)}$	$C_1^{(1)}$	1×1	1
		$C_2^{(1)}$	8×8	2
		$C_3^{(1)}$	1×1	1
	$B_2^{(2)}$	$C_1^{(2)}$	1×1	1
		$C_2^{(2)}$	7×7	1
		$C_3^{(1)}$	1×1	1
	$B_3^{(2)}$	$C_1^{(3)}$	1×1	1
		$C_2^{(3)}$	7×7	1
		$C_3^{(3)}$	1×1	1
P_3	$B_1^{(3)}$	$C_1^{(1)}$	1×1	1
		$C_2^{(1)}$	8×8	2
		$C_3^{(1)}$	1×1	1
	$B_2^{(3)}$	$C_1^{(2)}$	1×1	1
		$C_2^{(2)}$	7×7	1
		$C_3^{(1)}$	1×1	1
	$B_3^{(3)}$	$C_1^{(3)}$	1×1	1
		$C_2^{(3)}$	7×7	1
		$C_3^{(3)}$	1×1	1

Additionally, a more detailed description of each image is given as follows.

1) *IP:* The IP data set (Table II) was gathered in 1992 by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor [63] over an agricultural area in Northwestern Indiana. Specifically, it covers a set of agricultural fields with regular geometry and also irregular forest areas. The selected scene contains 145×145 pixels, with a total of 224 spectral bands in the wavelength range from 400 to 2500 nm, and spatial resolution of 20 meters per pixel (mpp). After removing four null bands and other 20 bands corrupted by the atmospheric water absorption effect, the remaining 200 bands have been considered for the experiments. Moreover, about half of the data (10 249 pixels from a total of 21 025) contains ground-truth information in the form of a single label from 16 different classes.
2) *UP:* The UP image (Table II) was acquired by the Reflective Optics System Imaging Spectrometer sensor [64] over the campus at the UP, Northern Italy. This data set mainly contains an urban environment with multiple solid structures (asphalt, gravel, metal sheets, bitumen, and bricks), natural objects (trees, meadows, and soil), and shadows. After discarding the noisy bands, the considered scene contains 103 spectral bands, with a size of 610×340 pixels in the spectral range from 0.43 to 0.86 μm and with spatial resolution of 1.3 mpp. About a 20% of the pixels (42 776 of 207 400) contain ground-truth information from nine different class labels.
3) *SV:* The SV scene (Table II) was collected by the 224-band AVIRIS sensor over the SV, CA, USA, and it is characterized by a spatial resolution of 3.7 mpp. The area covered comprises 512 lines by 217 samples. As in the case of the IP data set, we discard the 20 water absorption bands, i.e., [108–112], [154–167], and 224. This image was only available as at-sensor radiance data, and includes a total of 16 ground-truth classes, such as vegetables, bare soils, and vineyard fields.
4) *KSC:* The KSC data set (Table II) was collected by the AVIRIS instrument over the KSC in FL, USA, in 1996. Once noisy bands have been removed, the resulting image contains 176 bands with a 512×614 size, ranging from 400 to 2500 nm and with 20 mpp spatial resolution. A total of 5122 pixels labeled in 13 classes, representing different land-cover types, are considered for classification purposes.

B. Experimental Configuration

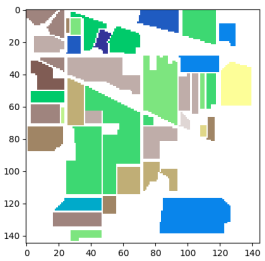
The proposed approach has been compared to a total of 10 different classification methods available in the literature: 1) SVM with radial basis function kernel [65]; 2) RF; 3) MLP; 4) extreme learning machine (ELM) [66]; 5) kernel ELM (KELM) [67]; 6) 1-D CNN; 7) 2-D CNN; 8) 3-D-CNN; 9) spectral–spatial ResNet (SSRN) [48]; and 10) deep fast CNN (DFCNN) [50]. All hyperparameters have been fixed in an optimal way for each method.

More specifically, the SVM, RF, MLP, ELM, KELM, and 1-D-CNN are spectral classifiers. 2-D-CNN is a spatial-based method, where PCA has been applied over the hyperspectral data in order to extract one principal component (i.e., it reduces the number of spectral bands N to 1), and 3-D-CNN, SSRN, DFCNN, together with the proposed approach are spectral–spatial techniques. Considering all these classification methods and the aforementioned data sets, we provide four different experiments to validate the performance of the proposed approach with respect to standard classifiers (experiment 1), considering different training data percentages (experiment 2), and drawing comparisons with two recent CNN-based spectral–spatial classifiers (experiments 3 and 4).

1) In our first experiment, the proposed network is compared with the standard SVM, RF, MLP, 2-D-CNN, and 3-D-CNN classification methods using a training set made up of 15% of the available labeled data for the

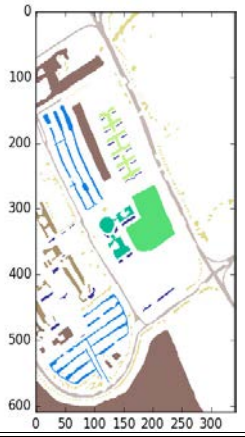
TABLE II
NUMBER OF SAMPLES OF THE IP, UP, AND SV HSI DATA SETS

INDIAN PINES (IP)



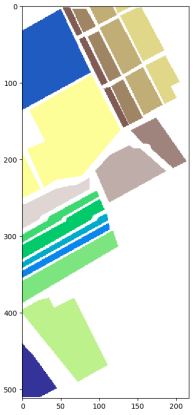
Color	Land-cover type	Samples
	Background	10776
	Alfalfa	46
	Corn-notill	1428
	Corn-min	830
	Corn	237
	Grass/Pasture	483
	Grass/Trees	730
	Grass/pasture-mowed	28
	Hay-windrowed	478
	Oats	20
	Soybeans-notill	972
	Soybeans-min	2455
	Soybean-clean	593
	Wheat	205
	Woods	1265
	Bldg-Grass-Tree-Drives	386
	Stone-steel towers	93
	Total samples	21025

UNIVERSITY OF PAVIA (UP)



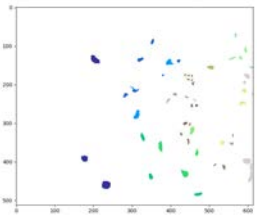
Color	Land-cover type	Samples
	Background	164624
	Asphalt	6631
	Meadows	18649
	Gravel	2099
	Trees	3064
	Painted metal sheets	1345
	Bare Soil	5029
	Bitumen	1330
	Self-Blocking Bricks	3682
	Shadows	947
	Total samples	207400

SALINAS (SV)



Color	Land-cover type	Samples
	Background	56975
	Brocoli-green-weeds-1	2009
	Brocoli-green-weeds-2	3726
	Fallow	1976
	Fallow-rough-plow	1394
	Fallow-smooth	2678
	Stubble	3959
	Celery	3579
	Grapes-untrained	11271
	Soil-vinyard-develop	6203
	Corn-senesced-green-weeds	3278
	Lettuce-romaine-4wk	1068
	Lettuce-romaine-5wk	1927
	Lettuce-romaine-6wk	916
	Lettuce-romaine-7wk	1070
	Vinyard-untrained	7268
	Vinyard-vertical-trellis	1807
	Total samples	111104

KENNEDY S.C. (KSC)



Color	Land-cover type	Samples
	Background	309157
	Scrub	761
	Willow-swamp	243
	CP-hammock	256
	Slash-pine	252
	Oak/Broadleaf	161
	Hardwood	229
	Swap	105
	Graminoid-marsh	431
	Spartina-marsh	520
	Cattail-marsh	404
	Salt-marsh	419
	Mud-flats	503
	Water	927
	Total samples	314368

IP, UP, and SV data sets. Additionally, the input spatial size is fixed to $N \times 11 \times 11$ for the 2-D-CNN, 3-D-CNN, and the proposed model, being N the number of spectral bands.

2) In our second experiment, we compare the classification accuracy of the proposed approach with regards to that obtained by spectral methods, in particular SVM, RF, MLP, ELM, KELM, and 1-D-CNN, by considering different training percentages over the IP and UP data sets, following the same configuration proposed in [7]. Specifically, we use 5%, 10%, 15%, 20%, and 25% training percentages and set the input patch size of the proposed approach to $N \times 7 \times 7$.
3) In our third experiment, the proposed approach is compared to the SSRN spectral–spatial classifier using four different spatial sizes, i.e., 5×5, 7×7, 9×9, 11×11, and the training configuration considered in [48]. That is, we consider 20% of the available labeled data for the IP and KSC data sets, and 10% of the available training data for the UP data set.
4) Finally, the fourth experiment compares the proposed approach with the DFCNN network using three different spatial sizes, 9×9, 15×15, and 19×19, and we use the training configuration considered in [50]. Specifically, the number of randomly selected training samples per labeled class is: 30, 150, 150, 100, 150, 150, 20, 150, 15, 150, 150, 150, 150, 150, 50, and 50 in the case of IP, and 548, 540, 392, 542, 256, 532, 375, 514, and 231 for UP.

In order to assess the results, three widely used quantitative metrics are used to evaluate the classification performance: overall accuracy (OA), average accuracy (AA), and Kappa coefficient. Regarding the hardware environment in which we have run the experiments, it is composed of the Sixth Generation Intel Core i7-6700K processor with 8M of Cache and up to 4.20 GHz (4 cores/8 way multi-task processing), 40 GB of DDR4 RAM with a serial speed of 2400 MHz, a GPU NVIDIA GeForce GTX 1080 with 8 GB GDDR5X of video memory and 10 Gb/s of memory frequency, a Toshiba DT01ACA HDD with 7200RPM and 2TB of capacity, and an ASUS Z170 progaming motherboard. Additionally, the used software environment is composed of Ubuntu 16.04.4 x64 as operating system, CUDA 8 and cuDNN 5.1.5, Python 2.7 as programming languages.

C. Experimental Results

1) Experiment 1: Tables III–V present the classification results for IP, UP, and SV data sets, corresponding to our first experiment. Specifically, the first column of each table indicates the corresponding data set class; the next five columns show the results obtained by SVM, RF, MLP, 2-D-CNN, and 3-D-CNN classifiers, and the last column contains the result

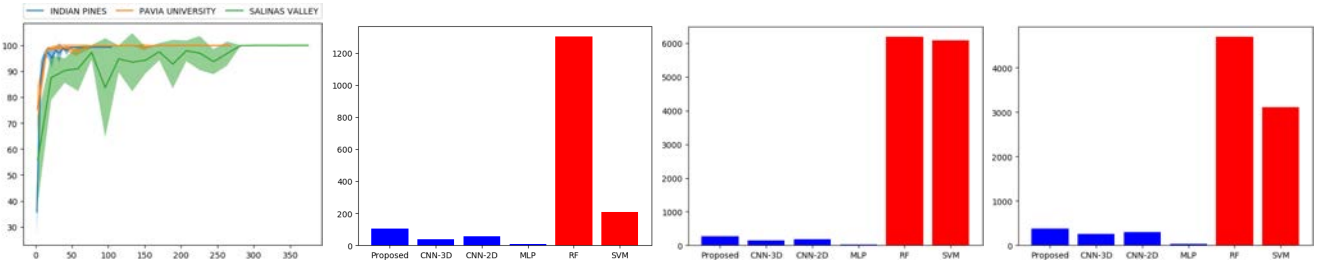


Fig. 4. (From left to right) (a) Achieved accuracy (vertical axis) versus employed computing time in seconds (horizontal axis) for the IP, Pavia University (PU), and SV data sets. Total execution times of each compared algorithm for (b) IP, (c) PU, and (d) SV data sets. Blue and red bars: performance of the GPU and CPU implementations, respectively.

TABLE III

CLASSIFICATION RESULTS FOR THE IP DATA SET USING 15% OF THE LABELED DATA FOR TRAINING AND 11 × 11 INPUT SPATIAL SIZE

Class	SVM	RF	MLP	2D-CNN	3D-CNN	Proposed
1	68.04 ±6.95	33.04 ±7.45	62.39 ±13.96	65.87 ±10.34	89.13 ±7.28	**93.04** ±7.58
2	83.55 ±1.31	66.68 ±1.67	83.84 ±2.46	81.04 ±3.28	98.33 ±0.71	**99.13** ±0.56
3	73.82 ±1.44	56.20 ±2.41	76.37 ±5.03	79.07 ±6.75	98.05 ±1.40	**99.54** ±0.36
4	71.98 ±3.86	41.10 ±2.50	68.35 ±6.12	82.70 ±8.34	98.23 ±0.62	**99.92** ±0.17
5	94.29 ±0.97	87.12 ±1.73	90.87 ±2.09	69.25 ±10.58	97.56 ±2.84	**99.83** ±0.24
6	97.32 ±0.97	95.32 ±1.79	96.95 ±1.10	88.29 ±5.51	98.93 ±1.14	**99.89** ±0.13
7	88.21 ±5.06	32.86 ±12.66	78.21 ±10.28	67.86 ±25.65	83.57 ±19.51	**99.29** ±1.43
8	98.16 ±0.75	98.49 ±0.81	98.08 ±0.90	96.26 ±1.60	99.41 ±0.61	**100.00** ±0.00
9	52.00 ±8.43	13.00 ±3.32	72.00 ±8.12	67.00 ±27.68	65.00 ±21.68	**99.00** ±2.00
10	79.49 ±2.76	69.95 ±4.31	82.17 ±5.41	68.82 ±9.80	97.22 ±0.31	**98.48** ±0.88
11	86.83 ±1.05	90.66 ±1.18	83.66 ±2.85	86.55 ±3.14	98.12 ±2.16	**99.58** ±0.22
12	83.41 ±2.26	55.43 ±4.80	75.89 ±3.33	73.41 ±6.07	93.09 ±5.85	**98.55** ±0.64
13	97.41 ±2.99	93.32 ±2.04	98.68 ±0.54	94.54 ±4.80	99.80 ±0.39	**99.51** ±0.98
14	96.14 ±0.97	96.45 ±0.76	96.17 ±1.02	96.24 ±2.33	99.43 ±0.33	**99.81** ±0.19
15	67.31 ±3.05	50.44 ±2.44	67.80 ±3.56	85.39 ±7.71	96.58 ±2.81	**99.53** ±0.30
16	92.47 ±4.14	85.27 ±3.37	88.71 ±2.77	92.90 ±3.97	93.12 ±3.82	**98.49** ±1.46
OA (%)	86.24 ±0.38	78.55 ±0.68	85.27 ±0.47	83.59 ±0.88	97.81 ±0.56	**99.40** ±0.08
AA (%)	83.15 ±1.10	66.58 ±0.93	82.51 ±1.04	80.95 ±1.55	94.10 ±2.00	**98.98** ±0.49
Kappa	84.27 ±0.45	75.20 ±0.81	83.20 ±0.53	81.23 ±1.04	97.50 ±0.64	**99.31** ±0.10
Time(s)	208.98 ±1.70	1,301.68 ±45.94	**7.31** ±0.15	56.45 ±0.19	39.62 ±0.67	103.21 ±0.47

TABLE IV

CLASSIFICATION RESULTS FOR THE UP DATA SET USING 15% OF THE LABELED DATA FOR TRAINING AND 11 × 11 INPUT SPATIAL SIZE

Class	SVM	RF	MLP	2D-CNN	3D-CNN	Proposed
1	95.36 ±0.30	93.52 ±0.45	94.17 ±1.73	93.43 ±2.70	99.16 ±0.25	**99.91** ±0.07
2	98.25 ±0.16	98.29 ±0.18	98.06 ±0.50	97.59 ±0.88	99.77 ±0.17	**99.99** ±0.01
3	82.93 ±0.91	75.56 ±1.86	79.27 ±7.04	89.96 ±3.30	96.95 ±1.78	**99.77** ±0.14
4	95.93 ±0.70	91.68 ±0.63	94.61 ±2.58	94.16 ±3.24	98.80 ±0.69	**99.80** ±0.09
5	99.46 ±0.36	98.88 ±0.49	99.63 ±0.27	97.97 ±2.69	99.90 ±0.17	**100.00** ±0.00
6	91.76 ±0.60	74.54 ±0.97	93.60 ±1.70	89.62 ±4.10	99.88 ±0.12	**100.00** ±0.00
7	88.59 ±0.65	81.01 ±1.74	88.53 ±3.47	80.20 ±4.82	96.54 ±1.41	**99.66** ±0.49
8	90.14 ±0.54	90.70 ±0.75	89.59 ±4.56	96.05 ±1.88	98.56 ±0.78	**99.92** ±0.09
9	99.97 ±0.05	99.75 ±0.26	99.63 ±0.28	99.48 ±0.27	99.79 ±0.19	**100.00** ±0.00
OA (%)	95.20 ±0.13	92.03 ±0.21	94.82 ±0.26	94.77 ±0.72	99.28 ±0.25	**99.94** ±0.01
AA (%)	93.60 ±0.14	89.33 ±0.33	93.01 ±0.60	93.16 ±1.23	98.81 ±0.33	**99.89** ±0.05
Kappa	93.63 ±0.17	89.30 ±0.28	93.13 ±0.34	93.05 ±0.97	99.04 ±0.32	**99.92** ±0.02
Time(s)	6,084.92 ±55.64	6,188.75 ±35.16	**29.10** ±0.92	172.29 ±0.71	140.09 ±1.63	269.19 ±0.66

TABLE V

CLASSIFICATION RESULTS FOR THE SV DATA SET USING 15% OF THE LABELED DATA FOR TRAINING AND 11 × 11 INPUT SPATIAL SIZE

Class	SVM	RF	MLP	2D-CNN	3D-CNN	Proposed
1	99.68 ±0.21	99.61 ±0.12	99.72 ±0.42	87.99 ±17.62	100.00 ±0.00	**100.00** ±0.00
2	99.87 ±0.12	99.86 ±0.07	99.88 ±0.15	99.75 ±0.23	99.99 ±0.01	**100.00** ±0.00
3	99.74 ±0.11	99.22 ±0.51	99.43 ±0.44	81.40 ±10.85	99.94 ±0.07	**100.00** ±0.00
4	99.48 ±0.18	99.28 ±0.44	99.61 ±0.27	95.11 ±5.51	99.83 ±0.23	**100.00** ±0.00
5	99.24 ±0.31	98.46 ±0.21	99.25 ±0.48	64.31 ±12.09	99.90 ±0.09	**100.00** ±0.00
6	99.92 ±0.06	99.80 ±0.09	99.92 ±0.07	99.60 ±0.11	100.00 ±0.00	**100.00** ±0.00
7	99.70 ±0.15	99.58 ±0.09	99.82 ±0.12	98.01 ±4.54	99.90 ±0.15	**99.99** ±0.01
8	90.87 ±0.39	84.41 ±1.34	85.41 ±8.00	91.89 ±2.44	90.67 ±6.83	**99.92** ±0.07
9	99.94 ±0.02	99.07 ±0.17	99.86 ±0.07	98.02 ±1.56	99.99 ±0.01	**100.00** ±0.00
10	98.26 ±0.27	93.40 ±0.58	97.15 ±0.77	97.05 ±0.67	99.27 ±0.43	**99.91** ±0.09
11	99.61 ±0.34	94.79 ±0.59	97.42 ±2.29	94.58 ±3.59	99.48 ±0.73	**99.96** ±0.07
12	99.93 ±0.05	99.08 ±0.29	99.80 ±0.14	92.67 ±5.75	99.76 ±0.38	**100.00** ±0.00
13	99.07 ±0.72	98.23 ±0.69	99.40 ±0.28	98.10 ±0.76	99.63 ±0.58	**99.98** ±0.04
14	98.08 ±1.00	92.81 ±1.04	97.58 ±0.94	95.25 ±5.74	99.94 ±0.11	**100.00** ±0.00
15	72.83 ±0.78	63.32 ±1.82	80.27 ±8.41	87.36 ±3.87	96.18 ±1.52	**99.95** ±0.04
16	99.45 ±0.25	98.17 ±0.36	98.97 ±0.38	93.72 ±1.66	99.39 ±0.42	**99.93** ±0.06
OA (%)	94.15 ±0.10	90.76 ±0.24	93.87 ±0.70	92.31 ±1.62	97.44 ±1.28	**99.97** ±0.02
AA (%)	97.23 ±0.11	94.94 ±0.12	97.09 ±0.33	92.18 ±2.72	98.99 ±0.40	**99.98** ±0.01
Kappa	93.48 ±0.11	89.70 ±0.26	93.18 ±0.77	91.43 ±1.81	97.15 ±1.42	**99.96** ±0.02
Time(s)	3,110.30 ±29.20	4,694.29 ±158.39	**36.42** ±0.11	296.62 ±3.52	260.41 ±6.09	372.51 ±1.46

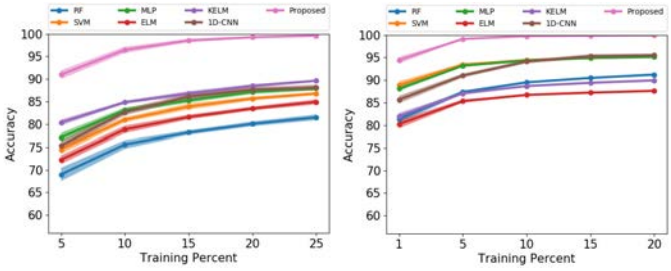


Fig. 5. OA (%) for SVM, RF, MLP, ELM, KELM, 1-D-CNN, and the proposed approach when considering different training percentages in (Left) IP and (Right) UP data sets.

of the proposed approach. Additionally, the OA, AA, Kappa coefficient, and computational time in seconds are provided in the last four rows. It should be mentioned that MLP, 2-D-CNN, 3-D-CNN, and the proposed approach take advantage of the GPU to accelerate the corresponding procedures. Also, in Fig. 4, we can observe the latency and execution time results of the proposed method.

2) Experiment 2: Fig. 5 shows the results obtained in our second experiment, where different training percentages are tested using IP and UP data sets. In particular, SVM, RF, MLP, ELM, KELM, 1-D-CNN, and the proposed method are tested considering 5%, 10%, 15%, 20%, and 25% of the labeled data for training. It should be also mentioned that Fig. 5 (left) contains the results for the IP data set, and Fig. 5 (right) contains the results for the UP data set.

3) Experiment 3: In addition to the global analysis conducted in the first two experiments, we also conduct two additional experiments to compare the proposed approach and two recent state-of-the-art spectral–spatial classification networks. In this experiment, we compare our approach with SSRN, which has been presented in [48]. Table VI provides the classification results obtained by SSRN and the proposed method. Specifically, the first column contains the considered spatial input size, and the next three columns show the OA for IP, KSC, and UP data sets, respectively. Note that we use the same training configuration used in [48], that is, 20% of the available labeled data for IP and KSC, and 10% of the available labeled data for UP.

4) Experiment 4: Table VII shows the results of the comparison between the DFCNN method (presented in [50]) and the proposed approach. In particular, three different spatial sizes are considered for the IP and UP data sets. Note that additional spatial configurations are not reported because the proposed approach already provides an optimal result.

To conclude this section, Figs. 6–8 complete the experimental comparison by providing some of the classification maps provided by the methods tested in the first experiment for the

TABLE VI
OA (%) ACHIEVED BY THE SSRN METHOD [48] AND THE PROPOSED APPROACH WHEN CONSIDERING DIFFERENT INPUT SPATIAL SIZES

	Indian Pines (IP)		Kennedy Space Center (KSC)		University of Pavia (UP)	
Spatial Size	**SSRN**	**Proposed**	**SSRN**	**Proposed**	**SSRN**	**Proposed**
5×5	92.83 $\pm$0.66	**98.80** $\pm$0.10	96.99 $\pm$0.55	**98.81** $\pm$0.07	98.72 $\pm$0.17	**99.52** $\pm$0.05
7×7	97.81 $\pm$0.34	**99.26** $\pm$0.06	99.01 $\pm$0.31	**99.51** $\pm$0.08	99.54 $\pm$0.11	**99.81** $\pm$0.09
9×9	98.68 $\pm$0.29	**99.64** $\pm$0.08	99.51 $\pm$0.25	**99.60** $\pm$0.05	99.73 $\pm$0.15	**99.87** $\pm$0.03
11×11	98.70 $\pm$0.21	**99.82** $\pm$0.07	99.57 $\pm$0.54	**99.79** $\pm$0.11	99.79 $\pm$0.08	**99.92** $\pm$0.02

TABLE VII
OA (%) ACHIEVED BY THE DFCNN METHOD [50] AND THE PROPOSED APPROACH WHEN CONSIDERING DIFFERENT INPUT SPATIAL SIZES

	Indian Pines (IP)		University of Pavia (UP)	
Spatial Size	**DFCNN**	**Proposed**	**DFCNN**	**Proposed**
9×9	93.94	**98.87** $\pm$0.19	-	-
15×15	-	-	98.87	**99.93** $\pm$0.02
19×19	96.29	**99.45** $\pm$0.14	-	-

IP, UP, and SV data sets. As it can be observed, the proposed method provides spatially consistent classification outputs with well-delineated object borders and very few classification interferers.

D. Discussion

According to the reported results, one of the first noticeable points is the high classification accuracy that the proposed approach is able to provide in the different considered scenarios. That is, the proposed network architecture achieves a consistent precision improvement when considering not only the standard spectral classification methods SVM, RF, MPL, ELM, KELM, and 1-D-CNN but also the spatial approach 2-D-CNN and, most importantly, the spectral–spatial methods 3-D-CNN, SSRN, and DFCNN.

In Tables III–V, it is possible to observe that the proposed approach provides the best average results as well as the highest accuracy values for each individual class in the IP, UP, and SV data sets. In particular, the average improvement over the second best classifier, the spectral–spatial 3-D-CNN, is $+1.59$, $+2.31$, and $+1.83$ for OA, AA, and Kappa metrics, respectively. Additionally, the network presented in this paper also shows a remarkable performance improvement when considering different percentages of training data. According to Fig. 5, the proposed approach obtains the highest accuracy result for all the tested training data percentages in IP and UP data sets. Besides, the proposed approach also tends to converge faster to the maximum accuracy value than the rest of the tested methods.

These results are also consistent with the corresponding classification maps shown in Figs. 6–8. On the one hand, spectral methods, such as SVM or MLP, tend to generate rather noisy classification maps because they do not take into account the spatial component when providing a pixel prediction. On the other hand, spatial classifiers, i.e., 2-D-CNN, are prone to alter some object shapes depending on the considered input spatial size. Precisely, spectral–spatial classifiers work for overcoming both the limitations. As we can see, the proposed approach certainly provides the classification results that are more similar with regards to the corresponding ground-truth classification maps for IP, UP, and SV data sets. In addition, it is possible to observe that the proposed method also reaches a higher performance. That is, class boundaries are better defined and background pixels are better classified according to the actual ground-truth image content. For instance, the classification map depicted in Fig. 7(h) shows that the proposed approach provides a clean classification result for the *self-blocking bricks* class in the UP scene, while noise and outliers are also significantly reduced with respect to the rest of the methods.

From this initial comparison, we can note that spatial–spectral classification algorithms are those which provide the best performance over all the considered data sets. More specifically, the RF spectral classifier obtains the lowest average OA in the conducted experiments (87.11%), followed by the spatial 2-D-CNN (90.22%) and the spectral MLP (91.32%) methods. Besides, the spectral SVM approach shows, on average, a slightly better performance (91.86%). Nonetheless, the performances provided by the spectral–spatial methods, i.e., the 3-D-CNN network (98.17%) and the proposed approach (99.77%), are significantly higher. Precisely, this is the reason why we conduct a more detailed performance comparison between the proposed approach and two recent spectral–spatial methods, SSRN, and DFCNN.

Regarding the SSRN performance comparison, Table VI shows some important points that deserve to be mentioned. Although both methods (SSRN and the proposed one) improve the classification accuracy when considering a higher input spatial size, the proposed approach provides a substantial precision gain, especially with smaller input spatial sizes. That is, the proposed approach pyramidal architecture provides the advantage of extracting more feature maps as the network residual units are deeper; therefore, it is able to better exploit the information contained within an input HSI cube in order to learn more robust spectral–spatial representations. As a result, the proposed method provides a more accurate (as well as robust) classification result than the SSRN. In other words, the proposed method consistently achieves higher accuracy results and lower standard deviation values than the SSRN, which means that the class uncertainty is significantly reduced, no matter the considered spatial size. Note that SSRN obtains some standard deviation values relatively large considering the high OA. For instance, it is the case of the KSC data set when considering a 11×11 spatial size. As we can see, SSRN obtains a $99.57 \pm 0.54\%$ OA, whereas the proposed approach result, $99.79 \pm 0.11\%$, achieves even a higher accuracy with a

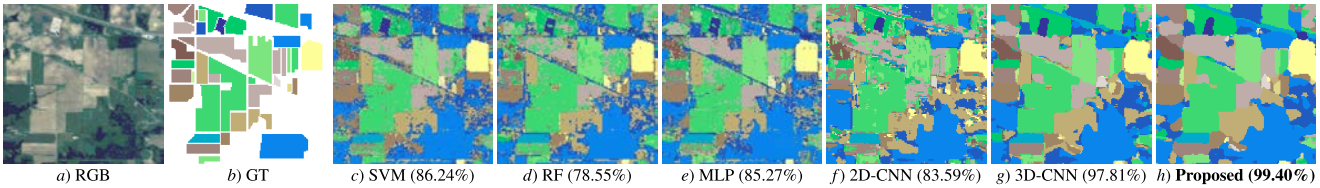


Fig. 6. Classification maps for the IP data set. (a) Simulated RGB composition of the scene. (b) Ground-truth classification map. (c)–(h) Classification maps corresponding to Table III. Note that the overall classification accuracies are shown in brackets and the best result is highlighted in bold font.

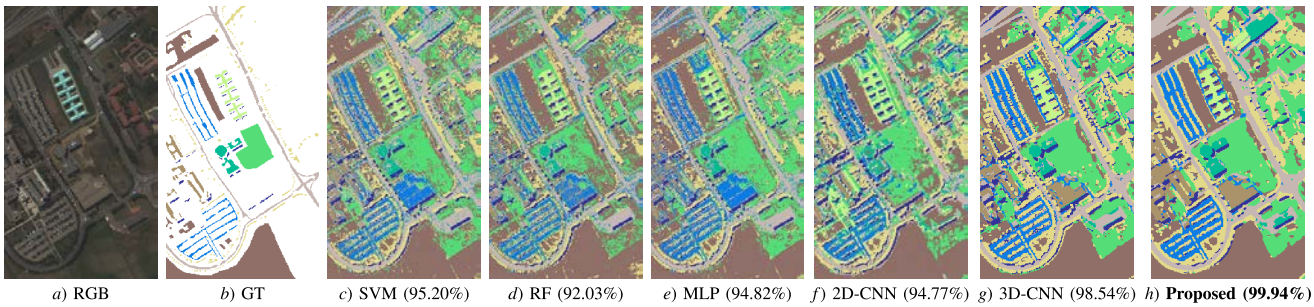


Fig. 7. Classification maps for the UP data set. (a) Simulated RGB composition of the scene. (b) Ground-truth classification map. (c)–(h) classification maps corresponding to Table IV. Note that the overall classification accuracies are shown in brackets and the best result is highlighted in bold font.

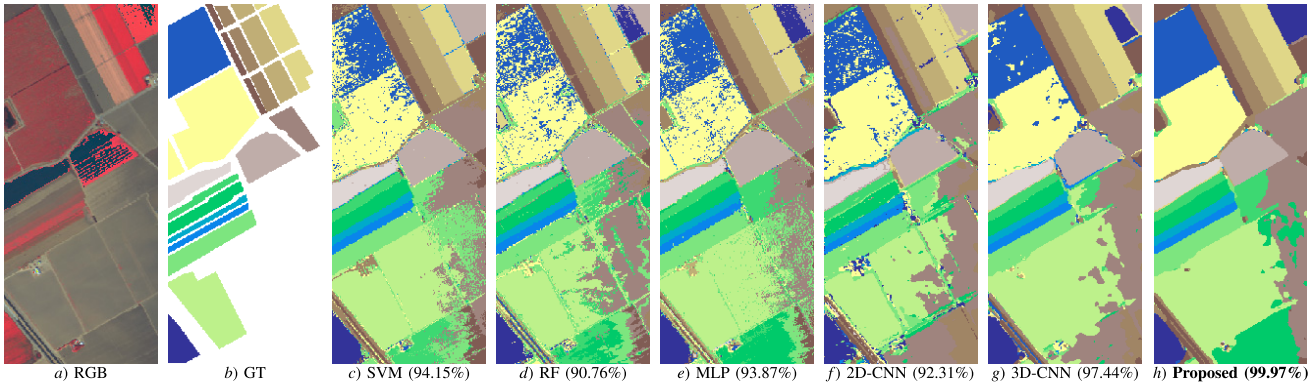


Fig. 8. Classification maps for the SV data set. (a) Simulated RGB composition of the scene. (b) Ground-truth classification map. (c)–(h) classification maps corresponding to Table V. Note that the overall classification accuracies are shown in brackets and the best result is highlighted in bold font.

five times lower standard deviation. In general, the proposed approach exhibits a better classification performance than SSRN for IP, KSC, and UP data sets, because it is able to obtain higher accuracy results with lower standard deviation values, which also shows that the proposal is robust in the presence of variability and noise.

A similar trend can be also observed in the reported DFCNN comparison (Table VII). In particular, the proposed approach obtains better OA than DFCNN for IP and UP data sets when considering 9×9, 15×15, and 19×19 spatial sizes, respectively. Taking all these observations into account, it is possible to state that the proposed approach provides a more accurate and robust classification result than all of the other tested methods. Even though the spectral–spatial classifiers 3-D-CNN, SSRN, and DFCNN have shown to obtain relatively high classification accuracies, the proposed architecture provides a more effective scheme to reduce the uncertainty when uncovering spectral–spatial features. That is, increasing the feature map dimension at all CONV layers, grouped in pyramidal residual blocks, allows the proposed approach to involve more locations as the network depth increases while balancing the workload among all units and preserving the time complexity per layer. As a result, the diversity of high-level spectral–spatial attributes can be gradually increased across layers to enhance the capability of the network to manage remotely sensed HSI data.

The obtained results also demonstrate that the proposed technique provides a remarkable quantitative improvement, which indicates that the presented spectral–spatial architecture is able to generate more distinctive features to effectively classify remotely sensed HSIs, achieving the best accuracy performance for all the conducted experiments (see Tables III–VII) and the most robust behavior when dealing with different input spatial sizes (see Tables VI and VII). The effectiveness of the proposed network (when compared with regular CNN models) lies in its architecture, which progressively increases the feature map dimension at all residual units, allowing the proposed approach to involve more 3-D volume locations as

the network depth increases. This fact eventually promotes uncovering a larger variety of high-level spectral–spatial features, balancing the workload among units to facilitate the network training process, and also allowing the model to reduce the declining-accuracy phenomenon when considering significantly deep networks. Based on the reported results with different HSI data sets, multiple training percentages, and several input spatial sizes, we can conclude that the proposed technique is able to better exploit the spectral–spatial information contained in an HSI data cube, thus maintaining a good quantitative performance even with small kernel spatial sizes.

According to the computational times reported in Tables III–V, it is also possible to highlight some important aspects among the tested methods. On average, SVM and RF classifiers are the most time-consuming methods, followed by the proposed approach, 2-D-CNN and 3-D-CNN. Finally, MLP has shown to be the most efficient technique in computational terms. Even though the adopted SVM and RF implementations do not take advantage of GPU acceleration, their corresponding optimal parameter search tasks are computationally demanding processes which highly affect the overall computational time. In the case of the tested neural network-based methods, the pyramidal residual blocks of the proposed approach logically require a larger amount of computational power than simpler architectures. Specifically, the proposed approach computational time is, on average, a 25% and 43% higher than the corresponding 2-D-CNN and 3-D-CNN costs. Despite the fact that the proposed approach obtains a higher computational time than MLP, 2-D-CNN, and 3-D-CNN networks, the resulting cost increase is moderate considering the high number of operations required by the proposed model when compared to simpler architectures. That is, the proposed network is able to find spectral–spatial relationships useful to obtain a relatively more effective model convergence as well as a remarkable classification improvement. Looking at Fig. 4, we can observe in Fig. 4(a) that the proposed approach takes relatively little time to reach a good accuracy (around 25 s), while in Fig. 4(b)–(d), we show the total execution time of each compared algorithm, being SVM and RF the two slowest methods. This is mainly due to the parameter searching process (which is performed in the CPU), which has a strong influence in the computation times. In contrast, the MLP is the fastest GPU-implemented classifier, while the proposed technique is one of the slowest GPU-implemented methods due to its more complex architecture, followed by the spatial CNN. Finally, it is also important to highlight that the proposed approach generally exhibits a lower computational time than SSRN according to the results reported in [48].

IV. Conclusion

This paper presents a novel CNN-based deep network architecture specifically designed to manage large hyperspectral data cubes. In particular, the proposed new hyperspectral pyramidal ResNet pursues to improve the straightforward residual model formulation by better exploiting the potential of the information available on each unit. The proposed architecture gradually increases the feature map dimension step by step at each pyramidal bottleneck residual blocks, composed of three convolutional layers, as a pyramid, in order to involve more feature map locations as the network depth increases, while balancing the workload among all units and preserving the time complexity per layer. The experimental part of the work, conducted over four well-known hyperspectral data sets and using 10 different classification methods, reveal that the new hyperspectral pyramidal residual model is able to provide a competitive advantage over state-of-the-art classification methods.

One of the main conclusions that arises from this paper is the relevance of using spectral–spatial information when classifying the hyperspectral data. In this regard, the newly proposed approach is able to uncover highly descriptive spectral–spatial classification features throughout the implemented network convolutional filters. That is, our adopted strategy for gradually increasing the feature map dimension at all residual-based units allows us to consider a higher variety of spectral–spatial attributes as the network depth increases, because more image locations can be simultaneously considered. Eventually, this fact leads to classification improvements by means of the combined spectral–spatial features, which help to better discern among classes in multiple HSI data sets and experimental settings. Although other recent approaches, such as SSRN and DFCNN, exhibit very good classification performance, the new proposed hyperspectral pyramidal residual model is able to outperform their results and also to provide a more robust behavior when considering different input spatial sizes. Another important point is related to the amount of data used for training purposes. Although deep learning methods usually require a significant amount of the labeled data, the proposed approach has shown to provide consistent performance improvements with respect to other state-of-the-art models using different percentages of training data.

Despite the good results provided by the proposed approach, there are several unresolved issues that may present challenges over time. In particular, our future work will be aimed at the following directions: 1) reducing the computational complexity of the proposed HSI classification network by developing new methods to optimize the model parameters; 2) developing more efficient parallel implementations of the proposed model; and 3) integrating advanced data augmentation and active learning schemes into the proposed classification framework.

Acknowledgment

The authors would like to thank the Editors and the Anonymous Reviewers for their constructive comments and suggestions, which greatly helped them to improve the technical quality and presentation of this paper.

References

[1] D. Landgrebe, "Hyperspectral image data analysis," *IEEE Signal Process. Mag.*, vol. 19, no. 1, pp. 17–28, Jan. 2002.

[2] J. M. Bioucas-Dias, A. Plaza, G. Camps-Valls, P. Scheunders, N. M. Nasrabadi, and J. Chanussot, "Hyperspectral remote sensing data analysis and future challenges," *IEEE Geosci. Remote Sens. Mag.*, vol. 1, no. 2, pp. 6–36, Jun. 2013.

[3] X. Zhang, Y. Sun, K. Shang, L. Zhang, and S. Wang, "Crop classification based on feature band set construction and object-oriented approach using hyperspectral images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 9, pp. 4117–4128, Sep. 2016.

[4] B. Uzkent, A. Rangnekar, and M. J. Hoffman, "Aerial vehicle tracking by adaptive fusion of hyperspectral likelihood maps," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 233–242.

[5] G. A. Carter *et al.*, "Remote sensing and mapping of tamarisk along the Colorado River, USA: A comparative use of summer-acquired hyperion, thematic mapper and quickbird data," *Remote Sens.*, vol. 1, no. 3, pp. 318–329, 2009.

[6] M. Fauvel, Y. Tarabalka, J. A. Benediktsson, J. Chanussot, and J. C. Tilton, "Advances in spectral–spatial classification of hyperspectral images," *Proc. IEEE*, vol. 101, no. 3, pp. 652–675, Mar. 2013.

[7] P. Ghamisi, J. Plaza, Y. Chen, J. Li, and A. J. Plaza, "Advanced spectral classifiers for hyperspectral images: A review," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 1, pp. 8–32, Mar. 2017.

[8] J. Haut, M. Paoletti, J. Plaza, and A. Plaza, "Cloud implementation of the K-means algorithm for hyperspectral image analysis," *J. Supercomput.*, vol. 73, no. 1, pp. 514–529, 2017.

[9] A. Marinoni and P. Gamba, "Unsupervised data driven feature extraction by means of mutual information maximization," *IEEE Trans. Comput. Imag.*, vol. 3, no. 2, pp. 243–253, Jun. 2017.

[10] A. Marinoni, G. C. Iannelli, and P. Gamba, "An information theory-based scheme for efficient classification of remote sensing data," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 10, pp. 5864–5876, Oct. 2017.

[11] E. G. Njoku, *Encyclopedia of Remote Sensing*. New York, NY, USA: Springer-Verlag, 2014.

[12] G. Camps-Valls, D. Tuia, L. Bruzzone, and J. A. Benediktsson, "Advances in hyperspectral image classification: Earth monitoring with statistical learning methods," *IEEE Signal Process. Mag.*, vol. 31, no. 1, pp. 45–54, Jan. 2014.

[13] G. Camps-Valls and L. Bruzzone, "Kernel-based methods for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 6, pp. 1351–1362, Jun. 2004.

[14] Y. Li, Z. Wu, J. Wei, A. Plaza, J. Li, and Z. Wei, "Fast principal component analysis for hyperspectral imaging based on cloud computing," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2015, pp. 513–516.

[15] J. Haut, M. Paoletti, A. Paz-Gallardo, J. Plaza, and A. Plaza, "Cloud implementation of logistic regression for hyperspectral image classification," in *Proc. 17th Int. Conf. Comput. Math. Methods Sci. Eng. (CMMSE)*, vol. 3, Jul. 2017, pp. 1063–2321.

[16] Y. Bazi and F. Melgani, "Gaussian process approach to remote sensing image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 1, pp. 186–197, Jan. 2010.

[17] J. Ham, Y. Chen, M. M. Crawford, and J. Ghosh, "Investigation of the random forest framework for classification of hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 3, pp. 492–501, Mar. 2005.

[18] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2094–2107, Jun. 2014.

[19] D. L. Donoho, "High-dimensional data analysis: The curses and blessings of dimensionality," *AMS Math Challenges Lecture*, vol. 1, p. 32, Aug. 2000.

[20] B. Rasti, M. O. Ulfarsson, and J. R. Sveinsson, "Hyperspectral feature extraction using total variation component analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 12, pp. 6976–6985, Dec. 2016.

[21] R. Hang *et al.*, "Robust matrix discriminative analysis for feature extraction from hyperspectral images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 5, pp. 2002–2011, May 2017.

[22] W. Sun, G. Yang, B. Du, L. Zhang, and L. Zhang, "A sparse and low-rank near-isometric linear embedding method for feature extraction in hyperspectral imagery classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 4032–4046, Jul. 2017.

[23] A. Martínez-Usó, F. Pla, J. M. Sotoca, and P. García-Sevilla, "Clustering-based hyperspectral band selection using information measures," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 12, pp. 4158–4171, Dec. 2007.

[24] S. A. Robila, "Band reduction for hyperspectral imagery processing," *Proc. SPIE, Comput. Imag. VIII*, vol. 7533, p. 75330W, Jan. 2010.

[25] X. Xu, Z. Shi, and B. Pan, "A new unsupervised hyperspectral band selection method based on multiobjective optimization," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 11, pp. 2112–2116, Nov. 2017.

[26] S. Feng, Y. Itoh, M. Parente, and M. F. Duarte, "Hyperspectral band selection from statistical wavelet models," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 4, pp. 2111–2123, Apr. 2017.

[27] J. Shen, X. Cao, Y. Li, and D. Xu, "Feature adaptation and augmentation for cross-scene hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 4, pp. 622–626, Apr. 2018.

[28] J. Li, J. M. Bioucas-Dias, and A. Plaza, "Spectral–spatial classification of hyperspectral data using loopy belief propagation and active learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 2, pp. 844–856, Feb. 2013.

[29] Z. Zhang and M. M. Crawford, "A batch-mode regularized multimetric active learning framework for classification of hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 11, pp. 6594–6609, Nov. 2017.

[30] C. Liu, L. He, Z. Li, and J. Li, "Feature-driven active learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 1, pp. 341–354, Jan. 2018.

[31] J. M. Haut, M. E. Paoletti, J. Plaza, J. Li, and A. Plaza, "Active learning with convolutional neural networks for hyperspectral image classification using a new Bayesian approach," *IEEE Trans. Geosci. Remote Sens.*, pp. 1–22, 2018, doi: 10.1109/TGRS.2018.2838665.

[32] L. He, J. Li, C. Liu, and S. Li, "Recent advances on spectral–spatial hyperspectral image classification: An overview and new guidelines," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 3, pp. 1579–1597, Mar. 2018.

[33] M. Khodadadzadeh, J. Li, A. Plaza, H. Ghassemian, J. M. Bioucas-Dias, and X. Li, "Spectral–spatial classification of hyperspectral data using local and global probabilities for mixed pixel characterization," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 10, pp. 6298–6314, Oct. 2014.

[34] W. Liao, M. Dalla Mura, J. Chanussot, and A. Pižurica, "Fusion of spectral and spatial information for classification of hyperspectral remote-sensed imagery by local graph," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 2, pp. 583–594, Feb. 2016.

[35] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[36] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[37] C. Zhao, X. Wan, G. Zhao, B. Cui, W. Liu, and B. Qi, "Spectral–spatial classification of hyperspectral imagery based on stacked sparse autoencoder and random forest," *Eur. J. Remote Sens.*, vol. 50, no. 1, pp. 47–63, 2017.

[38] T. Li, J. Zhang, and Y. Zhang, "Classification of hyperspectral image based on deep belief networks," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2014, pp. 5132–5136.

[39] P. Zhong, Z. Gong, S. Li, and C.-B. Schönlieb, "Learning to diversify deep belief networks for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 55, no. 6, pp. 3516–3530, Jun. 2017.

[40] X. Chen, S. Xiang, C.-L. Liu, and C.-H. Pan, "Vehicle detection in satellite images by hybrid deep convolutional neural networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 10, pp. 1797–1801, Oct. 2014.

[41] X. Ma, H. Wang, and J. Geng, "Spectral–spatial classification of hyperspectral image based on deep auto-encoder," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 9, pp. 4073–4085, Feb. 2016.

[42] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, "Deep feature extraction and classification of hyperspectral images based on convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 10, pp. 6232–6251, Oct. 2016.

[43] S. Mei, J. Ji, Q. Bi, J. Hou, Q. Du, and W. Li, "Integrating spectral and spatial information into deep convolutional neural networks for hyperspectral classification," in *Proc. Int. Geosci. Remote Sens. Symp.*, 2016, pp. 5067–5070.

[44] W. Li, G. Wu, F. Zhang, and Q. Du, "Hyperspectral image classification using deep pixel-pair features," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 2, pp. 844–853, Feb. 2017.

[45] J. Yang, Y. Zhao, J. C. W. Chan, and C. Yi, "Hyperspectral image classification using two-channel deep convolutional neural network," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2016, pp. 5079–5082.

[46] H. Zhang, Y. Li, Y. Zhang, and Q. Shen, "Spectral–spatial classification of hyperspectral imagery using a dual-channel convolutional neural network," *Remote Sens. Lett.*, vol. 8, no. 5, pp. 438–447, Jan. 2017.

[47] W. Shao and S. Du, "Spectral–spatial feature extraction for hyperspectral image classification: A dimension reduction and deep learning approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 8, pp. 4544–4554, Oct. 2016.

[48] Z. Zhong, J. Li, Z. Luo, and M. Chapman, "Spectral–spatial residual network for hyperspectral image classification: A 3-D deep learning framework," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 2, pp. 847–858, Feb. 2018.
[49] Y. Li, H. Zhang, and Q. Shen, "Spectral–spatial classification of hyperspectral imagery with 3D convolutional neural network," *Remote Sens.*, vol. 9, no. 1, p. 67, Jan. 2017. [Online]. Available: http://www.mdpi.com/2072-4292/9/1/67
[50] M. E. Paoletti, J. M. Haut, J. Plaza, and A. Plaza, "A new deep convolutional neural network for fast hyperspectral image classification," *ISPRS J. Photogramm. Remote Sens.*, to be published, doi: 10.1016/j.isprsjprs.2017.11.021.
[51] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Training very deep networks," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. New York, NY, USA: Curran Associates, 2015, pp. 2377–2385. [Online]. Available: http://papers.nips.cc/paper/5850-training-very-deep-networks.pdf
[52] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.
[53] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5987–5995.
[54] A. Veit, M. J. Wilber, and S. Belongie, "Residual networks behave like ensembles of relatively shallow networks," in *Advances in Neural Information Processing Systems 29*, D. D. Lee, MSugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. New York, NY, USA: Curran Associates, 2016, pp. 550–558. [Online]. Available: http://papers.nips.cc/paper/6556-residual-networks-behave-like-ensembles-of-relatively-shallow-networks.pdf
[55] D. Han, J. Kim, and J. Kim, "Deep pyramidal residual networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6307–6315, doi: 10.1109/CVPR.2017.668.
[56] Y. Yamada, M. Iwamura, and K. Kise. (2016). "Deep pyramidal residual networks with separated stochastic depth." [Online]. Available: https://arxiv.org/abs/1612.01230
[57] Z. Zhong, J. Li, L. Ma, H. Jiang, and H. Zhao, "Deep residual networks for hyperspectral image classification," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2017, pp. 1824–1827.
[58] L. Mou, P. Ghamisi, and X. X. Zhu, "Unsupervised spectral–spatial feature learning via deep residual Conv–Deconv network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 1, pp. 391–406, Jan. 2018.
[59] A. Villa, J. A. Benediktsson, J. Chanussot, and C. Jutten, "Hyperspectral image classification with independent component discriminant analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 12, pp. 4865–4876, Dec. 2011.
[60] A. A. Green, M. Berman, P. Switzer, and M. D. Craig, "A transformation for ordering multispectral data in terms of image quality with implications for noise removal," *IEEE Trans. Geosci. Remote Sens.*, vol. GRS-26, no. 1, pp. 65–74, Jan. 1988.
[61] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, J. Fürnkranz and T. Joachims, Eds. Madison, WI, USA: Omnipress, 2010, pp. 807–814.
[62] B. Xu, N. Wang, T. Chen, and M. Li. (2015). "Empirical evaluation of rectified activations in convolutional network." [Online]. Available: https://arxiv.org/abs/1505.00853
[63] R. O. Green *et al.*, "Imaging spectroscopy and the airborne visible/infrared imaging spectrometer (AVIRIS)," *Remote Sens. Environ.*, vol. 65, no. 3, pp. 227–248, Sep. 1998.
[64] B. Kunkel, F. Blechinger, R. Lutz, R. Doerffer, H. van der Piepen, and M. Schroder, "ROSIS (Reflective Optics System Imaging Spectrometer)—A candidate instrument for polar platform missions," *Proc. SPIE, Optoelectron. Technol. Remote Sens. Space*, vol. 868, pp. 134–142, Apr. 1988.
[65] B. Waske, S. van der Linden, J. Benediktsson, A. Rabe, and P. Hostert, "Sensitivity of support vector machines to random feature selection in classification of hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 7, pp. 2880–2889, Jul. 2010.
[66] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 2, pp. 513–529, Apr. 2012.
[67] G.-B. Huang and C.-K. Siew, "Extreme learning machine with randomly assigned RBF kernels," *Int. J. Inf. Technol.*, vol. 11, no. 1, pp. 16–24, 2005.

Mercedes E. Paoletti (S'17) received the B.Sc. and M.Sc. degrees in computer engineering from the University of Extremadura, Cáceres, Spain, in 2014 and 2016, respectively, where she is currently pursuing the Ph.D. degree with the Hyperspectral Computing Laboratory, Department of Technology of Computers and Communications through a University Teacher Training Program from the Spanish Ministry of Education.

Her research interests include remote sensing and analysis of very high spectral resolution with a focus on deep learning and high-performance computing.

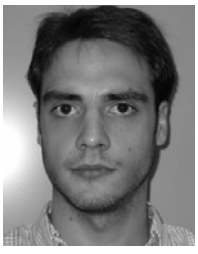
Juan Mario Haut (S'17) received the B.Sc. and M.Sc. degrees in computer engineering from the University of Extremadura, Cáceres, Spain, in 2011 and 2014, respectively, where he is currently pursuing the Ph.D. degree with the Hyperspectral Computing Laboratory, Department of Computers and Communications through a University Teacher Training Program from the Spanish Ministry of Education.

His research interests include remote sensing and analysis of very high spectral resolution with a focus on deep learning and cloud computing.

Ruben Fernandez-Beltran received the B.Sc. degree in computer science, the M.Sc. degree in intelligent systems, and the Ph.D. degree in computer science from Universitat Jaume I, Castellón de la Plana, Spain, in 2007, 2011, and 2016, respectively.

He was a Visiting Scientist with the University of Bristol, Bristol, U.K. Since 2017, he has been a Visiting Post-Doctoral Researcher with the Hyperspectral Computing Laboratory, University of Extremadura, Cáceres, Spain. He is currently a Post-Doctoral Researcher with the Computer Vision Group, University Jaume I, where he is a member of the Institute of New Imaging Technologies. His research interests include multimedia retrieval, spatiospectral image analysis, and pattern recognition techniques applied to image processing and remote sensing.

Dr. Fernandez-Beltran is member of the Spanish Association for Pattern Recognition and Image Analysis, which is a part of the International Association for Pattern Recognition. He was a recipient of the Outstanding Ph.D. Dissertation Award from Universitat Jaume I in 2017.



Javier Plaza (M'09–SM'15) received the M.Sc. and Ph.D. degrees in computer engineering from the Hyperspectral Computing Laboratory, Department of Technology of Computers and Communications, University of Extremadura, Cáceres, Spain, in 2004 and 2008, respectively.

He is currently a member of the Hyperspectral Computing Laboratory. He has authored more than 150 publications, including over 50 JCR journal papers, 10 book chapters, and 90 peer-reviewed conference proceeding papers. He has guest edited four special issues on hyperspectral remote sensing for different journals. His research interests include hyperspectral data processing and parallel computing of remote sensing data.

He was a recipient of the Outstanding Ph.D. Dissertation Award from the University of Extremadura in 2008, the Best Column Award of *IEEE Signal Processing Magazine* in 2015, the most highly cited paper in the *Journal of Parallel and Distributed Computing* from 2005 to 2010, the Best Paper Awards at the IEEE International Conference on Space Technology and the IEEE Symposium on Signal Processing and Information Technology. He is an Associate Editor of IEEE GEOSCIENCE AND REMOTE SENSING LETTERS and an Associate Editor of the IEEE Remote Sensing Code Library.

Antonio J. Plaza (M'05–SM'07–F'15) received the M.Sc. and Ph.D. degrees in computer engineering from the Hyperspectral Computing Laboratory, Department of Technology of Computers and Communications, University of Extremadura, Cáceres, Spain, in 1999 and 2002, respectively.

He is currently the Head of the Hyperspectral Computing Laboratory. He has authored more than 600 publications, including over 200 JCR journal papers (over 160 in the IEEE journals), 23 book chapters, and around 300 peer-reviewed conference proceeding papers. He has guest edited 10 special issues on hyperspectral remote sensing for different journals. His research interests include hyperspectral data processing and parallel computing of remote sensing data.

Dr. Plaza was a recipient of the Recognition of Best Reviewers of IEEE GEOSCIENCE AND REMOTE SENSING LETTERS in 2009, the Recognition of Best Reviewers of the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING in 2010, the Recognition as an Outstanding Associate Editor of the IEEE ACCESS in 2017, the Best Column Award of *IEEE Signal Processing Magazine* in 2015, the 2013 Best Paper Award of the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING, and the most highly cited paper in the *Journal of Parallel and Distributed Computing* from 2005 to 2010, the Best Paper Awards at the IEEE International Conference on Space Technology and the IEEE Symposium on Signal Processing and Information Technology. He was a member of the Editorial Board of the IEEE Geoscience and Remote Sensing Newsletter from 2011 to 2012 and *IEEE Geoscience and Remote Sensing Magazine* in 2013. He was also a member of the Steering Committee of JSTARS. He served as an Associate Editor for the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING from 2007 to 2012 and currently an Associate Editor for the IEEE ACCESS. He served as the Director of Education Activities for the IEEE Geoscience and Remote Sensing Society (GRSS) from 2011 to 2012 and as the President of the Spanish Chapter of the IEEE GRSS from 2012 to 2016. He has reviewed more than 500 manuscripts for over 50 different journals. He served as the Editor-in-Chief for the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING from 2013 to 2017.

Filiberto Pla received the B.Sc. and the Ph.D. degrees in physics from the Universitat de València, Valencia, Spain, in 1989 and 1993, respectively.

He has been a Visiting Scientist with the Silsoe Research Institute, Bedford, U.K., the University of Surrey, Surrey, U.K., the University of Bristol, Bristol, U.K., CEMAGREF, Montpelier, France, the University of Genoa, Genoa, Italy, the Instituto Superior Tcnico of Lisbon, Lisbon, Portugal, the Swiss Federal Institute of Technology, Zürich, Switzerland, the idiap Research Institute, Martigny, Switzerland, the Technical University of Delft, Delft, The Netherlands, and Mid Sweden University, Sundsvall, Sweden. He has been the Director of the Institute of New Imaging Technologies, University Jaume I, Castellón de la Plana, Spain. He is currently a Full Professor with the Departament de Llenguatges i Sistemes Informtics, University Jaume I, Castellón, Spain. His research interests include color and spectral image analysis, visual motion analysis, 3-D image visualization, and pattern recognition techniques applied to image processing.

Dr. Pla is a member of the Spanish Association for Pattern Recognition and Image Analysis, which is a member of the International Association for Pattern Recognition.