A Single Model CNN for Hyperspectral Image Denoising

Alessandro Maffei, Juan M. Haut, *Member, IEEE*, Mercedes E. Paoletti, *Student Member, IEEE*, Javier Plaza, *Senior Member, IEEE*, Lorenzo Bruzzone, *Fellow, IEEE*, and Antonio Plaza, *Fellow, IEEE*

***Abstract*— Denoising is a common preprocessing step prior to the analysis and interpretation of hyperspectral images (HSIs). However, the vast majority of methods typically adopted for HSI denoising exploit architectures originally developed for grayscale or RGB images, exhibiting limitations when processing high-dimensional HSI data cubes. In particular, traditional methods do not take into account the high spectral correlation between adjacent bands in HSIs, which leads to unsatisfactory denoising performance as the rich spectral information present in HSIs is not fully exploited. To overcome this limitation, this article considers deep learning models—such as convolutional neural networks (CNNs)—to perform spectral–spatial HSI denoising. The proposed model, called HSI single denoising CNN (HSI-SDeCNN), efficiently takes into consideration both the spatial and spectral information contained in HSIs. Experimental results on both synthetic and real data demonstrate that the proposed HSI-SDeCNN outperforms other state-of-the-art HSI denoising methods. Source code: https://github.com/mhaut/HSI-SDeCNN**

***Index Terms*— Convolutional neural networks (CNNs), denoising, hyperspectral images (HSIs), spatial–spectral information.**

I. Introduction

HYPERSPECTRAL sensors (also called imaging spectrometers) collect the information across the electromagnetic spectrum in several contiguous and narrow bands, producing high-dimensional hyperspectral images (HSIs) (or data cubes) with hundreds of spectral bands [1], [2]. Compared to other kinds of remotely sensed images, HSIs are characterized by the rich spectral information that they convey. Rather than focusing on spatial variations, the analysis of HSIs mainly focuses on spectral variations. In fact, the main idea behind HSIs is to enlarge the spectral dimension of a pixel so that it contains a unique and characteristic spectral signature for the underlying objects on the surface of the Earth. In this context, each pixel in an HSI is given by a B-dimensional vector, where B is the number of spectral channels or bands [3].

Although hyperspectral satellites are still poorly represented in space-borne missions, HSIs allow for better class discrimination than multispectral images [4], fostering their use in a wide range of application domains, including classification [5], spectral unmixing [6], and target detection [7], among many others. The quality of spectral signatures becomes crucial for the correct interpretation of HSIs. However the acquisition process introduces a significant amount of noise in the data, which leads to intraclass variability and interclass similarity [8]. This noise degradation is mainly due to two factors: instrumental acquisition limitations and atmospheric distortions [9].

In order to overcome these issues, image denoising is typically adopted as a preprocessing step for noise removal prior to HSI data analysis [10]. This is crucial for obtaining accurate results in tasks such as classification, unmixing and target detection. However, many techniques adopted for HSI denoising are based on approaches that were originally developed for grayscale or RGB images, disregarding the rich spectral information contained in each HSI pixel. Moreover, standard methods adopted for HSI denoising process the data in a band-by-band fashion, applying traditional 1-D or 2-D convolution kernels. Thus, they take into account only the spatial information and disregard the information across the bands, which is crucial for the analysis of spectral signatures. For instance, available models, such as block matching and 3-D filtering (BM3D) [11] or weighted nuclear norm minimization (WNNM) [12], have been applied to HSI images by considering each band as a 2-D image, which leads to large spectral distortions.

Another widely used strategy to denoise HSIs is to take into account groups of three adjacent bands at a time as in the case of the 3-D denoising convolutional neural network (3D-DnCNN) [13]. This strategy, which is adapted from techniques for RGB image denoising, often provides better performance due to the fact that it considers the spectral correlation between adjacent bands. However, given the large amount of spectral bands contained in HSIs, considering groups of three channels only represents a significant limitation. In the

Manuscript received June 5, 2019; revised September 25, 2019; accepted October 20, 2019. Date of publication November 26, 2019; date of current version March 25, 2020. This work was supported in part by the Spanish Ministry under Grant FPU14/02012-FPU15/02090, in part by the Junta de Extremadura (Decreto 14/2018, de 6 de febrero, por el que se establecen las bases reguladoras de las ayudas para la realizacion de actividades de investigacion y desarrollo tecnologico, de divulgacion y de transferencia de conocimiento por los Grupos de Investigacion de Extremadura, Ref. GR18060), and in part by the European Union's Horizon 2020 Research and Innovation Programme (EOXPOSURE) under Grant 734541. *(Corresponding author: Juan M. Haut.)*

A. Maffei and L. Bruzzone are with the Remote Sensing Laboratory, Department of Information Engineering and Computer Science, University of Trento, 38123 Trento, Italy (e-mail: alessandro093.maffei@hotmail.com; lorenzo.bruzzone@ing.unitn.it).

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

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

Digital Object Identifier 10.1109/TGRS.2019.2952062

literature, HSI denoising techniques have evolved to incorporate spectral information. Available methods can be divided into two main classes [14]: spatial filtering methods and transform-domain filtering methods.

1) *Spatial Filtering Methods:* include the algorithm proposed by Othman and Qian [15], a hybrid spatial–spectral noise reduction (HSSNR) scheme that operates almost independently in the spatial and spectral domains, trying to accommodate the dissimilarity between the spatial and the spectral dimensions. In this scheme, noise is first removed from the spatial domain, where the signal is relatively regular. Then, additional noise (as well as those artifacts that may have been introduced during the spatial denoising) are removed in the spectral domain. Letexier and Bourennane [16] adapted a generalized multidimensional Wiener filter (MWF) to HSIs. The main disadvantage of spatial filtering methods such as those mentioned earlier is that they are quite sensitive to the transform domain, and cannot consider the differences in terms of geometrical characteristics of HSIs. As a result, they are not widely used for HSI data denoising.
2) *Transform-Domain Filtering Methods:* include the approach by Yuan *et al.* [17], a spectral–spatial adaptive total variation(SSAHTV) model in which the spectral noise differences and the spatial information differences are both considered in the process of noise reduction. Jiang *et al.* [18] proposed an extension of the BM4D algorithm [19], which exploits the principal component analysis (PCA) to perform HSI denoising. The method by Lu *et al.* [20] is based on the spatial–spectral adaptive sparse representation (SSASR), while Zhao and Yang [21] fuse a sparse coding approach together with a low-rank method by exploiting the fact that HSIs are characterized by global and local redundancy, and correlation in the spatial and spectral domains. Zhang *et al.* [22] proposed a method called low-rank matrix recovery (LRMR), in which the low-rank property of HSIs is exploited, suggesting that a clean HSI patch can be regarded as a low-rank matrix. A subsequent method by He *et al.* [23], called spatial–spectral total variation regularized local LRMR (LLRSSTV), adopts a global total variation strategy to reconstruct the clean patches. Finally, the low-rank tensor approximation (LRTA) method by Li *et al.* [24] preserves the global structure of HSIs and simultaneously removes outliers and different types of noise.

The major drawback of the aforementioned spatial-domain and transform-domain methods is that, to achieve good performance, they need to fine-tune the hyperparameters for each HSI. This process is expensive from the viewpoint of computational time and resources, and often requires an external operator (i.e., a human expert) to correctly tune such parameters for different HSIs.

In the last few years, deep learning [25], in general, and CNNs [26], in particular, have been successfully used for automatic processing of image data [27], with outstanding results in tasks such as classification and object detection [28], [29]. This is mainly due to the following reasons: 1) the availability of very large training sets, with millions of labeled examples; 2) the possibility to use powerful graphics processing unit (GPU)-based implementations that make possible the efficient training of very large models in practice; and 3) the definition of accurate model regularization strategies, such as dropout [30]. In this sense, the application of deep learning architectures [31], [32] and CNN models resulted in powerful HSI data analysis techniques [33]–[36], including denoising methods too. However, many CNN-based denoising approaches are developed for grayscale or RGB images, and cannot fully exploit the rich spectral information contained in HSIs. Yuan *et al.* [37] proposed a residual CNN learning-based (HSID-CNN) method for HSI denoising, taking into consideration both the spatial and the spectral information and without the need to manually tune the hyperparameters for different HSI. This offers versatility, scalability, and generalization properties when dealing with HSI denoising tasks. Indeed, this method achieved the best HSI denoising performance among all the available methods in literature. However, it requires to train different models for each level of noise present in the data, which does not provide a global solution to the denoising problem.

In this article, an improved CNN architecture is developed to efficiently perform HSI denoising. The proposed architecture, inspired by a network typically used for grayscale and RGB images (named FFDNet [38]), is called HSI single-denoising CNN (HSI-SDeCNN). Instead of considering only the spatial information contained in the scene, our newly developed approach is able to jointly consider both the spatial and the spectral correlation, outperforming previously available techniques used in HSI denoising. The proposed HSI-SDeCNN model takes as input a 3-D HSI scene, i.e., a $h \times w$ data cube (being h the height and w the width), and whose spectral information is composed by a single band, together with its adjacent K bands (coupled with a noise-level map). Then it returns, as output, a single denoised image for each considered band. In practice, it takes as input a volume of $K + 1$ bands (stacked together with a noise-level map), and returns the central noise-free band. Thus, this method allows us to perform the denoising of the central band, taking as input its previous and subsequent $K/2$ bands, resulting in a spectral–spatial integration when denoising the HSI data.

The main advantages of the proposed HSI-SDeCNN with respect to previous models can be enumerated as follows.

1) It provides a fast solution to the HSI denoising problem, exploiting a down-sampling kernel that allows the network to perform very fast without losing performance.
2) It takes as input a noise-level map, i.e., an estimation of the noise level present in the volume to be denoised, which allows us to control the trade-off between denoising performance and detail preservation. This makes our network flexible and adaptive to multilevel noise, without the need to train different models for different noise levels as it is, for example, the case in [37].
3) In our experimental analysis, it provided excellent results on both synthetic images corrupted by additive white Gaussian noise (AWGN) [9] and real HSI images,

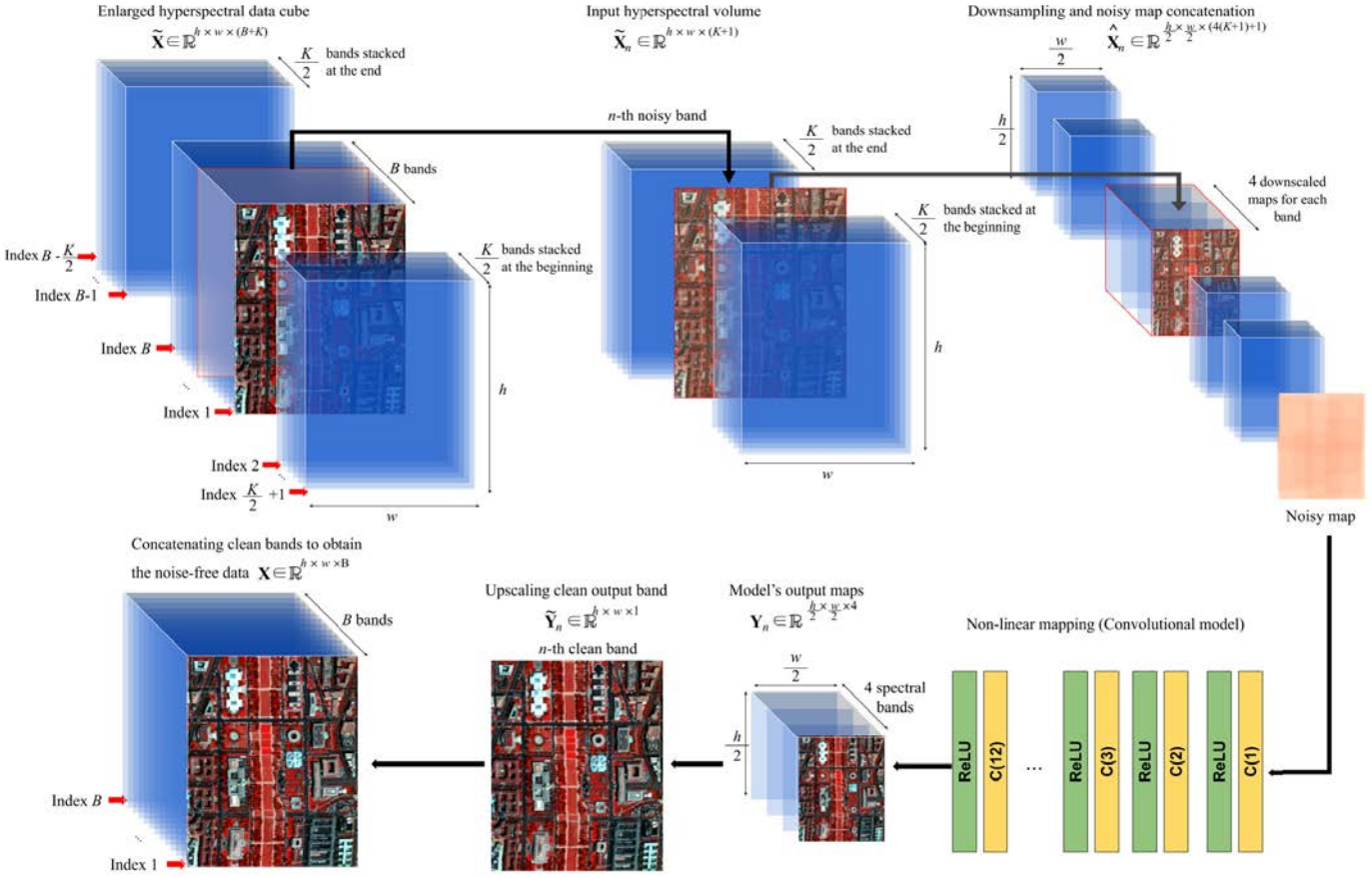


Fig. 1. Graphical illustration of the proposed methodology, composed of three main parts: 1) a preprocessing step; 2) a deep nonlinear mapping; and 3) a postprocessing step. First, a three-step preprocessing method is applied where $K/2$ adjacent bands from the original HSI scene are stacked at the beginning and at the end of the cube $\tilde{\mathbf{X}}$ in order to consider all the B spectral bands, including spectral–spatial information, to perform the denoising task. Then, for the nth band (with $n = \{1, \ldots, B\}$), the data cube $\tilde{\mathbf{X}}_n$ is extracted considering the $K/2$ adjacent front and back bands. The obtained data cube is reorganized in order to introduce the spatial-downsampled data representation $\hat{\mathbf{X}}_n$. This $\hat{\mathbf{X}}_n$ is sent to the CNN model, obtaining (at the end) a noise-free data representation $\mathbf{Y}_n$, which is upscaled to obtain a clean representation of the original nth spectral band $\tilde{\mathbf{Y}}_n$. Finally, the clean bands are concatenated together to recover a noise-free HSI scene $\mathbf{X}$.

demonstrating its full potential for practical HSI denoising applications.

The remainder of this article is organized as follows. Section II describes the proposed method and the adopted network architecture. Section III illustrates data sets, implementation details and considered metrics. Furthermore, Sections IV and V present, respectively, the experimental results obtained on simulated and real data, together with a comparison with the state-of-the-art methods. Finally, Section VI concludes this article with some remarks and hints for plausible future research lines.

II. Methodology

In this section, we describe the proposed HSI-SDeCNN model and how it can be applied to HSI denoising tasks, which can be processed by spectral, spatial, or spectral–spatial models [9]. In this sense, the HSI scene can be considered as a 3-D data structure, i.e., a volume denoted by $\mathbf{X} \in \mathbb{R}^{h \times w \times B}$, where $h \times w$ indicates the number of spectral samples (pixels), being each one a B-dimensional spectral vector $\mathbf{x}_{i,j} \in \mathbb{R}^B = [x_{i,j,1}, \ldots, x_{i,j,B}]$. On the one hand, standard spectral-based models consider the pixel $\mathbf{x}_{i,j}$ as an independent element, processing the spectral information in an isolated way and disregarding the spatial information that surrounds it. On the other hand, the spatial models only consider the spatial information extracted from a neighborhood window, disregarding the spectral correlation between bands. In this context, the proposed deep learning inspired model attempts to overcome these limitations by considering spatial–spectral patches. Consequently, the proposed model can be regarded as a spatial–spectral one.

The denoising process has been carried out under the assumption of AWGN. In particular, from the original noise-free version of the HSI scene $\mathbf{X}$, a noisy representation $\tilde{\mathbf{X}}$ is obtained by introducing an AWGN denoted as $N = \mathcal{N}(0, \sigma)$, which is based on a normal distribution with zero-mean and variance σ to easily control the noise level. This allows to obtain an independent and identically distributed noise, introducing a controlled noise intensity and simulating the effect of many random and uncontrolled processes that occur in real scenarios, such as the remote sensing data acquisition

$$\tilde{\mathbf{X}} = \mathbf{X} + N. \tag{1}$$

In this sense, the goal of the proposed HSI-SDeCNN model is to accurately recover from the corrupted data cube $\tilde{\mathbf{X}}$ a

noise-free image, $\mathbf{X}$, cleaning the data band by band, and incorporating spatial–spectral information. Fig. 1 provides a graphical illustration of the proposed method. As we can observe, the overall structure of the proposed method can be divided into three main parts: 1) a preprocessing stage, where the HSI data is prepared to be given as input to the neural model, performing a spectral elongation of the data coupled with downsampling and noise-level map concatenation steps; 2) the path through the neural model, which extracts more abstract representations of the data, performing a nonlinear mapping to obtain a noise-free output; and 3) a postprocessing stage to recover the full HSI scene without noise, which includes an upsampling of the network's output and the concatenation of the denoised spectral bands. In the following, these stages will be describe in detail.

A. Spectral Elongation of the HSI Cube

The adopted strategy scans the spectral dimension in a raster way and performs denoising one band at a time. That is, the proposed HSI-SDeCNN model performs the denoising task band by band, including the spatial information contained into a neighborhood region $h \times w$ and the spectral correlation between the target band and the K adjacent bands. The motivation for this choice is that neighboring bands exhibit high correlation, which decreases for bands that are further away in terms of wavelength [39].

Based on this insight, for the nth band with spatial size $h \times w$, a data cube of size $h \times w \times (K+1)$ is considered, where K is the number of adjacent bands with respect to the central one, at position $K/2+1$. This data cube is obtained by taking into account the front and back adjacent $(K/2)$-bands, i.e., from the $n-(K/2)$th band to the $n+(K/2)$th spectral band. For example, if we consider $K=24$, the input volume fed to the HSI-SDeCNN model will be of size $h \times w \times 25$, and the output denoised band will be the one at position 13 (i.e., the central one). In this way, the proposed model exploits the spatial–spectral correlation between the central band and the K adjacent bands to provide a noise-free version of the central band.

In this sense, in order to perform the denoising process on the B available bands, the original HSI scene of size $h \times w \times B$ must be spectrally elongated so that further contiguous bands are concatenated to the data cube at the beginning and at the end, generating a volume of size $h \times w \times (B+K)$. The bands are stacked in reverse order, as shown in Fig. 1. In this way, we can perform the denoising of all the B bands in the original HSI cube, scanning for bands in a raster way. In order to follow a simple mathematical notation, we will denote the elongated and noisy data cube as $\tilde{\mathbf{X}} \in \mathbb{R}^{h \times w \times (B+K)}$ and the input data cube obtained from the nth spectral band as $\tilde{\mathbf{X}}_n \in \mathbb{R}^{h \times w \times (K+1)}$.

B. Downsampling

This operation is performed by a downsampling kernel that reshapes the input HSI volume $\tilde{\mathbf{X}}_n \in \mathbb{R}^{h \times w \times (K+1)}$ into several downsampled subcubes in order to reduce the spatial dimension of the cube without losing information. Indeed, the applied downsampling operation is a way of doubling the receptive field, which sensibly reduces runtimes and memory requirements while maintaining the denoising performance. The scale factor is set to 2. In practice, this operation takes the tth band of $\tilde{\mathbf{X}}_n$ (i.e., t works as an index of the $(K+1)$ spectral bands of $\tilde{\mathbf{X}}_n$, being $t = \{n-K/2, \ldots, n+K/2\}$) and reorganizes the $h \times w$ pixels contained in it into four subcubes, each one with size $w/2 \times h/2$, rearranging the pixels in the different channels of the output image according to [40]

$$\hat{\mathbf{X}}_n(i, j, t) = \tilde{\mathbf{X}}_n \left(2i + (t \bmod 2),\ 2j + \left\lfloor \frac{t}{2} \right\rfloor,\ \left\lfloor \frac{t}{4} \right\rfloor \right) \quad (2)$$

where (i, j) indicates the spatial position of the resulting pixel at band t, while "mod" and $\lfloor\ \rfloor$ denote the magnitude and the "floor" operations, respectively. Moreover, $\tilde{\mathbf{X}}_n$ is the input image extracted from $\tilde{\mathbf{X}}$, and $\hat{\mathbf{X}}_n$ is the output of the downsampling operation. In fact, this $\hat{\mathbf{X}}_n$ will be the input of the CNN model, whose goal is to recover a noise-free image of the nth band.

Further details about the downsampling layer employed in the proposed method can be found in [40] and [41]. This process is applied to all the spectral channels, and the obtained subcubes are concatenated along the spectral dimension, generating the output volume $\hat{\mathbf{X}}_n \in \mathbb{R}^{h/2 \times w/2 \times 4(K+1)}$. This operation allows the network to be fast, without losing information, which in the end improves the denoising performance.

C. Noise-Level Map Concatenation

In order to complete the information contained into the network's input data $\hat{\mathbf{X}}_n$, as a previous step, a noise-level map $\mathbf{M} \in \mathbb{R}^{h/2 \times w/2}$ is concatenated to the generated subcubes, obtaining a volume of size $h/2 \times w/2 \times (4(K+1)+1)$. The noise-level map gives an estimation of the level of noise σ present in the image. It is inserted as a map having the same spatial dimension as the subcubes, in order to avoid any mismatch in terms of dimensionality [38].

In this way, the network exploits this prior information to control the trade-off between denoising performance and detail preservation. This is because, as opposed to common residual learning methods, adding a noise-level map makes the model parameters (i.e., weights and biases) invariant to the noise level of the input image. Thus, with this approach, it is possible to handle both different noise levels and spatially variant noise, with a single network architecture. Most model-based denoising methods aim to solve the following problem:

$$\arg\min_{\mathbf{X}} \frac{1}{\sigma^2} \|\tilde{\mathbf{X}} - \mathbf{X}\|^2 + \lambda \Phi(\mathbf{X}) \quad (3)$$

where $(1/\sigma^2)\|\tilde{\mathbf{X}} - \mathbf{X}\|^2$ is the discrepancy between the noise-free data $\mathbf{X}$ and the noisy data $\tilde{\mathbf{X}}$ (with noise level σ) and $\Phi(\mathbf{X})$ is a regularization term associated with the image prior. In this regard, the noise map $\mathbf{M}$ plays the role of λ in controlling the trade-off between detail preservation and denoising performance [38]. This improves the network's flexibility, which can handle images with various noise levels by simply specifying the associated noise level map. $\mathbf{M}$ in our case is a uniform matrix in which all elements are σ. Notice that the value of

the noise level map (we refer to this as input noise level) can differ from the noise level effectively present in the image (we refer to this as ground-truth noise level). For this reason, in the following, we denote the input noise level as $\hat{\sigma}$. In the testing phase, we obtain the best results when the input noise level map matches the noise level of the input image ($\hat{\sigma} = \sigma$). The results are degraded when there is a mismatch in the values. Further analyses are presented in Section V.

D. Nonlinear Mapping

At this point, the obtained volume $\hat{\mathbf{X}}_n$ is fed to a standard CNN. This model is composed by a stack of convolutional and nonlinear activation layers. In this sense, the convolutional (Conv) layer performs the basic feature extraction task of the model, obtaining at each time a deep and abstract representation of the input data. In particular, the proposed model exploits Conv layers defined by 2-D kernels of size $K \times k \times k$, where the lth layer (denoted as $C^{(l)}$), receives as input the feature data $\mathbf{X}^{(l-1)}$ obtained by the previous layer. Thus, K filters of size $k \times k$ are overlapped over $\mathbf{X}^{(l-1)}$; sliding across the width and height with a particular stride s, the following equation indicates:

$$\begin{aligned} \mathbf{Y}^{(l)} &= (\mathbf{W}^{(l)} *_{k\times k} \mathbf{X}^{(l-1)}) + b^{(l)} \\ y_{i,j}^{(l)_z} &= \sum_{\hat{i}=1}^{k}\sum_{\hat{j}=1}^{k} \left(x_{(i\cdot s+\hat{i}),(j\cdot s+\hat{j})}^{(l-1)} \cdot w_{\hat{i},\hat{j}}^{(l)_z} \right) + b^{(l)}. \end{aligned} \tag{4}$$

In this sense, the output data $\mathbf{Y}^{(l)}$ could be obtained by overlapping the layer's weights $\mathbf{W}^{(l)}$ with the input volume data $\mathbf{X}^{(l-1)}$. In fact, this operation performs a linear dot product between the $(\hat{i}, \hat{j})$th weight of the zth filter that is composed of the Conv layer, $w_{\hat{i},\hat{j}}^{(l)_z}$ (with $z = \{1, \ldots, K\}$), and the corresponding (i, j)th element of the input data, $x_{i,j}^{(l-1)}$. Finally, the bias of the layer $b^{(l)}$ is added to the dot product, obtaining as a result the (i, j)th element of the zth filter of the output volume, $y_{i,j}^{(l)_z}$.

With the exception of the last layer, each Conv layer is followed by a nonlinear activation layer, which is introduced in order to extract the activation maps from the convolutional output volume $\mathbf{Y}^{(l)}$. In particular, this layer performs a function $\mathcal{H}(\cdot)$ to obtain the nonlinear relationships between the data

$$\mathbf{X}^{(l)} = \mathcal{H}(\mathbf{Y}^{(l)}). \tag{5}$$

$\mathcal{H}(\cdot)$ can be implemented by many activation functions, such as the tanh, sigmoid or rectified linear unit (ReLU) [42]. In the proposed model, the ReLU function has been selected, which allows a faster training of the model due to its high computational efficiency. The resulting output volume $\mathbf{X}^{(l)}$ is then sent as input to the next pair of Conv-ReLU layers.

Focusing on the proposed HSI-SDeCNN model, the implemented CNN aims to learn a nonlinear mapping function being able to recover the noise-free image from the noisy one. It takes as input the data $\hat{\mathbf{X}}_n$ of size $h/2 \times w/2 \times (4(K + 1) + 1)$, obtaining as output the data $\mathbf{Y}_n$ of size $h/2 \times w/2 \times 4$. The output volume $\mathbf{Y}_n$ represents the four downsampled noise-free subcubes. For this reason, the last layer does not have any activation function in order to keep the extracted features. We have set the kernel size of each layer to 3×3, while zero padding is employed to maintain the original size of the feature maps. The number of layers in the CNN is fixed to 14, while the number of channels for each convolutional layer is set to 128, except for the last one, where we use 4. The main reason why we use a larger number of channels with respect to the standard FFDNet is the fact that our network takes more channels as input, and hence more features are required. As mentioned earlier, the noise-level map controls the trade-off between denoising performance and detail preservation. Furthermore, when the noise-level map given as input to the network contains too high values compared to the noise level of the input image, the obtained denoised image is corrupted by artifacts [38]. For this reason, the proposed HSI-SDeCNN model initializes the parameters of each Conv layer using the orthogonal initialization method, making the network more robust to changes in the noise level.

E. Upsampling and Concatenation

The final layer of our HSI-SDeCNN method is an upsampling kernel that performs the inverse function of downscaling, taking as input the four downsampled, noise-free images that compose the CNN output volume $\mathbf{Y}_n$, and provides as output a noise-reduced single band $\tilde{\mathbf{Y}}_n$ of size $h \times w \times 1$. The reason for which the number of channels of the last CNN layer is set to 4 is that we only expect one denoised band as output, in particular the denoised version of the nth original band selected as an input of the model. Thus, with an upsampling factor of 2, this layer takes as input four subimages and provides a single noise-free band.

Finally, once the B spectral bands have been processed, their corresponding denoised version are stacked together in order to compose the noise-free HSI scene $\mathbf{X}$ with size $h \times w \times B$. A graphical representation of the overall process is shown in Fig. 1, while Table I provides the details of the implemented HSI-SDeCNN topology.

The methodology used in the proposed HSI-SDeCNN allows us to achieve better performance than the standard network when performing HSI data denoising tasks. Two main improvements can be highlighted in comparison with other denoising models. First and foremost, our method takes as input a significantly larger number of bands, which allows the network to exploit the spectral correlation between channels (which is very high in HSIs). Second, since our network considers an overlapping volume of bands, it can learn from a larger amount of data, resulting in much better denoising performance. In fact, the proposed method exhibits better performance (both in terms of denoising and computational time) when compared with other learning-based methods. This is mainly due to 1) the downsampling layer, which allows us to make the network faster without degrading performance, and 2) the input noise-level map, which is used as prior information in order to achieve better denoising performance.

III. Design of Experiments

We have evaluated the proposed HSI-SDeCNN method using both synthetic and real HSIs. First, the effectiveness

TABLE I
DATA VOLUMES AND CNN TOPOLOGY

Down-sampling				
Layer ID	*Input data size*		*Output data size*	
DS	$\hat{\mathbf{X}}_n \in \mathbb{R}^{h\times w\times(K+1)}$		$\hat{\mathbf{X}}_n \in \mathbb{R}^{\frac{h}{2}\times\frac{w}{2}\times 4(K+1)}$	
Noisy-map Concatenation				
Layer ID	*Input data size*		*Output data size*	
CN	$\hat{\mathbf{X}}_n \in \mathbb{R}^{\frac{h}{2}\times\frac{w}{2}\times 4(K+1)}$		$\hat{\mathbf{X}}_n \in \mathbb{R}^{\frac{h}{2}\times\frac{w}{2}\times(4(K+1)+1)}$	
Non-linear mapping				
Layer ID	*Kernel size*	*Number of kernels*	*Stride*	*Activation function*
$C^{(input)}$	3×3	128	1	ReLU
$C^{(1)}$	3×3	128	1	ReLU
$C^{(2)}$	3×3	128	1	ReLU
$C^{(3)}$	3×3	128	1	ReLU
$C^{(4)}$	3×3	128	1	ReLU
$C^{(5)}$	3×3	128	1	ReLU
$C^{(6)}$	3×3	128	1	ReLU
$C^{(7)}$	3×3	128	1	ReLU
$C^{(8)}$	3×3	128	1	ReLU
$C^{(9)}$	3×3	128	1	ReLU
$C^{(10)}$	3×3	128	1	ReLU
$C^{(11)}$	3×3	128	1	ReLU
$C^{(12)}$	3×3	128	1	ReLU
$C^{(output)}$	3×3	4	1	
Up-sampling				
Layer ID	*Input data size*		*Output data size*	
US	$\mathbf{Y}_n \in \mathbb{R}^{\frac{h}{2}\times\frac{w}{2}\times 4}$		$\hat{\mathbf{Y}}_n \in \mathbb{R}^{h\times w\times 1}$	

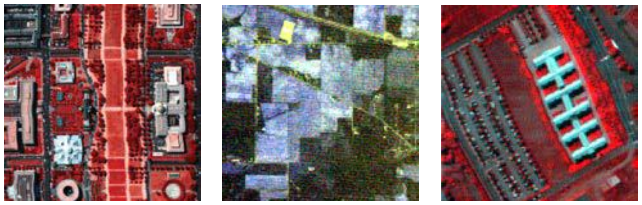
of the method has been validated using simulated data. Then, the method has been applied to real noisy images and the results are compared with those of the current mainstream approaches typically adopted in HSI denoising: HSSNR [15], LRTA [24], block matching and 4-D algorithm (BM4D) [18], LRMR [22], and HSI denoising exploiting a spatial–spectral deep residual CNN (HSID-CNN) [37]. A quantitative and qualitative analysis has been conducted for both simulated and real data. Several quantitative metrics have been adopted, together with a qualitative interpretation of false-color and grayscale images.

A. Data Sets

In order to assess the effectiveness of the proposed method, three HSIs have been considered: one of them is employed to train the network and to conduct experiments by introducing simulated noise, while the other two are used to evaluate the proposed approach in real scenarios.

1) Training Data Set: In order to train the proposed model, we have selected a part of the Washington DC Mall image acquired by the Hyperspectral Digital Imagery Collection Experiment (HYDICE) airborne sensor. This sensor records 210 spectral bands in the 0.4- to 2.4-μm region of the visible and infrared spectrum. Bands in the 0.9- and 1.4- μm regions (in which atmospheric interferers are present) have been removed from the data set, resulting in a total of 191 bands. The size of the Washington DC Mall image is therefore $1208 \times 307 \times 191$. The image has been divided into two parts: one is used for training the proposed network and the other is used for testing purposes. For the testing part, we have cropped a region of size $200 \times 200 \times 191$ from the full image (the remaining parts are used for training).

2) Testing Data Sets: In order to evaluate the effectiveness of the proposed method in real scenarios, experiments have been conducted on the following data sets.



(a) (b) (c)

Fig. 2. Images used in the experiments: (a) Washington DC Mall, (b) Indian Pines, and (c) University of Pavia.

1) *Washington DC Mall:* A cropped part of the entire image (with size $200 \times 200 \times 191$) has been employed for experiments on simulated data in which synthetic noise is added to the original image.
2) *Indian Pines:* This data set, acquired by the Airborne Visible Infra-Red Imaging Spectrometer (AVIRIS), consists of 145×145 pixels and 224 spectral bands. After removing the water absorption bands ($150 - 163$), the remaining 206 bands are retained for experiments.
3) *University of Pavia:* This data set, acquired by the Reflective Optics Spectrographic Imaging System (ROSIS), consists of 610×610 pixels and 103 spectral bands. For testing purposes, only a cropped part of size $200 \times 200 \times 103$ has been employed for experiments.

Fig. 2 shows a false-color composition of the three images used in the experiments (we emphasize that the Washington DC Mall and University of Pavia are cropped versions of the original images).

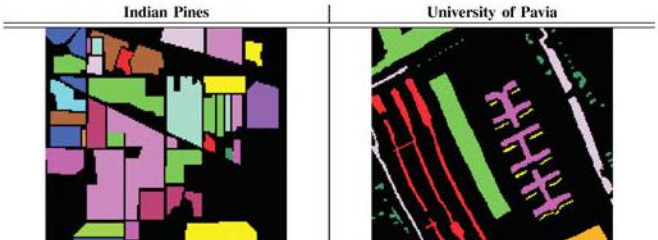
B. Accuracy Metrics

In order to evaluate the performance of the proposed approach on the simulated data, three commonly employed metrics have been adopted: mean peak signal-to-noise-ratio (MPSNR), mean structural similarity index (MSSIM), and mean spectral angle (MSA). These metrics are respectively used to calculate the average of the peak SNR (PSNR), the structural similarity index (SSIM) [43], and the spectral angle (SA) [44], [45] in the spectral domain.

For the real data experiments, since we do not have a reference clean image, the performance of the method was evaluated by conducting classification tasks. First, we apply the denoising method to the real data, and then we conduct classification (before and after the denoising process). As a result, in the real data experiments, the quality metrics employed are the overall accuracy (OA) and the kappa coefficient of the resulting classification maps. Fig. 3 shows the ground-truth images of the Indian Pines and University of Pavia data sets used in this article to evaluate the accuracy of the classification task.

C. Implementation Details

In the following, we describe some implementation details regarding the experiments. Before the denoising process, each band of the considered HSIs has been scaled between [0,1]. In order to make a proper comparison with the HSID-CNN network in [37], the number of adjacent spectral bands K given as input to the network has been fixed to $K = 24$.



Indian Pines			University of Pavia		
Color	Land-cover type	Samples	Color	Land-cover type	Samples
	Background	10659		Background	30690
	Alfalfa	54		Asphalt	1275
	Corn-notill	1434		Meadows	2998
	Corn-min	834		Gravel	10
	Corn	234		Trees	402
	Grass/Pasture	497		Painted metal sheets	1345
	Grass/Trees	747		Bare Soil	1319
	Grass/pasture-mowed	26		Bitumen	46
	Hay-windrowed	489		Self-Blocking Bricks	1700
	Oats	20		Shadows	215
	Soybeans-notill	968			
	Soybeans-min	2468			
	Soybean-clean	614			
	Wheat	212			
	Woods	1294			
	Bldg-Grass-Tree-Drives	380			
	Stone-steel towers	95			
	Total samples	21025		Total samples	40000

Fig. 3. Ground-truth maps and related number of training samples of the AVIRIS Indian Pines and the ROSIS University of Pavia Scenes.

The denoising task is performed for one band at a time, meaning that for denoising a single band, the network takes as input a volume of size $h \times w \times K + 1$. All the bands are scanned in a raster way. In order to perform the denoising process on the first and last $K/2$ bands, further adjacent bands are concatenated to the full image of size $h \times w \times B$, at the beginning and at the end, generating a volume of size $h \times w \times B + K$, as described in Section II. The proposed model was trained with the Adam [46] optimizer, adopted to minimize the following loss function:

$$L(\Theta) = \frac{1}{2N} \sum_{i=1}^{N} \left\| x_i^{\text{denoised}} - x_i^{\text{noise-free}} \right\|^2 \tag{6}$$

where N is the number of batches, x^{denoised} is the output of the network (i.e., the denoised batch), and $x^{\text{noise-free}}$ is the label batch. We set the patch size to 20, with stride equal to 20, and we used rotation and flip-based data augmentation during the training process, in which the noisy patches are generated by adding different levels of AWGN noise ($\sigma = [0, 100]$) to the clean patches. Note that the network has been trained following the model proposed in (1). However, rather than adding noise to the entire clean HSI, noise has been inserted in a patch-wise manner, with different noise configurations at each epoch. In this way, the network is able to learn different noise configurations, thus avoiding the problem of the redundancy of the data in the training process.

Regarding the noise-level map M, it is given as input at the same time as the specific noisy patch. For example, let us assume that, for one specific patch, AWGN noise with level $\sigma = 25$ is inserted. Then, the noise-level map M will be a uniform matrix of size $w/2 \times h/2 \times 1$ in which all elements are equal to $\hat{\sigma} = 25$. This allows the network to handle different levels of noise without changing the model, only by simply changing the input noise-level map. We set the mini-batch size to 128.

The proposed HSI-SDeCNN has been trained with patches extracted from the Washington DC Mall image. The total number of patches extracted was 162 350 and, after data augmentation, we obtained a total number of patches $n_{\text{patches}} = 324\,864$. We employed the MatConvNet [47] framework to train the proposed HSI-SDeCNN on a PC having a sixth Generation Intel Core i7-6700K processor with 8 M of Cache and up to 4.20 GHz (4 cores/8-way multitask processing), 40 GB of DDR4 RAM with a serial speed of 2400 MHz, an NVIDIA GeForce GTX 1080 GPU with 8-GB GDDR5X of video memory and 10 Gb/s of memory frequency, a Toshiba DT01ACA HDD with 7200 rpm and 2 TB of capacity, and an ASUS Z170 pro-gaming motherboard. The software environment is composed of Ubuntu 16.04.4 x64 as the operating system, Matlab R2018b, and the compute device unified architecture (CUDA) 9 for GPU functionality. The training process is performed using 200 epochs.

IV. Experimental Results: Simulated Data

In this section, we present the results obtained on simulated data related to the Washington DC Mall data set (test image). In order to perform the experiments, AWGN noise has been added to the noise-free HSI. We considered the same maximum level of noise for each band, where $\sigma_n = [5, 100]$. Here, n indicates a generic band with $n \in [1, B]$.

A. Results

Table II shows the results obtained for different noise levels by the proposed method and the other mainstream techniques used for comparison. The best metric values are presented in bold. The reported values (mean and standard deviation) are obtained by averaging the results on ten runs with different noise configurations. The results of the last column (in blue) were obtained after an ensemble of ten different runs for each noise level and are displayed only for illustrative purposes (they are not intended to make a comparison with the other methods). As shown in Table II, our method provided the best results for high noise levels. For low noise levels (such as $\sigma_n = 5$), it exhibited performances comparable to those of the others methods but using only one model.

For visual comparison purposes, we have selected bands 57, 27 and 17 to generate false-color images. Fig. 4 displays the results obtained with $\sigma_n = 100$. Specifically, Fig. 4(a) shows the noisy image before the denoising process, while Fig. 4(b) shows the ground-truth image. Fig. 4(c)–(h) shows the resulting images obtained after applying different denoising methods. We can see that the HSID-CNN and the proposed method outperform all other methods. In particular, the denoised images provided by HSSNR and LRMR present significant residual noise, while the images produced by BM4D and LRTA contain artifacts. Instead, HSID-CNN and the proposed HSI-SDeCNN generate denoised images that are very similar to the ground-truth one. Furthermore, Fig. 5 shows the PSNR and SSIM for each band. We can see that, in band 57, the proposed method obtains lower performance with regard

TABLE II

QUANTITATIVE EVALUATION OF THE PROPOSED METHOD AGAINST THE MAINSTREAM METHODS FOR HSI DENOISING (SIMULATED DATA SET)

Noise Level	Index	HSSNR	LRTA	BM4D	LRMR	HSID-CNN	HSI-SDeCNN	Ensemble
$\sigma_n = 5$	MPSNR	39.890 ± 0.0023	39.009 ± 0.0034	41.188 ± 0.0023	40.878 ± 0.0036	$\mathbf{41.684 \pm 0.0025}$	39.989 ± 0.0032	41.4990 ± 0.0025
	MSSIM	0.9946 ± 0.0001	0.9926 ± 0.0002	0.9962 ± 0.0001	0.9952 ± 0.0001	$\mathbf{0.9966 \pm 0.0001}$	0.9954 ± 0.0000	0.9965 ± 0.0000
	MSA	2.3552 ± 0.0013	2.7008 ± 0.0015	1.9326 ± 0.0008	2.2760 ± 0.0011	$\mathbf{1.8318 \pm 0.0012}$	2.2356 ± 0.0019	1.9845 ± 0.0005
$\sigma_n = 25$	MPSNR	28.018 ± 0.0024	30.672 ± 0.0033	31.136 ± 0.0025	33.029 ± 0.0023	33.050 ± 0.0028	$\mathbf{33.444 \pm 0.0080}$	36.7429 ± 0.0039
	MSSIM	0.9361 ± 0.0001	0.9629 ± 0.0002	0.9685 ± 0.0002	0.9809 ± 0.0001	0.9813 ± 0.0001	$\mathbf{0.9822 \pm 0.0000}$	0.9901 ± 0.0000
	MSA	8.1332 ± 0.0034	5.7962 ± 0.0056	5.0514 ± 0.0048	4.6097 ± 0.0028	4.2641 ± 0.0026	$\mathbf{3.9129 \pm 0.0080}$	2.8833 ± 0.0030
$\sigma_n = 50$	MPSNR	22.232 ± 0.0036	26.832 ± 0.0052	26.752 ± 0.0034	28.806 ± 0.0043	28.968 ± 0.0039	$\mathbf{29.612 \pm 0.0111}$	33.0088 ± 0.0096
	MSSIM	0.8233 ± 0.0001	0.9246 ± 0.0001	0.9208 ± 0.0002	0.9952 ± 0.0001	0.9532 ± 0.0001	$\mathbf{0.9608 \pm 0.0001}$	0.9787 ± 0.0001
	MSA	14.413 ± 0.0048	7.4996 ± 0.0054	7.1405 ± 0.0056	6.8008 ± 0.0034	6.2197 ± 0.0045	$\mathbf{5.3806 \pm 0.0137}$	3.6854 ± 0.0041
$\sigma_n = 75$	MPSNR	18.780 ± 0.0047	24.682 ± 0.0054	24.261 ± 0.0035	26.306 ± 0.0046	26.753 ± 0.0039	$\mathbf{27.351 \pm 0.0119}$	30.6889 ± 0.0070
	MSSIM	0.7082 ± 0.0002	0.8866 ± 0.0001	0.8670 ± 0.0001	0.9192 ± 0.0001	0.9273 ± 0.0001	$\mathbf{0.9371 \pm 0.0002}$	0.9648 ± 0.0001
	MSA	19.904 ± 0.0053	8.4426 ± 0.0057	8.6010 ± 0.0064	8.5644 ± 0.0067	7.5246 ± 0.0052	$\mathbf{6.4767 \pm 0.0159}$	4.2273 ± 0.0067
$\sigma_n = 100$	MPSNR	16.314 ± 0.0065	23.175 ± 0.0048	22.577 ± 0.0054	24.310 ± 0.0047	25.296 ± 0.0043	$\mathbf{25.753 \pm 0.0121}$	28.9791 ± 0.0104
	MSSIM	0.6049 ± 0.0001	0.8494 ± 0.0003	0.8119 ± 0.0002	0.8799 ± 0.0002	0.9014 ± 0.0001	$\mathbf{0.9121 \pm 0.0002}$	0.9487 ± 0.0001
	MSA	24.732 ± 0.0065	9.1219 ± 0.0072	9.7611 ± 0.0068	10.468 ± 0.0074	8.4061 ± 0.0063	$\mathbf{7.3951 \pm 0.0182}$	4.7012 ± 0.0074

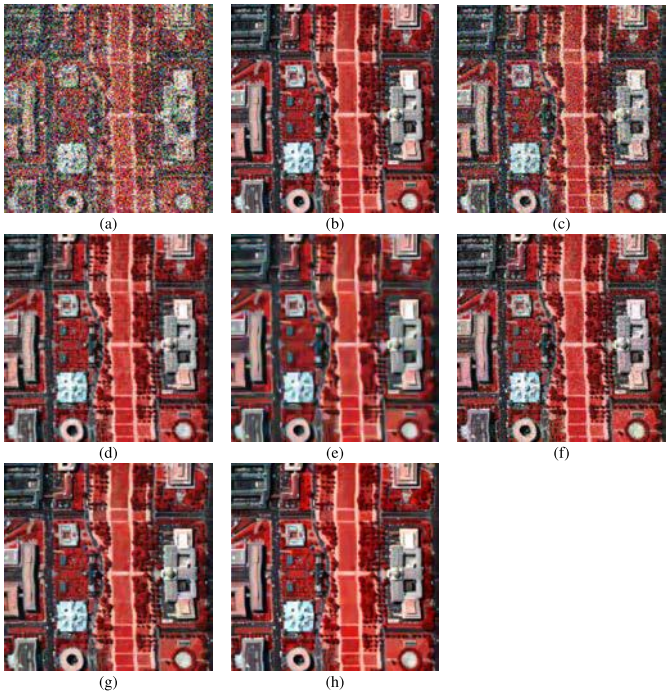


Fig. 4. Denoising results on the Washington DC Mall image (experiments on simulated data, with $\sigma_n = 100$). Bands 57, 27, and 17 are selected to generate false-color images. (a) Noisy. (b) Ground-truth. (c) HSSNR. (d) LRTA. (e) BM4D. (f) LRMR. (g) HSID-CNN. (h) Proposed.

to HSID-CNN. This is the reason why there is no visual improvement in the reported bands.

A more detailed assessment is presented in Fig. 6, which displays two zoomed regions of the Washington DC Mall test image. It is possible to notice that the proposed and the HSID-CNN methods obtain apparently similar results from a visual point of view, but a quantitative analysis demonstrates that our method performs better: HSID-CNN obtains an MPSNR of 25.29 ± 0.0043, while the proposed method obtains an MPSNR of 25.75 ± 0.0121. If we compare the denoised images in Fig. 6(g) and in Fig. 6(b)–(e), it is clear that the one obtained by the proposed method presents lower residual noise than those produced by the other techniques, without introducing as much blurring as the BM4D. This is due to the fact that our method exploits the prior information given from the input noise-level map, allowing the network to maintain a good trade-off between denoising performance and detail preservation.

It is important to emphasize that the quality of spectral signatures is crucial for HSI interpretation, due to the fact that they allow the discrimination of the physical properties of different ground objects. In order to further provide information about the effectiveness of the proposed method versus HSID-CNN, Fig. 7 reports an analysis of the spectral signature of a pixel. We can see that the spectral signature obtained with the proposed method for the analyzed pixel is the most spectrally similar to the corresponding spectral signature in the original image.

B. Sensitivity to Parameter Tuning

In all our simulated experiments, we have set the input noise-level map M to the same level of the noise added to the image (i.e., ground-truth noise). Lower performances are obtained when the input noise-level map differs from the actual noise level present in the image. Roughly speaking, on the one hand, when we set the input noise level to be higher than the ground-truth noise (i.e., $\hat{\sigma} > \sigma$), this means that we perform too much denoising, smoothing out some image details. On the other hand, if the input noise level is lower than the ground-truth one (i.e., $\hat{\sigma} < \sigma$), less denoising is performed, leaving some residual noise in the output image. Thus, a correct setting of the noise-level map (i.e., of the input noise level) is important to obtain optimal performance as displayed in Fig. 8, where different experiments are presented setting different noise level maps. Specifically, denoising is performed by using the same model employed in the other experiments, but changing the input noise level from $\hat{\sigma} = 5$ to $\hat{\sigma} = 100$ with an interval of 5. The ground-truth noise in the image is fixed to $\sigma_n = 50$ for all the bands, and for the evaluation, the MPSNR is chosen as the metric. Note that we achieve the best results when the input noise-level map is set to the same level of the ground-truth noise (i.e., $\hat{\sigma} = 50$). However, after analyzing the plot, one can see that it is not necessary to perfectly adjust the input noise level to achieve good performance. Indeed, our method outperforms the HSID-CNN even if we set the input noise level to a value that does not perfectly match the ground-truth noise. In this regard, as shown in Fig. 8, it is important to note that, for the considered data set, setting a higher value of the input noise-level map is better than setting a lower value of the map (with respect to the ground-truth noise).

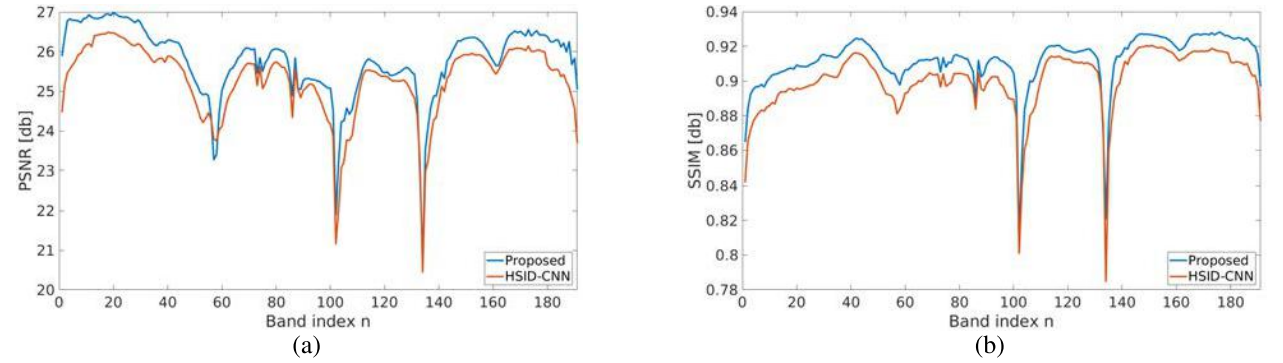


Fig. 5. Values of the different denoising methods in each band of the simulated data set with noise level $\sigma_n = 100$. (a) PSNR. (b) SSIM.

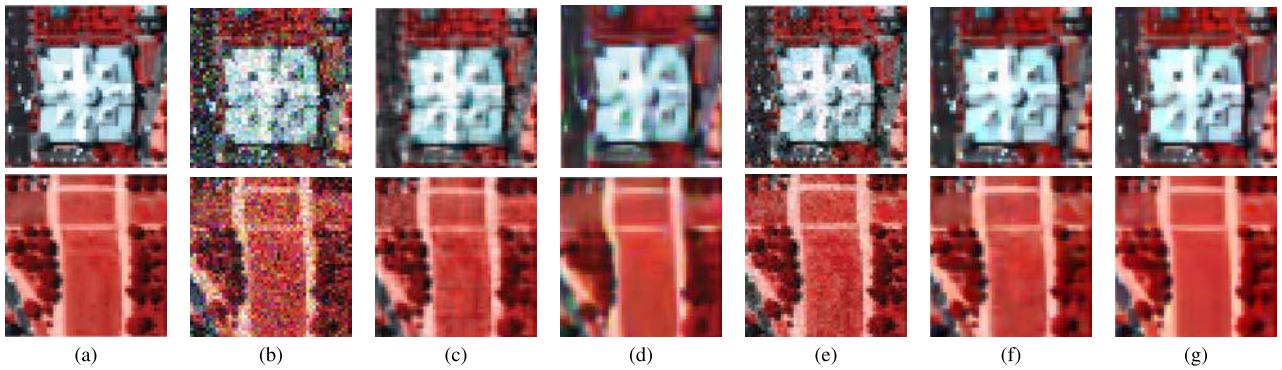


Fig. 6. Zoomed-in denoising results on the Washington DC Mall image (experiments on simulated data, with $\sigma_n = 100$). Bands 57, 27, and 17 are selected to generate false-color images. (a) Ground-truth. (b) HSSNR. (c) LRTA. (d) BM4D. (e) LRMR. (f) HSID-CNN. (g) Proposed.

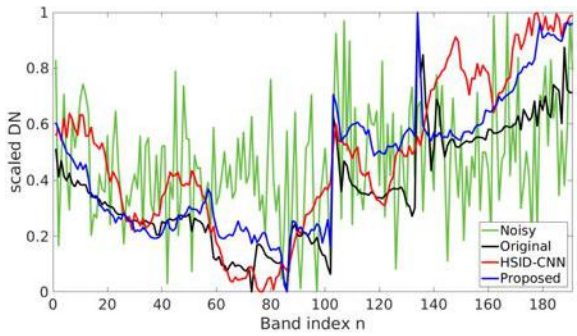


Fig. 7. Analysis of the quality of the restoration of the spectral signature at pixel (83, 175) in the original image: noisy version (green color), original signature (black color), signature obtained after applying HSID-CNN (red), and signature obtained after applying the proposed method (blue). The vertical axis (digital number) is scaled in the range [0, 1].

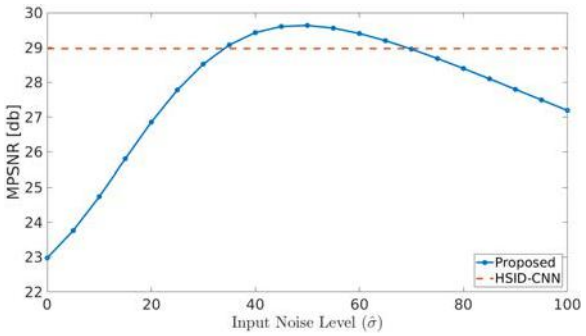


Fig. 8. MPSNR versus different input noise levels. The blue plot represents the performance of our method against different input noise levels, while the orange plot represents the results obtained with the HSID-CNN algorithm (simulated data set).

We emphasize that the input noise level is the only parameter that needs to be tuned: it allows us to perform denoising at multiple noise levels. In fact, all the results that we have obtained with HSI-SDeCNN are extracted with only one model, trained with different levels of noise from 0 to 100.

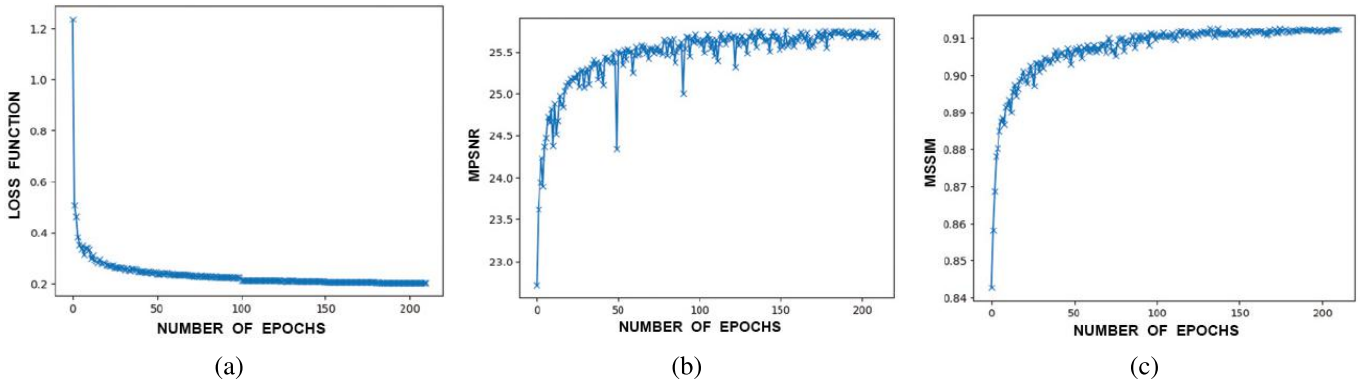
Thus, from both the qualitative and quantitative comparisons using simulated data, we can conclude that our method outperforms all the other considered methods. In the next section we discuss real HSI experiments to verify the effectiveness of our method in real scenarios, in which the noise level differs from one band to another.

C. Training Evolution

To conclude this section, we show the training evolution over 200 epochs. The validation phase of the network has been carried out at each epoch, with both input noise level (i.e., the noise-level map) and ground-truth noise level equal to 100. Fig. 9 displays the training evolution versus the number of epochs for the loss function [see Fig. 9(a)], the MPSNR [see Fig. 9(b)], and the MSSIM [see Fig. 9(c)].

V. Experimental Results: Real Data

In this section we present the results obtained on the AVIRIS Indian Pines and ROSIS University of Pavia real HSIs. We emphasize that also in this case the results have been extracted with the model trained only on the Washington DC Mall image. In order to assess the effectiveness of the proposed method with these HSIs, classification experiments are conducted (a ground-truth noise-free image is not available for these data). The quality of the denoising is measured



(a) (b) (c)

Fig. 9. Training evolution after each epoch. (a) Loss. (b) MPSNR. (c) MSSIM. The validation process has been carried out by setting both the input noise level and the ground-truth noise to 100.

by analyzing the classification accuracy before and after the denoising process. The metrics adopted are the OA and the kappa coefficient. A support vector machine (SVM) with linear kernel has been employed as a simple classifier. For the training of the classifier, we randomly selected 10% of the available labeled samples from each class, and used the remaining labeled samples for testing purposes.

Since the noise level is unknown in real HSIs, the proposed denoising algorithm has been applied by empirically setting the input noise-level map to the one that shows the best performance among the following input noise levels: $\hat{\sigma} = 5$, $25, 50, 75, 100$. For both the Indian Pines and the University of Pavia data sets, this resulted in the selection of $\hat{\sigma} = 50$.

A. Indian Pines Data Set

The Indian Pines data set is seriously degraded by Gaussian noise and impulse noise. For visual assessment purposes, we use band 2 for grayscale visualization. Fig. 10 shows the grayscale images obtained after applying the different methods.

By analyzing Fig. 10, it is possible to see that the HSSNR method leaves significant residual noise in the image, in particular dense noise and stripes. BM4D and LRMR methods, instead, exhibit superior denoising performance, but they still leave residual noise in the image (the BM4D output presents heavy strip noise, while the LRMR algorithm shows higher ability in the task of reducing this kind of noise, but still presents dense residual noise). In turn, HSID-CNN and the proposed HSI-SDeCNN exhibit much better performance, as it can be observed in the magnified region shown in Fig. 10. Indeed, both methods remove dense and strip noise without introducing any significant blur. From a visual point of view, the two methods perform similarly. However, we can note that the denoising performance of the proposed method varies from one band to another, depending on the noise level present in the specific band. Indeed, we have obtained better performance for the specific bands when the ground-truth noise level matches the input noise level map (M is fixed at noise level $\sigma = 50$ for all the bands).

In order to conduct the quantitative analysis, classification is performed on the Indian Pines data set: 16 ground-truth classes were used for testing the classification results obtained after

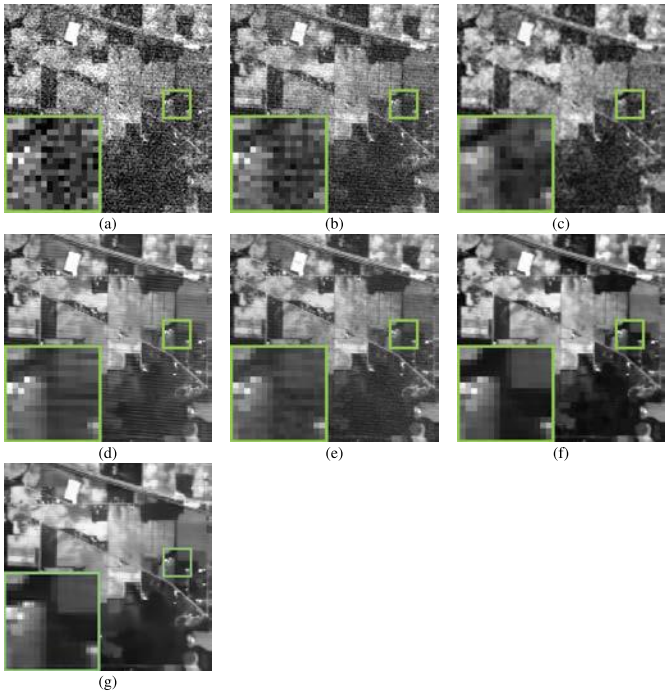


Fig. 10. Results obtained by different methods on the Indian Pines data set (grayscale visualization using band 2). (a) Noisy image. (b) HSSNR. (c) LRTA. (d) BM4D. (e) LRMR. (f) HSID-CNN. (g) Proposed.

applying the different denoising methods. The obtained results are shown in Table III. In the second column, we report the OA and kappa scores obtained with the original noisy image, and in the subsequent columns, we show the OA and kappa obtained for the HSI denoised with different methods.

On this data set, both the BM4D and the HSID-CNN algorithms obtain good performance, with an OA of 83.97% and 85.65%, respectively. Among all the compared methods, the proposed HSI-SDeCNN obtains the highest improvement, going from an OA of 75.96 (original noisy image) to an OA of 95.58% (denoised image). As a result, from a quantitative point of view, our method exhibits superior performance to those obtained by the other methods on the Indian Pines data set. This can be also appreciated in Fig. 11. In particular, Fig. 11(a) shows the ground-truth, while Fig. 11(b) shows the classification map obtained with the original noisy image.

TABLE III
CLASSIFICATION RESULTS OBTAINED AFTER DENOISING THE INDIAN PINES IMAGE USING DIFFERENT METHODS

	Original	HSSNR	LRTA	BM4D	LRMR	HSID-CNN	HSI-SDeCNN
OA	75.96%	78.78%	77.49%	83.97%	79.44%	85.65%	**95.58%**
Kappa	0.7220	0.7437	0.7387	0.8162	0.7641	0.8338	**0.9497**

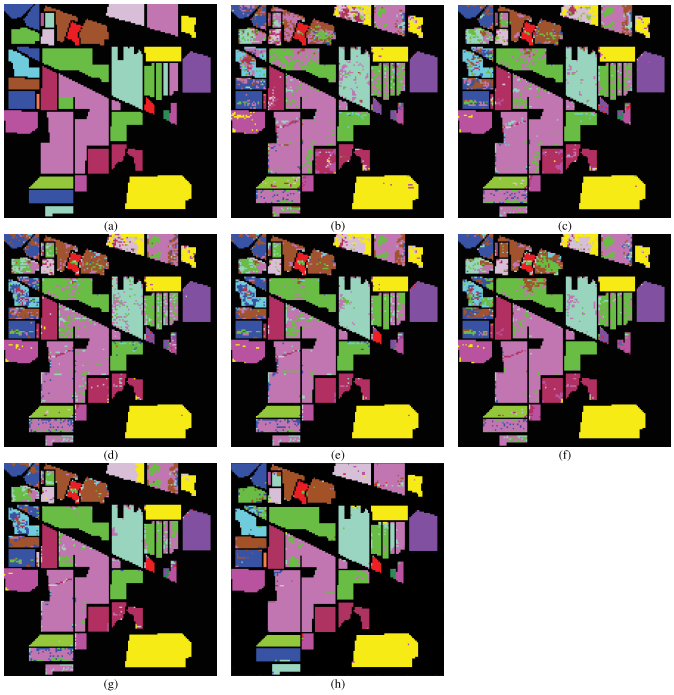


Fig. 11. Classification maps obtained on the Indian Pines scene after applying different denoising methods. (a) GT. (b) Original. (c) HSSNR. (d) LRTA. (e) BM4D. (f) LRMR. (g) HSID-CNN. (h) Proposed.

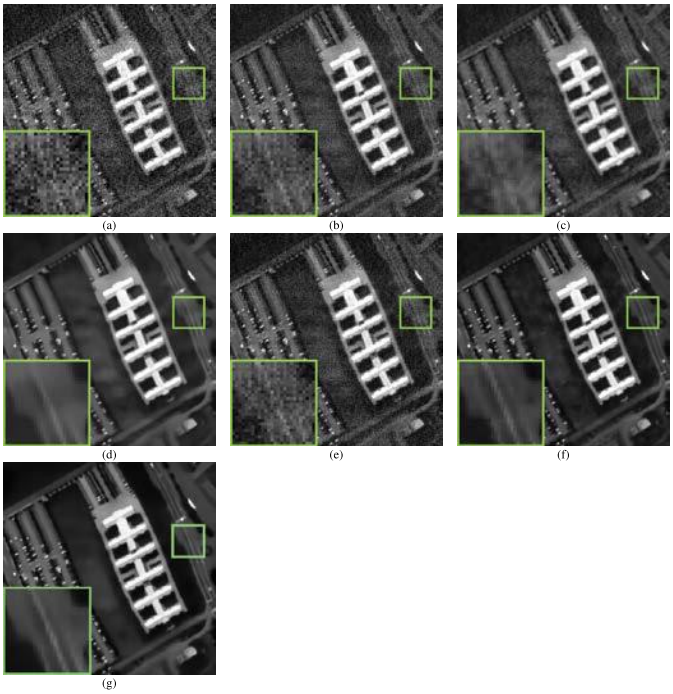


Fig. 12. Results for the University of Pavia data set (grayscale visualization using band 2). (a) Noisy. (b) HSSNR. (c) LRTA. (d) BM4D. (e) LRMR. (f) HSID-CNN. (g) Proposed.

The subsequent maps are the results of the different methods. It can be seen that our method produces a map that is less fragmented and contains many correctly classified regions that are misclassified with the images denoised by the other methods.

B. University of Pavia Data Set

In the University of Pavia data set, the noise is mainly present in the first bands. Fig. 12 shows the denoised (grayscale) results after applying different methods to band 2. On the one hand, it is possible to see that the outputs of the LRMR and HSSNR methods contain a large amount of residual noise. On the other hand, LRTA and BM4D present better denoising performance but introduce significant blurring. HSID-CNN and the proposed HSI-SDeCNN provide good results, confirming superior denoising performance. We emphasize that, for the Pavia data set, M is fixed at a noise level of $\sigma = 50$ for all the bands. To provide a quantitative analysis of this data set, classification experiments have also been conducted. The classification task is performed on nine classes, before and after denoising. Since the noise is mainly present in the first bands, the classification task has been performed only using the first 20 spectral channels. The obtained results are shown in Table IV. It is possible to see that the proposed method outperforms all the other methods. Specifically, the OA accuracy obtained with the original image is 70.09%, while the OA obtained with the image after denoising using the proposed HSI-SDeCNN is 91.74%. Furthermore, our method exhibits superior performance in terms of OA and kappa accuracy when compared to the other considered methods. Notice that the improvements obtained in this experiment are less significant than those obtained for the Indian Pines data set. This is mainly due to the fact that we are using only 20 bands from the 103 present in the University of Pavia data set. The effectiveness of our HSI-SDeCNN can be better appreciated in the classification maps shown in Fig. 13, where one can see that the proposed method obtains the most similar results to the ground-truth classification map in Fig. 13(a).

C. Computational Efficiency

In order to evaluate the computational efficiency of the proposed denoising algorithm, we compare the running time of the proposed HSI-SDeCNN with that of the HSID-CNN, which obtained the best results (in terms of running time) among the state-of-the-art considered algorithms (see results in [37]). The running time has been calculated for both experiments on simulated and real data, using the same computing

TABLE IV

CLASSIFICATION RESULTS OBTAINED AFTER DENOISING THE UNIVERSITY OF PAVIA IMAGE USING DIFFERENT METHODS

	Original	HSSNR	LRTA	BM4D	LRMR	HSID-CNN	HSI-SDeCNN
OA	70.09%	71.66%	72.56%	78.88%	83.95%	86.99%	**91.74%**
Kappa	0.6157	0.6373	0.6467	0.7302	0.8148	0.8319	**0.8972**

Fig. 13. Classification maps obtained by different methods for the University of Pavia scene after applying different denoising methods. (a) Ground-truth image. (b) Original image. (c) HSSNR. (d) LRTA. (e) BM4D. (f) LRMR. (g) HSID-CNN. (h) Proposed.

TABLE V

AVERAGE RUNTIME (IN SECONDS) MEASURED FOR THE HSID-CNN AND THE PROPOSED HSI-SDECNN METHODS

Dataset	Size	HSID-CNN	Proposed
Wash. DC Mall	$200 \times 200 \times 191$	7.2255 ± 0.0161	$\mathbf{3.0754 \pm 0.0238}$
Pavia	$200 \times 200 \times 103$	3.8997 ± 0.0184	$\mathbf{1.5886 \pm 0.0058}$

environment with MATLAB R2018b and a Laptop with GPU GTX1050Ti. Also in this case, the results provided in Table V have been averaged over ten runs. We can observe that our method is more than two times faster than the HSID-CNN, improving at the same time the denoising performance.

VI. CONCLUSION

We have presented a new learning-based method for HSI denoising, called single denoising CNN (HSI-SDeCNN). This method considers the spatial–spectral correlation present in HSIs, taking as input a full data cube instead of a single band. The main characteristics of this method are: a downsampling layer that allows the network to be faster without losing denoising performance, and a noise-level map that is used to give as input to the network an estimation of the amount of noise. The proposed method outperformed other mainstream methods commonly adopted in HSI denoising on synthetic and real data sets, with only one single trained model. In particular, it exhibits superior performance both in terms of denoising capability and computational efficiency. The performance of the method depends on the input noise level map M, that is, the only hyperparameter that needs to be tuned. This parameter, as demonstrated from the results, is flexible in handling different levels of noise.

As with any new approach, there are still some future research avenues that can be further explored. Specifically, the proposed network makes the denoising at only one level for all the bands. Such a level is specified by the input noise-level map. However, in HSIs, the noise generally differs from one band to another. For this reason, a further improvement of the method will focus on adapting the input noise level to each specific band.

ACKNOWLEDGMENT

The authors would like to thank the Associate Editor and the three anonymous reviewers for their outstanding comments and suggestions, which greatly helped them to improve the technical quality and presentation of this article.

REFERENCES

[1] R. P. Iyer, A. Raveendran, S. K. T. Bhuvana, and R. Kavitha, "Hyperspectral image analysis techniques on remote sensing," in *Proc. 3rd Int. Conf. Sens., Signal Process. Secur. (ICSSS)*, May 2017, pp. 392–396.

[2] T. Adão *et al.*, "Hyperspectral imaging: A review on UAV-based sensors, data processing and applications for agriculture and forestry," *Remote Sens.*, vol. 9, no. 11, p. 1110, 2017.

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

[4] J. Transon, R. D'Andrimont, A. Maugnard, and P. Defourny, "Survey of hyperspectral earth observation applications from space in the sentinel-2 context," *Remote Sens.*, vol. 10, p. 157, Jan. 2018.

[5] M. E. Paoletti *et al.*, "Capsule networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 4, pp. 2145–2160, Apr. 2019.

[6] R. Fernandez-Beltran, A. Plaza, J. Plaza, and F. Pla, "Hyperspectral unmixing based on dual-depth sparse probabilistic latent semantic analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 11, pp. 6344–6360, Nov. 2018.

[7] B. Yang, M. Yang, A. Plaza, L. Gao, and B. Zhang, "Dual-mode FPGA implementation of target and anomaly detection algorithms for real-time hyperspectral imaging," *IEEE J. Sel. Topics Appl. Earth Observat. Remote Sens.*, vol. 8, no. 6, pp. 2950–2961, Jun. 2015.

[8] P. Ghamisi *et al.*, "Advances in hyperspectral image and signal processing: A comprehensive overview of the state of the art," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 4, pp. 37–78, Dec. 2017.

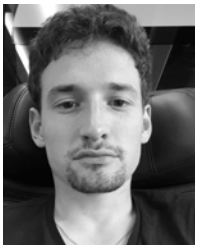
[9] B. Rasti, P. Scheunders, P. Ghamisi, G. Licciardi, and J. Chanussot, "Noise reduction in hyperspectral imagery: Overview and application," *Remote Sens.*, vol. 10, no. 3, p. 482, 2018.

[10] M. Vidal and J. Amigo, "Pre-processing of hyperspectral images. Essential steps before image analysis," *Chemometrics Intell. Lab. Syst.*, vol. 117, pp. 138–148, Aug. 2012.

[11] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, Aug. 2007, doi: 10.1109/TIP.2007.901238.

[12] S. Gu, L. Zhang, W. Zuo, and X. Feng, "Weighted nuclear norm minimization with application to image denoising," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Washington, DC, USA, Jun. 2014, pp. 2862–2869, doi: 10.1109/CVPR.2014.366.

[13] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian Denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.
[14] M. C. Motwani, M. C. Gadiya, R. C. Motwani, and F. C. Harris, Jr., "Survey of image denoising techniques," in *Proc. GSPX*, 2004, pp. 1–8.
[15] H. Othman and S.-E. Qian, "Noise reduction of hyperspectral imagery using hybrid spatial-spectral derivative-domain wavelet shrinkage," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 2, pp. 397–408, Feb. 2006.
[16] D. Letexier and S. Bourennane, "Noise removal from hyperspectral images by multidimensional filtering," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 7, pp. 2061–2069, Jul. 2008.
[17] Q. Yuan, L. Zhang, and H. Shen, "Hyperspectral image denoising employing a spectral–spatial adaptive total variation model," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, pp. 3660–3677, Oct. 2012.
[18] C. Jiang, H. Zhang, L. Zhang, H. Shen, and Q. Yuan, "Hyperspectral image denoising with a combined spatial and spectral weighted hyperspectral total variation model," *Can. J. Remote Sens.*, vol. 42, no. 1, pp. 53–72, Apr. 2016.
[19] M. Maggioni, V. Katkovnik, K. Egiazarian, and A. Foi, "Nonlocal transform-domain filter for volumetric data denoising and reconstruction," *IEEE Trans. Image Process.*, vol. 22, no. 1, pp. 119–133, Apr. 2013, doi: 10.1109/TIP.2012.2210725.
[20] T. Lu, S. Li, L. Fang, Y. Ma, and J. A. Benediktsson, "Spectral–spatial adaptive sparse representation for hyperspectral image denoising," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 1, pp. 373–385, Jan. 2016.
[21] Y.-Q. Zhao and J. Yang, "Hyperspectral image denoising via sparse representation and low-rank constraint," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 1, pp. 296–308, Jan. 2015.
[22] H. Zhang, W. He, L. Zhang, H. Shen, and Q. Yuan, "Hyperspectral image restoration using low-rank matrix recovery," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 8, pp. 4729–4743, Aug. 2014.
[23] W. He, H. Zhang, H. Shen, and L. Zhang, "Hyperspectral image denoising using local low-rank matrix recovery and global spatial–spectral total variation," *IEEE J. Sel. Topics Appl. Earth Observat. Remote Sens.*, vol. 11, no. 3, pp. 713–729, Mar. 2018.
[24] C. Li, Y. Ma, J. Huang, X. Mei, and J. Ma, "Hyperspectral image denoising using the robust low-rank tensor recovery," *J. Opt. Soc. Amer. A, Opt. Image Sci.*, vol. 32, no. 9, pp. 1604–1612, Sep. 2015.
[25] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015.
[26] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.
[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[28] A. G. Howard, "Some improvements on deep convolutional neural network based image classification," 2013, *arXiv:1312.5402*. [Online]. Available: https://arxiv.org/abs/1312.5402
[29] E. A. Smirnov, D. M. Timoshenko, and S. N. Andrianov, "Comparison of regularization methods for ImageNet classification with deep convolutional neural networks," *AASRI Procedia*, vol. 6, no. 1, pp. 89–94, 2014.
[30] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," 2013, *arXiv:1311.2901*. [Online]. Available: https://arxiv.org/abs/1311.2901
[31] Y. Li, H. Zhang, X. Xue, Y. Jiang, and Q. Shen, "Deep learning for remote sensing image classification: A survey," *Wiley Interdiscipl. Rev., Data Mining Knowl. Discovery*, vol. 8, no. 6, p. e1264, 2018.
[32] X. Yang, Y. Ye, X. Li, R. Y. K. Lau, X. Zhang, and X. Huang, "Hyperspectral image classification with deep learning models," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 9, pp. 5408–5423, Sep. 2018.
[33] 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.
[34] Y. Chen, X. Zhao, and X. Jia, "Spectral–spatial classification of hyperspectral data based on deep belief network," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 6, pp. 2381–2392, Jun. 2015.
[35] L. Mou, P. Ghamisi, and X. X. Zhu, "Deep recurrent neural networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3639–3655, Jul. 2017.
[36] B. Palsson, J. Sigurdsson, J. R. Sveinsson, and M. O. Ulfarsson, "Hyperspectral unmixing using a neural network autoencoder," *IEEE Access*, vol. 6, pp. 25646–25656, 2018.
[37] Q. Yuan, Q. Zhang, J. Li, H. Shen, and L. Zhang, "Hyperspectral image denoising employing a spatial–spectral deep residual convolutional neural network," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 2, pp. 1205–1218, Feb. 2019.
[38] K. Zhang, W. Zuo, and L. Zhang, "FFDNet: Toward a fast and flexible solution for CNN-based image denoising," *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4608–4622, Sep. 2018.
[39] D. Manolakis, R. Lockwood, and T. Cooley, "On the spectral correlation structure of hyperspectral imaging data," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, vol. 2, Jul. 2008, pp. II-581–II-584.
[40] M. Tassano, J. Delon, and T. Veit, "An analysis and implementation of the FFDNet image denoising method" *Image Process. Line*, vol. 9, pp. 1–25, Jan. 2019.
[41] M. Gharbi, G. Chaurasia, S. Paris, and F. Durand, "Deep joint demosaicking and denoising," *ACM Trans. Graph.*, vol. 35, Nov. 2016, Art. no. 191.
[42] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, 2010, pp. 807–814.
[43] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
[44] J. D. O'Sullivan, P. R. Hoy, and H. N. Rutt, "An extended spectral angle map for hyperspectral and multispectral imaging," in *Proc. Laser Appl. Photonic Appl. (CLEO)*, May 2011, pp. 1–2.
[45] J. Li, Q. Yuan, H. Shen, and L. Zhang, "Hyperspectral image recovery employing a multidimensional nonlocal total variation model," *Signal Process.*, vol. 111, pp. 230–248, Jun. 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0165168414005970
[46] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: https://arxiv.org/abs/1412.6980
[47] A. Vedaldi and K. Lenc, "MatConvNet—Convolutional neural networks for MATLAB," 2014, *arXiv:1412.4564*. [Online]. Available: https://arxiv.org/abs/1412.4564



Alessandro Maffei received the bachelor's degree in electronic and telecommunications engineering and the master's degree in information and communication engineering from the University of Trento, Trento, Italy, in 2016 and 2019, respectively.

He made his master's thesis in collaboration with the University of Extremadura, Cáceres, Spain. He is currently working with the Sector of Informatic Consulting, Reply Technology s.r.l., Italy.

Mr. Maffei received the recent 2019 Outstanding Paper Award issued from the Whispers Conference.

Juan M. Haut (S'17–M'19) received the B.Sc. and M.Sc. degrees in computer engineering from the University of Extremadura, Cáceres, Spain, in 2011 and 2014, respectively, and the Ph.D. degree in information technology through a University Teacher Training Programme from the Spanish Ministry of Education, University of Extremadura, in 2019.

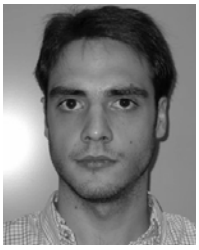
He is currently a member of the Hyperspectral Computing Laboratory, Department of Technology of Computers and Communications, University of Extremadura. His research interests include remote sensing and analysis of very high spectral resolution with the current focus on machine (deep) learning and cloud computing.

Dr. Haut received the recognition as a Best Reviewer of the IEEE GEOSCIENCE AND REMOTE SENSING LETTERS in 2018 and the Outstanding Paper Award at the Whispers 2019 Congress. He has been a Reviewer of the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING, and the IEEE GEOSCIENCE AND REMOTE SENSING LETTERS.

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 through a University Teacher Training Programme from the Spanish Ministry of Education.

She is currently a member of the Hyperspectral Computing Laboratory, Department of Technology of Computers and Communications, University of Extremadura. Her research interests include remote sensing and analysis of very high spectral resolution with the current focus on deep learning and high-performance computing.

Ms. Paoletti was a recipient of the 2019 Outstanding Paper Award at the Whispers 2019 Congress. She has been a Reviewer of the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING and the IEEE GEOSCIENCE AND REMOTE SENSING LETTERS.



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, Department of Technology of Computers and Communications, University of Extremadura. He has authored more than 150 publications, including over 50 JCR journal articles, ten book chapters, and 90 peer-reviewed conference proceeding papers. His main research interests include hyperspectral data processing and parallel computing of remote sensing data.

Dr. Plaza was a recipient of the Outstanding Ph.D. Dissertation Award at the University of Extremadura in 2008. He was also a recipient of the Best Column Award of the *IEEE Signal Processing Magazine* in 2015 and the Most Highly Cited Paper (2005–2010) in the *Journal of Parallel and Distributed Computing*. He received the best paper awards at the IEEE International Conference on Space Technology and the IEEE Symposium on Signal Processing and Information Technology. He has guest-edited four special issues on hyperspectral remote sensing for different journals. He is also an Associate Editor of the IEEE GEOSCIENCE AND REMOTE SENSING LETTERS and IEEE Remote Sensing Code Library. Additional information: http://www.umbc.edu/rssipl/people/jplaza

Lorenzo Bruzzone (S'95–M'98–SM'03–F'10) received the Laurea (M.S.) degree *(summa cum laude)* in electronic engineering and the Ph.D. degree in telecommunications from the University of Genoa, Genoa, Italy, in 1993 and 1998, respectively.

He is currently the Founder and the Director of the Remote Sensing Laboratory, Department of Information Engineering and Computer Science, University of Trento, Trento, Italy, where he is also a Full Professor of telecommunications and teaches remote sensing, radar, and digital communications. He is also the Principal Investigator of many research projects. Among the others, he is also the Principal Investigator of the Radar for Icy Moon exploration (RIME) instrument in the framework of the JUpiter ICy moons Explorer (JUICE) mission of the European Space Agency (ESA) and the Science Lead for the High Resolution Land Cover project in the framework of the Climate Change Initiative of ESA. He has authored or coauthored 247 scientific publications in refereed international journals (183 in IEEE journals), more than 310 articles in conference proceedings, and 21 book chapters. He has edited or co-edited 18 books/conference proceedings and 1 scientific book. His articles are highly cited, as proven from the total number of citations (more than 29800) and the value of the H-index (80) (source: Google Scholar). His research interests include remote sensing, radar and SAR, signal processing, machine learning, and pattern recognition. He promotes and supervises research on these topics within the frameworks of many national and international projects.

Dr. Bruzzone is currently a member of the Permanent Steering Committee of this series of workshops. He has been a member of the Administrative Committee of the IEEE Geoscience and Remote Sensing Society (GRSS) since 2009, where since 2019, where he is the Vice President for Professional Activities. Since 1998, he was a recipient of many international and national honors and awards, including the recent IEEE GRSS 2015 Outstanding Service Award, the 2017 IEEE IGARSS Symposium Prize Paper Award, and the 2018 IEEE IGARSS Symposium Prize Paper Award. He ranked first place in the Student Prize Paper Competition, 1998 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Seattle, July 1998. He is the Co-Founder of the IEEE International Workshop on the Analysis of MultiTemporal Remote-Sensing Images (MultiTemp) series. He was a Guest Co-Editor of many Special Issues of international journals. Since 2003, he has been the Chair of the SPIE Conference on Image and Signal Processing for Remote Sensing. He has been the Founder of the *IEEE Geoscience and Remote Sensing Magazine* for which he was the Editor-in-Chief from 2013 to 2017. He is currently an Associate Editor of the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING. He was invited as a Keynote Speaker in more than 32 international conferences and workshops. He has been a Distinguished Speaker of the IEEE Geoscience and Remote Sensing Society from 2012 to 2016.

Antonio Plaza (M'05–SM'07–F'15) received the M.Sc. degree and the Ph.D. degree 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, Department of Technology of Computers and Communications, University of Extremadura. He has authored more than 600 publications, including over 200 JCR journal articles (over 160 in IEEE journals), 23 book chapters, and around 300 peer-reviewed conference proceeding papers. His research interests include hyperspectral data processing and parallel computing of remote sensing data.

Dr. Plaza was a member of the Editorial Board of the *IEEE Geoscience and Remote Sensing Newsletter* from 2011 to 2012 and the IEEE GEOSCIENCE AND REMOTE SENSING MAGAZINE in 2013. He was also a member of the Steering Committee of the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING (JSTARS). He is also a fellow of IEEE for contributions to hyperspectral data processing and parallel computing of earth observation data. He received the recognition as a Best Reviewer of the IEEE GEOSCIENCE AND REMOTE SENSING LETTERS, in 2009, and the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, in 2010, for which he has served as an Associate Editor from 2007 to 2012. He was also a recipient of the Most Highly Cited Paper (2005–2010) in the *Journal of Parallel and Distributed Computing*, the 2013 Best Paper Award of the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING (JSTARS), and the Best Column Award of the *IEEE Signal Processing Magazine* in 2015. He received Best Paper Awards at the IEEE International Conference on Space Technology and the IEEE Symposium on Signal Processing and Information Technology. He has 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 IEEE GRSS from 2012 to 2016. He has reviewed more than 500 manuscripts for over 50 different journals. He has served as the Editor-in-Chief of the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING from 2013 to 2017. He has guest-edited ten special issues on hyperspectral remote sensing for different journals. He is also an Associate Editor of IEEE ACCESS (received the recognition as an Outstanding Associate Editor of the journal in 2017). Additional information: http://www.umbc.edu/rssipl/people/aplaza