

Received August 23, 2020, accepted September 19, 2020, date of publication September 29, 2020, date of current version October 12, 2020. Digital Object Identifier 10.1109/ACCESS.2020.3027776

# **Rotation Equivariant Convolutional Neural Networks for Hyperspectral Image Classification**

## MERCEDES E. PAOLETTI<sup>®1</sup>, (Member, IEEE), JUAN M. HAUT<sup>®1</sup>, (Senior Member, IEEE), SWALPA KUMAR ROY<sup>©2</sup>, (Student Member, IEEE), AND ELIGIUS M. T. HENDRIX<sup>3</sup>

<sup>1</sup>Hyperspectral Computing Laboratory, Department of Technology of Computers and Communications, Escuela Politécnica, Universidad de Extremadura, 10003 Caceres, Spain

<sup>2</sup>Department of Computer Science and Engineering, Jalpaiguri Government Engineering College, Jalpaiguri 735102, India

<sup>3</sup>Department of Computer Architecture, Universidad de Málaga, 29016 Malaga, Spain

Corresponding author: Mercedes E. Paoletti (mpaoletti@unex.es)

This work was supported in part by the Spanish Ministry under Grant RTI2018-095993-B-100, and in part by the European Regional Development Fund (ERDF).

**ABSTRACT** Detection of surface material based on hyperspectral imaging (HSI) analysis is an important and challenging task in remote sensing. It is widely known that spectral-spatial data exploitation performs better than traditional spectral pixel-wise procedures. Nowadays, convolutional neural networks (CNNs) have shown to be a powerful deep learning (DL) technique due their strong feature extraction ability. CNNs not only combine spectral-spatial information in a natural way, but have also shown to be able to learn translation-equivariant representations, i.e. a translation of input features into an equivalent internal CNN feature map. This provides great robustness to spatial feature locations. However, as far as we know, CNNs do not exhibit a natural way to exploit rotation equivariance, i.e. make use of the fact that data patches in a HSI data cube are observed in different orientations due to their orientation or on the varying paths/orbits of the airborne/spaceborne spectrometers. This article presents a rotation-equivariant CNN2D model for HSI analysis, where traditional convolution kernels have been replaced by circular harmonic filters (CHFs). The obtained results over three well-known HSI datasets showcase the potential of the approach.

**INDEX TERMS** Hyperspectral imaging (HSI), convolutional neural network (CNN), harmonic network (H-net), rotation invariance.

#### I. INTRODUCTION

Imaging spectrometry data, in general, and hyperspectral imaging (HSI) in particular [1] are widely used remote sensing technologies for a large variety of applications due their abundant information about earth-surface material, such as: natural resources management [2], for instance in those activities related to forestry [3]-[5], geology/mineralogy [6]-[8] or hydrology [9]–[11]; precision agriculture [12]–[14], soil degradation [15]-[17] and crop stress/disease detection [18]–[20]; urban planning [21]–[23]; risk prevention [24]-[26], and disaster monitoring [27]-[29], among others. This information is gathered by operational HSI systems (aerial and satellite-based imaging spectrometers) [30], [31] in hundreds of channels that collect the surface solar reflectance by measuring the electromagnetic spectrum

The associate editor coordinating the review of this manuscript and approving it for publication was Yongqiang Zhao<sup>10</sup>.

between the visible spectrum and the short-wave infrared with a very narrow bandwidth. As a result, a huge 3D image cube is obtained, where pixels represent narrow and continuous spectral signatures, which are unique for each captured land-cover material. This interesting peculiarity makes HSI data a powerful tool for target detection and land-cover categorization tasks.

## A. TRADITIONAL MACHINE LEARNING METHODS FOR SPECTRAL-SPATIAL HSI PROCESSING

Traditionally, HSI information has been exploited in machine learning (ML) by pixel-wise methods which consider HSI data as a list of spectral vectors, assuming that each pixel is pure and typically labeled as a single land cover type [32]–[34]. In the current literature, there are abundant pixel-wise methods, such as the popular support vector machines (SVMs) [35], [36], K-nearest neighbor (KNN) [37], [38], multinomial logistic regression (MLR) [39], [40],

random forests (RFs) [41], [42] and standard artificial neural networks (ANNs) [43], among others. As these methods conduct only spectral processing to assign each pixel to its corresponding land-cover class, their reliability (in terms of classification accuracy) is strongly related to the number and spectral-quality of training samples. However, HSI scenes usually exhibit a low spatial resolution. For instance the AVIRIS instrument [30] has a ground sample distance of 20 mpp. Therefore, a pixel collects the reflectance of a very large area, which may contain different materials, such that the captured spectral signature will be very mixed. Moreover, uncontrolled changes (such as atmospheric conditions), introduce great variability in the spectral domain. In this sense, pixel-wise classification methods suffer a high intra-class variability (for instance, class-centered pixels and frontier pixels that belong to the same class), and also the high inter-class similarity (frontier pixels that belong to different classes). Furthermore, the so-called "salt and pepper" noise problem appears frequently, since these methods ignore the spatial dependencies between neighboring pixels [44].

Recent investigation has shown that taking spatial information into account in pixel-wise HSI processing may improve classification performance [45]. This is based on the general assumption that adjacent pixels commonly belong to the same class, such that exploiting their information may reduce intra-class variance and labeling uncertainty due to the characterization of the contextual features. Literature discusses several ML techniques to perform spectral-spatial HSI classification [46]. For instance, Tarabalka *et al.* [47] and Zhang *et al.* [48] combined the SVM with Markov random field (MRF) modelling to process spectral and contextual information, reaching promising results. Also, Li *et al.* [49] and Sun *et al.* [50] applied MRFs to refine the classification results obtained by the MLR pixel-wise classifier.

Other spectral-spatial methods take advantage of the texture features, for instance Huo and Tang [51], and Seifi Majdar and Ghassemian [52] used Gabor filtering and SVMs to extract texture information from HSI data. Moreover, He et al. [53] presented the improved discriminative low-rank Gabor filtering (DLRGF) to extract suitable spectral-spatial features of the HSI data. Many other image filtering methods have been successfully applied for spectral-spatial HSI processing, for instance Kang et al. [54] introduced edge-preserving filtering (EPF) to refine the SVM classification map. Guo et al. [55] applied guided filtering (GF) to the KNN classifier, while Wang et al. [56] combined GF, principle component analysis (PCA) and deep neural networks to extract discriminative multi-features from HSI scenes. In [57], Liao and Wang fused two spatial-based filters, in particular curvature (CF) and domain transform recursive (DTRF) filtering to enhance the performance of HSI classification, and in [58] they implemented an adaptive manifold filter with spatial correlation feature (AMSCF) approach.

Morphological profiles (MPs) [59], [60] have also been successfully applied to extract relevant spatial information

179576

from the HSI data cube. For instance, Benediktsson et al. [61] built one MP (by applying a set of openings and closings) for each principal component extracted from the HSI data, and used all together in one extended MP (EMP), applying an ANN to perform the final classification. In [62], Plaza et al. applied multiscale sequences of EMP transformations to extract spatial relations within HSI data, while in [63] the authors proposed a new multi-channel MP approach with SVM. Fauvel et al. [64] combined morphological and spectral information within a SVM classifier, and Huang et al. [65] conducted an extensive study about several strategies for producing the base images for MPs, among traditional PCA. As an extension of MPs, attribute profiles (APs), extended APs (EAPs) and extended multi-attribute profiles (EMAPs) have been considered as powerful tools to better model the spatial information contained within the HSI data cube in comparison to conventional MP-based techniques, while reducing the computational complexity of MPs. In this context, Dalla et al. [66] enhanced their obtained accuracy results when conducting HSI classification through EAPs and EMAPs, while Falco et al. [67] conducted an independent component analysis (ICA) spectral analysis with reduced APs, in order to avoid the spatial information redundancy.

In recent years, the use of local binary patterns (LBPs) [68] has shown good spatial-classification performance in HSI classification. For instance, Li *et al.* [69] extracted spatial LBPs information and combined it with the spectral information through an extreme learning machine (ELM) classifier [70], [71]. Recently, Ye *et al.* [72] developed segmented LBPs in order to address the limitations of the original approach in the representation of physical meaning due to the existing noise bands, while preserving the intrinsic geometrical structure of original data. And Sidike *et al.* [73] used the interesting multi-scale completed LBP (CLBP) to enhance the spatial-processing of HSI scenes.

Image segmentation is quite interesting in spatial processing methods, where they split the HSI scene into non-overlapping homogeneous regions according to some partitioning criteria. In this sense, Tarabalka *et al.* [74] performed spectral-spatial HSI classification combining pixel-wise SVM classification results and the partitional clustering segmentation map. Furthermore, in [75], the authors combined the SVM classification results with watershed segmentation. Zhang *et al.* [76] adopted an active learning strategy with hierarchical segmentation to perform spectral-spatial HSI analysis and pixel classification. Also, Tarabalka *et al.* [77] proposed a multiple spectral–spatial classification (MSSC) scheme, which combines several pixel-wise and segmentation-based classifiers to process the HSI scene.

Among these approaches, many kernel-based techniques have been developed to perform spectral-spatial HSI classification [78]. For instance Camps-Valls *et al.* [79] presented composite kernels (CKs) as an efficient and flexible spectral-spatial processing framework, balancing between the spatial and spectral information comprised by HSI data. Li *et al.* [80] extended CKs by avoiding weight parameters (generalized composite kernel machines –G-CKs–), reaching promising results with a MLR classifier. Also, Camps-Valls *et al.* [81] generalized the CK behaviour by graph kernels (GKs), which compute spatial similarities in a proper feature space.

Despite their notable results (in terms of accuracy), these standard ML-based techniques have to face several challenges. In particular, their performance strongly depends on the parameters involved in the algorithm, the selected classification criteria or discriminative measurement, and/or some handcrafted features, which are very hard to obtain and are not flexible. In this sense, the success of these methods hinges largely on the user/programmer knowledge, involving a high computational burden (some of them) and being quite difficult to train.

## B. DEEP LEARNING MODELS FOR SPECTRAL-SPATIAL HSI PROCESSING

In contrast to previous ML-based methods, deep learning (DL) models [82], [83] offer a bunch of quite interesting neural architectures to automatically process the spectral-spatial information contained in HSI data, without requiring a priori knowledge about the data distribution/features. In this context, deep neural networks (DNNs) can be interpreted as hierarchical stacks of L operational layers, where each layer l applies a transformation function,  $\mathbf{x}^{(l)} = f^{(l)}(\mathbf{x}^{(l-1)}, \mathbf{P}^{(l)})$  over its inputs  $\mathbf{x}^{(l-1)}$  to obtain a certain output data representation  $\mathbf{x}^{(l)}$ , which will depend on the layer parameters  $\mathbf{P}^{(l)}$  (weights and biases). By adjusting their parameters, DNNs aim at automatically extracting those representations that best fit the problem, in order to optimize the architecture and achieve an adequate accuracy. This behaviour, combined with a large variety of architectures and their great versatility (DNNs are considered universal approximators), makes DNNs a pretty powerful tool for the analysis of complex non-linear data, exhibiting a greater ability of feature abstraction than standard ML models and shallow architectures.

Some HSI literature has tackled the spectral-spatial processing through deep architectures. For instance, Chen et al. [84] developed a stacked autoencoder (SAE) to perform deep spectral-spatial feature extraction and classification, concatenating several flattened principal components from a spatial pixel neighborhood with the corresponding spectral features of the pixel. Following the same procedure, Chen et al. introduced deep belief networks (DBN) [85] to extract high-level spectral-spatial features from the HSI data cube. These models overcome the limitations expressed by traditional ML models, as they automatically extract the most relevant spectral-spatial features when classifying HSI data. However, these models are originally based on fully-connected layers, which only process vectorized data, i.e. 1-dimensional arrays. Consequently, the spatial representations must be flattened and vectorized in order to be processed by these architectures, i.e. spatial relations (in terms of feature locations) are lost.

Among these methods, convolutional neural networks (CNNs) [86] appear to be a promising DL technique within image processing tasks, in general, and HSI data analysis in particular [87]. The corresponding model offers a natural way to combine spectral-spatial information due to its hierarchy of convolution filters and pooling operations, providing a flexible mechanism to process *n*-dimensional arrays. As a result, the CNN model has shown a good performance in HSI data classification [88]. Although CNNs are excellent tools for feature extraction, without enough training samples, they exhibit the typical deep learning limitation of overfitting. There are two main reasons: on the one hand, there are not enough samples to cover all the variability of the HSI scene, and on the other hand, the parameters are over-adjusted to the training data distribution. In this context, despite the great performance during the training stage, the behavior of the network deteriorates significantly during the inference phase due to its over-adjusted parameters and the lack of knowledge about unseen samples. Moreover, they are not really effective in exploring some spatial relations among features and their generalization power depends on the transformations present in the training data.

## C. SPATIAL TRANSFORMATIONS EQUIVARIANCE OF CNN MODEL FOR HSI CLASSIFICATION

In this context, affine transformations such as shifting, scaling, translations and rotations can appear within the data, where objects that belong to the same class appear at different locations, with different sizes and orientations. In practical HSI data, scenarios can be observed with a varying orientation due to the capturing spectrometer position, the altitude at which it is located and the path followed by the airborne spectrometer. From this perspective, achieving a certain invariance/equivariance to these transformations can be quite useful for classifying HSI data, making classification models more reliable, improving their generalization ability and reducing the number of training parameters. Eq. (1) provides the mathematical formulation of transformationinvariance/equivariance, where  $f : \mathcal{X} \to \mathcal{Y}$  defines the feature extraction function, which is invariant to any transformation  $g(\cdot)$  of the multidimensional input data **X** (which belongs to a group of transformations  $\mathcal{G}$ , i.e.  $g \in \mathcal{G}$ ) if the output of  $f(\cdot)$  remain unchanged. On the contrary,  $f(\cdot)$  is equivariant to every  $g \in \mathcal{G}$  if there is another group  $\mathcal{H}$  such that, for every  $g(\cdot) \in \mathcal{G}$  there exists a  $h(\cdot) \in \mathcal{H}$  that reflects the transformation. Therefore, the concept invariant means that there is no variation at all, while equivariant implies a variation in a similar or equivalent proportion.

invariance 
$$\rightarrow f(g[\mathbf{X}]) = f(\mathbf{X})$$
  
equivariance  $\rightarrow f(g[\mathbf{X}]) = h[f(\mathbf{X})]$  (1)

Moreover, becoming robust to rotation transformations is a quite important topic within ML and DL due to the high



**FIGURE 1.** CNN translation (left) and rotation (right) equivariance. In the first case, the translation  $g(\cdot)$  of the HSI scene X is equivalent to the translation  $h(\cdot)$  of the feature maps, i.e. the translation is preserved by each layer. However, the rotation of the input X by  $g(\cdot)$  has not a equivalent transformation  $h(\cdot)$ , which could produce the same result if it is applied over the feature maps, and if an  $h(\cdot)$  exits, it is very hard to find.

complexity and the negative effects it introduces during data processing [68], [89]–[96]. Some interesting efforts in remote sensing HSI data analysis can be found in [97]–[100]. However, these approaches are usually based on traditional ML techniques and feature extractors such as SIFT. Focusing on deep CNN models, convolution kernels capture translations within the data, i.e. they are translation-equivariant due the particular form of parameter sharing, while pooling operations provide some invariance to small translations [83]. However, the CNN model is not naturally equivariant to rotation transformations. In fact, rotations in the inputs do not generate an equally logical-response in the outputs [94], as we can observe in Fig. 1. Therefore, other mechanisms have to be introduced in order to aid these transformations.

Traditionally, spatial regularization [101] and data augmenting techniques [102]–[104] have been developed as reinforcement to help CNN learning, using normally a large number of rotated data variations. However, this procedure works as a black box, where the learning mechanism does not exploit the characteristics of the problem [94]. Moreover, these techniques do not ensure local equivariances within the CNN model, which is compelled to learn several copies of the same filter to fit different feature rotations, increasing both the number of redundant parameters and the learning complexity.

Several methods have been developed for computer vision, optical image and remote sensing processing to include rotation-equivariance within CNN models, which can be classified into those that 1) rotate the inputs (input data or feature maps) to generate rotation invariant representations [91], [105] and 2) rotate the CNN filters or activation [106]. For instance, Fasel and Gatica-Perez [91] rotated the input data before sending it into several stacked CNNs, generating rotation invariant representations through pooling. Cohen and Welling [106] generalized the convolution layer behaviour

to perform rotations and reflections equivariances by rotating kernels within each layer.

In this context, this article tackles the rotation-equivariance within CNNs based on 2D kernel for spectral-spatial HSI data processing and classification. In particular:

- Instead of achieving rotation-equivariance by transforming the inputs, we propose to transform the standard convolution filters of the network into steerable filters [107]. In this way, we develop a new rotation-equivariant CNN2D architecture for remote sensing HSI data classification to deal with all local rotation information present in HSI data.
- Therefore, our deep model uses as structural element a kind of steerable filter, named circular harmonic filter (CHF) [94] to represent all rotated versions of a filter (i.e., 360°-rotation equivariance) by considering a linear combination of a number of steering bases.
- In this way, the new architecture aims at reducing the high redundancy in the filters learned by the CNN, simplifying the learning process and improving the performance when conducting HSI data classification.
- Obtained results over three well-know HSI datasets demonstrate the potential of the approach.

The remainder of the paper is organized as follows. Section II describes the used methodology of the rotation-equivariance CNN model. Section III compares its performance with baseline CNNs over three widely-used HSI data set instances. Finally, Section IV concludes the paper with some remarks and possible future research lines.

## **II. METHODOLOGY**

## A. THE CONTEXT OF THE CNN MODEL

A CNN architecture can be divided into two main parts: i) the feature extractor (FE), which is usually composed of one or more convolution layers, followed by data normalization, non-linearity function and, often, by a subsampling or pooling layer, which obtains high level representations (feature maps) of the original input data, and ii) the classification net, composed of several fully-connected layers.

The design of each FE-stage is intrinsically related to the input data structure, i.e., the dimensional arrangement of the input array. In this context, FE-stage layers can process *n*-dimensional arrays, taking the maximum advantage of the information contained in the data. Thus, considering a standard CNN model with *L* FE-stages, let  $\mathbf{X}^{(l-1)} \in \mathbb{R}^{N^{(l-1)} \times N^{(l-1)} \times F^{(l-1)}}$  denote the input volume of the *l*-th layer  $(l \in \{1, \ldots, L\})$ , which is composed of  $N^{(l-1)} \times N^{(l-1)}$ features and  $F^{(l-1)}$  channels.<sup>1</sup> Therefore, the *l*-th FE-stage will apply its corresponding FE-function  $f^{(l)}(\cdot)$  to  $\mathbf{X}^{(l-1)}$  in order to obtain the output volume  $\mathbf{X}^{(l)} \in \mathbb{R}^{N^{(l)} \times N^{(l)} \times F^{(l)}}$ . As mentioned before, we suppose  $f^{(l)}(\cdot)$  is composed of convolution  $f^{(l)}_{C}(\cdot)$ , normalization  $f^{(l)}_{B}(\cdot)$ , activation  $f^{(l)}_{H}(\cdot)$  and pooling  $f^{(l)}_{P}(\cdot)$  layers, so we can reinterpret each FE-stage as [108]:

$$\mathbf{X}^{(l)} = f^{(l)} \left( \mathbf{X}^{(l-1)}, \mathbf{P}^{(l)} \right)$$
  
=  $f_P^{(l)} \left( f_H^{(l)} \left( f_B^{(l)} \left( f_C^{(l)} \left( \mathbf{X}^{(l-1)}, \mathbf{P}^{(l)} \right) \right) \right) \right)$  (2)

where  $\mathbf{P}^{(l)}$  represents the learnable kernel parameters (i.e., weights  $\mathbf{W}^{(l)}$  and biases  $\mathbf{b}^{(l)}$ ). Each sub-mapping function is defined as follows:

$$\mathbf{Z}^{(l)} = f_C^{(l)} \left( \mathbf{X}^{(l-1)}, \mathbf{W}^{(l)}, \mathbf{b}^{(l)} \right)$$
(3a)

$$\hat{\mathbf{Z}}^{(l)} = f_B^{(l)} \left( \mathbf{Z}^{(l)} \right) \tag{3b}$$

$$\tilde{\mathbf{Z}}^{(l)} = f_H^{(l)} \left( \hat{\mathbf{Z}}^{(l)} \right) \tag{3c}$$

$$\mathbf{X}^{(l)} = f_P^{(l)} \left( \tilde{\mathbf{Z}}^{(l)} \right) \tag{3d}$$

Eq. (3a) defines the convolution layer operation, which is composed of  $F^{(l)}$  filters that are applied to small regions of  $\mathbf{X}^{(l-1)}$ . This region is defined by the local receptive field  $K^{(l)} \times K^{(l)} \times F^{(l-1)}$ . In this sense, the corresponding weights  $\mathbf{W} \in \mathbb{R}^{K^{(l)} \times K^{(l)} \times F^{(l-1)} \times F^{(l)}}$  are applied along the input like a sliding window, where each kernel application combines the features contained in the receptive field along the channels. Eq. (4) details the mathematical expression of  $f_C^{(l)}$  given by Eq. (3a). As we can observe, the convolution (\*) between the weights  $\mathbf{W}^{(l)}$  and the data  $\mathbf{X}^{(l-1)}$  plus the bias vector  $\mathbf{b}^{(l)}$  obtains the output feature volume  $\mathbf{Z}^{(l)} \in \mathbb{R}^{N^{(l)} \times N^{(l)} \times F^{(l)}}$ , where the (i, j) feature in channel t, i.e.  $z_{i,j,t}^{(l)}$  is obtained after the affine transformation [109] between the corresponding weights and input data, where  $i \in \{1, \ldots, N^{(l)}\}$  and  $j \in \{1, \ldots, N^{(l)}\}$  index along spatial dimensions of the output and input arrays,  $\hat{i} \in \{1, \ldots, K^{(l)}\}$  and  $\hat{j} \in \{1, \ldots, K^{(l)}\}$   $t \in \{1, \ldots, F^{(l)}\}$  indexes along the spectral dimension of the convolution kernel and output volume, and  $\tilde{t} \in \{1, \ldots, F^{(l-1)}\}$  indexes along the spectral dimension of the input volume.

$$\mathbf{Z}^{(l)} = \mathbf{W}^{(l)} * \mathbf{X}^{(l-1)} + \mathbf{b}^{(l)}$$
  
$$z^{(l)}_{i,j,t} = \sum_{\hat{i},\hat{j},\tilde{t}} w^{(l)}_{\hat{i},\hat{j},\tilde{t},t} x^{(l-1)}_{i+\hat{i},j+\hat{j},\tilde{t}} + b^{(l)}_{t}$$
(4)

After convolution  $f_C^{(l)}$ , the data is usually normalized by the batch normalization function  $f_B^{(l)}$  in order to recover its original distribution. This limits the internal covariate shift effect introduced by  $f_C^{(l)}$  by forcing the resulting output distribution to have mean  $\mu = 0$  and variance  $\sigma^2 = 1$  [110], while making the landscape of the optimizer more smooth [111]. Eq. (5) details the mathematical terms behind Eq. (3b), where the step-size  $\epsilon$  provides certain numerical stability and learnable parameters  $\gamma$  and  $\beta$  control the distribution  $\mathcal{N}(\mu, \sigma^2)$ , making it more flexible.

$$\hat{\mathbf{Z}}^{(l)} = \frac{\mathbf{Z}^{(l)} - \operatorname{mean}\left[\mathbf{Z}^{(l)}\right]}{\sqrt{\operatorname{var}\left[\mathbf{Z}^{(l)}\right] + \epsilon}} \cdot \gamma + \beta$$
(5)

Obtained  $\hat{\mathbf{Z}}^{(l)} \sim \mathcal{N}(\mu \approx 0, \sigma^2 \approx 1) \in \mathbb{R}^{N^{(l)} \times N^{(l)} \times F^{(l)}}$ is then processed by a non-linear activation function  $f_H^{(l)}$ to extract the nonlinear representations from the data. In this sense,  $f_H^{(l)}$  is interpreted as the detector unit of the FE-stage [83]. Indeed, these activations indicate the presence of the features identified by the convolution kernel. Usually, the rectified linear unit (ReLU) [112]–[114] is implemented, so  $\tilde{z}_{i,j,t}^{(l)} = \max\left(0, \hat{z}_{i,j,t}^{(l)}\right), \forall i, j \in \{1, \ldots, N^{(l)}\}$  and  $\forall t \in \{1, \ldots, F^{(l)}\}$ .

Following the activation function, a sample-based discretization process  $f_P^{(l)}$  is conducted by adopting a downsampling strategy. In this sense, the pooling step summarizes the spatial information of each feature map in  $\tilde{\mathbf{Z}}^{(l)}$ , retaining the most relevant features by applying a max, average or sum operation within a neighborhood window [115]. For instance, the max-pooling operator selects the largest feature from the rectified volume  $\tilde{\mathbf{Z}}^{(l)}$ , considering a spatial window of  $P \times P$ . As a result, pooling summarizes  $\tilde{\mathbf{Z}}^{(l)}$  into  $\tilde{\mathbf{X}}^{(l)}$  by dividing the spatial dimensions into  $N^{(l)}/P$  parts. This reduces the number of parameters, which helps to control the overfitting problem.

#### **B. LIMITATIONS OF CONVOLUTION KERNELS**

As mentioned before, the *l*-th convolution kernel comprises  $F^{(l)}$  learnable filters with size  $K^{(l)} \times K^{(l)}$ , which are applied in a "sliding window" fashion across the entire layer input  $\mathbf{X}^{(l-1)}$ . In this regard, the *weight sharing* mechanism provides an easy way to reuse the same  $K^{(l)} \times K^{(l)}$  weights given by filter *t* across the entire input,  $\forall t \in \{1, \dots, F^{(l)}\}$ . This mechanism significantly reduces the number of weights that have to be learned. In addition, the filter is automatically adapted to detect a particular feature, such as a combination of edges. Therefore, if there are similar features within the

<sup>&</sup>lt;sup>1</sup>Focusing on HSI processing, the first FE-stage l = 1 will receive the original HSI patch  $\mathbf{X} \in \mathbb{R}^{N \times N \times F}$ , composed of  $N \times N$  vector pixels, where pixel (i, j) contains the reflectance measurements within F spectral bands,  $\mathbf{x}_{i,j} = [x_{i,j,1}, \dots, x_{i,j,F}] \in \mathbb{R}^{F}$ 



FIGURE 2. CNN's max-pooling behaviour when translation and rotation is introduced. In the first row we can observe the original patch, whereas in the second, third and fourth rows we can observe the main feature only translated, only rotated 90°, and both rotated and translated, respectively. Depending on the transformations (first, second rows), max-pool obtains similar results, in the sense that the most predominant element always appears in the upper left corner, although local translation and rotation are lost. However, with bigger spatial transformations, the feature map changes significantly.

input data, the filter will detect them independently of their location in the inputs, recording their position in the output volume and preserving the translation at each layer [106]. Thus, a convolution kernel is translation-equivariant, as the spatial translation in inputs generates a consequent translation in outputs [83], following Eq. (1):

$$\mathbf{Z}^{(l)} = f_C^{(l)} \left( g \left[ \mathbf{X}^{(l-1)} \right] \right) = h \left[ f_C^{(l)} \left( \mathbf{X}^{(l-1)} \right) \right]$$
(6)

However, low-level features, such as borders and edges can appear with different orientations. This may affect the CNN performance dramatically, because filters cannot fit the data properly, as has been observed in filterbanks visualized by Zeiler and Fergus [116]. In this sense, the CNN is compelled to learn the rotated version of the same filter, investing a significant amount of computation to learn all these redundant weights, which are actually copies of one, which is rotated in different ways [94].

This behavior should not be misunderstood with the invariance offered by pooling layers. In this sense, the pooling layer helps the network model to obtain a data representation approximately invariant to small translations and rotations of the input, as we observe in Fig. 2.

However, none of these layers is naturally equivariant to rotation. Focusing on convolution layers, if  $\mathbf{X}^{(l-1)}$  is rotated, the feature vectors obtained by  $f_C^{(l)}$  do not rotate in an easily predictable way [94], while in pooling layers, the rotation equivariance is not skipped at all. In fact, it has to be approximated by small translations, normally requiring data augmentation (i.e. including in the training set data with every possible rotation) and leading to high sample complexity,

while CNN models are highly sensitive to little input data perturbations [117].

Encoding these properties into the neural network architecture can enhance data understanding, improving the learning procedure and reducing the number of parameters [118].

## C. CIRCULAR HARMONIC FILTERS

In order to encode the 360°-rotation equivariance within the CNN for remote sensing HSI data classification, the developed spectral-spatial model uses CHF [94] as the structural element to perform the FE stage. The CHF belongs to the steerable filter family [107], [119], which defines a set of orientation-selective convolution kernels that can be constructed at any rotation by simply using a a linear combination of a finite number of appropriate base filters [120]. Within a continuous Euclidean space, the CHF is defined by a sinusoidal angular part multiplied with a radial function [119]

$$W_m(r,\phi;R,\beta) = R(r)e^{i(m\phi+\beta)},\tag{7}$$

where  $(r, \phi)$  denotes the polar coordinates of feature maps  $(r \ge 0$  is the radial distance and  $\phi \in [0, 2\pi)$  is the horizontal plane angle),  $R : \mathbb{R}^+ \to \mathbb{R}$  is the radial profile function to control the filter overall shape, and  $\beta \in [0, 2\pi)$  is a phase offset term that gives the filter orientation-selectivity [94]. In particular, R and  $\beta$  must be learned during the training step. An interesting parameter is the rotation order  $m \in \mathbb{Z}$ , which limits the kind of transformation applied to the data. Hereof, following Eq. (6), we can consider  $g^{\theta} \in \mathcal{G}$  as the  $\theta$ -degree rotation<sup>2</sup> of the input  $\mathbf{X}(r, \phi)$  (in polar coordinates, superscripts have been avoided to facilitate the understanding of mathematical concepts), so  $\mathbf{X}(r, g^{\theta}[\phi]) = \mathbf{X}(r, \phi - \theta)$ . Thereby, the application of the CHF on the rotated input must be equivalent to the application of the same filter on the original data, considering  $h^{\theta} = e^{im\theta}$ , i.e.

$$\begin{bmatrix} \mathbf{W}_m(r,\phi;R,\beta) \star \mathbf{X}(r,\phi-\theta) \end{bmatrix}$$
  
=  $e^{im\theta} \begin{bmatrix} \mathbf{W}_m(r,\phi;R,\beta) \star \mathbf{X}(r,\phi) \end{bmatrix}$ , (8)

where ( $\star$ ) indicates the cross-correlation operation.<sup>3</sup> Fig. 3 provides a graphical representation of Eq. (8). We can observe that, for instance, if m = 0, then  $h^{\theta} = e^{i \cdot 0\theta} = 1$ , defining the invariance, while if m = 1, then  $h^{\theta} = e^{i \cdot 1\theta}$ , defining linear equivariance (i.e. 1-equivariant output).

Taking back the superscripts to indicate the FE stage in which we are, and considering the CNN as a deep stack of CHFs, it must be noted that the cross-correlation between the feature maps  $\mathbf{X}_{m_i}^{(l-1)}$  (with rotation order  $m_i$ ) and a filter  $\mathbf{W}_{m_i}^{(l)}$ 

 $<sup>^2 \</sup>rm We$  follow the counter-clockwise rotation, where positive angles are measured counterclockwise and negative angles go clockwise.

 $<sup>^{3}</sup>$ Cross-correlation (\*) and convolution (\*) are quite similar image-operations where the first one slides a kernel across the image –this operation is much closer to the concept of convolution that we have within the CNN, given by Eq. (4)– and the second one slides a flipped kernel across the image. Actually, many DL frameworks are implementing CNN models with cross-correlation, such as *TensorFlow* [121] or *Pytorch* [122] which learn the same weight values but in a flipped orientation, in order to avoid the flipping operation, which adds some unnecessary complexity to the CNN.



**FIGURE 3.** Rotation equivariance with CHF. In both cases, output data is obtained into the feature space as the cross-correlation between the input data and the CHF. The rotated and unrotated output volumes are related by the associated transformation  $h^{\theta} = e^{im\theta} \in \mathcal{H}$ . It illustrates how the phase of the output features rotates with the input at frequency *m*.

with rotation order  $m_j$  will lead to new feature maps  $\mathbf{X}_{m_i+m_j}^{(l)}$  with rotation order  $m_i+m_j$ . However, considering the summation of two feature maps with the same order m, the obtained result remains of order m (non-linear activation functions do not affect the rotation order) [94]. Consequently, to construct a M-equivariant output, the chained cross-correlation constraint of rotation orders is applied along any data-path of the CNN model. Moreover, the equivariance condition given by Eq. (9) must be met at every feature map:

$$M = \sum_{i=1}^{N} m_i. \tag{9}$$

Our deep rotation-equivariant CNN2D for spectral-spatial HSI data classification has been developed with a dual-path architecture, replacing standard convolution kernels  $\mathbf{W}^{(l)}$  by its CHF counterpart  $\mathbf{W}_m^{(l)}$ , generating *m*-equivariant feature maps with rotation order m. In particular, each path represents a different stream with constant rotation order outputs. As can be observed from Fig. 4, the upper stream exhibits an m = 0 rotation order (i.e. its outputs are rotation-invariant), while the lower stream sets its rotation order to m = 1(i.e. its outputs are rotation-equivariant). To keep the rotation order constant along streams, each one comprises several rotation order zero cross-correlation and non-lineal activation functions. Moreover, every cross-correlation output of each stream is combined within the opposite stream through cross-correlations of rotation order equal to the difference between those two streams,<sup>4</sup> holding at every connection the equivariance condition.

Furthermore, to translate the continuous cross-correlation function into the discrete space of real HSI data, the data is bandlimited and resampled before cross-correlation via Gaussian resampling [94]. Each feature of the resulting downsampled data is a *center of equivariance* where the feature maps will exhibit local rotation equivariance. Also, the complex cross-correlation function has been split into four real cross-correlation:

$$\mathbf{W}_{m} \star \mathbf{X} = \underbrace{\mathbf{W}_{m}^{Re} \star \mathbf{X}^{Re} - \mathbf{W}_{m}^{Im} \star \mathbf{X}^{Im}}_{\text{Real part}}$$

$$\underbrace{i\mathbf{W}_{m}^{Re} \star \mathbf{X}^{Im} - \mathbf{W}_{m}^{Im} \star \mathbf{X}^{Re}}_{\text{Imaginary part}} \quad (10)$$

Following Eq. (10), the CHFs are implemented by resampling the polar-based filters into a grid-resampled version through the the trigonometric transformation, where any coordinate  $(r, \phi)$  can be represented as  $(r \cos(\phi), r \sin(\phi))$  [94].

We summarize the concept of the the network architecture. First of all, the spectral dimension of the HSI scene is reduced to one band, F = 1, applying Maximum Noise Fraction (MNF). Then, following the procedure outlined in [86], the HSI scene is cropped into spectral-spatial patches of size  $N \times N \times 1$ , where the label of a patch corresponds to that of the central pixel. These patches feed the network, whose architecture is composed of two streams with rotation orders m = 0 and m = 1, respectively. Each stream comprises three CHF-based layers, which are followed by data normalization and ReLU activation functions [94]. Moreover, these CHF-based layers contains  $5 \times 5 \times 8$ ,  $5 \times 5 \times 16$  and  $5 \times 5 \times 32$ filters, respectively. This is illustrated in Fig. 4, where each cross-correlation observes Eq. (9). At the end of the network, the two streams come together and are joined into a final layer with m = 0 and with kernel size  $5 \times 5 \times C$ , where C is the number of different land cover classes. This final data representation is vectoriced to obtain the final classification result. The model has been optimized with ADAM optimizer, considering a starting learning rate of 1e - 3. This is updated every 50 epochs, where it is divided by two in order to refine the behavior of the optimizer. 250 epochs have been employed, using a batch size of 100 elements.

#### **III. EXPERIMENTS**

#### A. EXPERIMENTAL ENVIRONMENT

In order to evaluate the performance of the developed approach called reCNN2D (rotation-equivariant CNN2D), we use as hardware environment a 6th Generation Intel<sup>®</sup> Core<sup>TM</sup>i7-6700K processor with 8M of Cache and up to 4.20GHz (4 cores/8 way multitask processing), 40GB of DDR4 RAM with a serial speed of 2400MHz, a graphical processing unit (GPU) NVIDIA GeForce GTX 1080 with 8GB GDDR5X of video memory and 10Gbps of memory frequency, a Toshiba DT01ACA HDD with 7200RPM and 2TB of capacity, and an ASUS Z170 pro-gaming motherboard. The used software environment consists of Ubuntu

<sup>&</sup>lt;sup>4</sup>As we can see, to combine the outputs of stream m = 1 with the outputs of stream m = 0, filters of order m = -1 are needed. In this case, for negative orders, the conjugated weights are considered (Fig. 5), i.e. if we consider  $\mathbf{W}_{-m} = \underline{R}(r)e^{i(m\phi+\beta)}$  as the negative order filter, its conjugated counterpart will be  $\overline{\mathbf{W}_m} = R(r)e^{-i(m\phi+\beta)}$ .



**FIGURE 4.** Graphical overview of the proposed neural network architecture. It is composed by two order-contant streams: the upper one represents order m = 0 (rotation invariance) while the bottom stream represents order m = 1 (rotation equivariance). The output responses of CHFs at each stream are combined by the sum of the feature maps obtained, keeping the stream rotation order constant. The final layers obtain a vectorized representation of the proposed network, where C defines the number of different land cover classes. This vector is used for the final classification.



**FIGURE 5.** Real (upper row) and imaginary (bottom row) parts of the Gaussian-based CHF  $W_m(r, \phi; R, \beta)$ , setting  $R = e^{-r^2}$  and  $\beta = 0$  and considering different rotation orders, in particular m = -4, -2, -1, 0, 1, 2, 4.

16.04.4 x64 as operating system and Python 2.7 as programming language.

## B. EXPERIMENTAL HYPERSPECTRAL DATASETS

For the test instances, three well-know HSI datasets have been used. The first one is known as **Indian Pines** (IP) (see Table 1), which was gathered by AVIRIS [30] in 1992, over a set of agricultural fields with regular geometry and with multiple crops and irregular patches of forest in Northwestern Indiana. The IP scene has  $145 \times 145$  pixels with 224 spectral bands in the range from 400 to 2500nm, with 10nm of spectral resolution, 20m spatial resolution and 16 bits radiometric resolution. After an initial analysis, 4 zero bands and another 20 bands with lower signal-to-noise ratio (SNR) due to atmospheric absorption have been removed, retaining only 200 spectral channels. Moreover, about half of the pixels in the HSI image (10249 of 21025) contain ground-truth information, which comes in the form of a single label assignment having a total of 16 ground-truth classes.

The second HSI dataset, known as **Pavia University** (PU) (see Table 1) was collected by ROSIS [123] during a flight campaign over Pavia, northern Italy. The dataset covers an urban environment with various solid structures (asphalt, gravel, metal sheets, bitumen, bricks), natural objects (trees,

meadows, soil), and shadows (9 classes in total). Other objects with a composition which differs from the labeled ones are considered as clutter. The PU scene was collected over the university area. It contains 103 spectral bands of  $610 \times 340$  pixels in the spectral range from 0.43 to  $0.86 \mu m$ , with a spatial resolution of 1.3m/pixel. About 20.62% of the pixels in the HSI image (42776 of 207400) contain ground-truth information.

The third HSI dataset was also collected by the AVIRIS instrument, in this case over **Salinas Valley** (SV), California (Table 1). The covered area has  $512 \times 217$  samples and the spatial resolution is 3.7m per pixel. 204 out of the 224 bands are kept after 20 water absorption bands are removed. The ground-truth is composed of 54129 pixels and 16 land-cover classes, including vegetables (such as lettuce and brocoli), bare soil, and vineyard fields.

## C. EXPERIMENTAL DISCUSSION

Four experiments have been performed with the aim to test the developed method for HSI classification:

• The first experiment performs a general stateof-the-art comparison, comparing the developed rotation-equivariant CNN2D model for HSI data classification with six popular HSI classifiers [87]: i) RF [41], ii) MLR [124], iii) SVM with radial basis function (RBF) [125], iv) multilayer perceptron (MLP) [126], v) CNN1D [127] and vi) CNN2D [128].

Apart from reCNN2D and the standard CNN2D (implemented similarly), the considered classifiers are traditional pixel-wise methods, whereas CNN2D-based architectures are spectral-spatial classifiers, whose inputs comprises patches of  $31 \times 31$  pixels with one principal component extracted from IP, PU and SV datasets through PCA. The spatial size has been chosen following [86] in order to obtain enough spatial information

INDIAN	PINES		PAVIA UNIVERSI	IVERSITY SALINAS VALLE			
		0 100 200 300 400 500		250 300			20
Color Land-cover	type Samples	Color	Land-cover type	Samples	Color	Land-cover type	Samples
Backgrou	nd 10776		Background	164624		Background	56975
Alfalfa	46		Asphalt	6631		Brocoli-green-weeds-1	2009
Corn-not	ill 1428		Meadows	18649		Brocoli-green-weeds-2	3726
Corn-mi	n 830		Gravel	2099		Fallow	1976
Corn	237		Trees	3064		Fallow-rough-plow	1394
Grass/Past	ure 483		Painted metal sheets	1345		Fallow-smooth	2678
Grass/Tre	ees /30		Bare Soll	5029		Stubble	3939
Grass/pasture-			Diaman	1220		Calaria	2570
H9V-Win/ir/	mowed 28		Bitumen	1330		Celery	3579
	mowed 28 owed 478		Bitumen Self-Blocking Bricks	1330 3682		Celery Grapes-untrained	3579 11271
Oats	mowed 28 owed 478 20		Bitumen Self-Blocking Bricks Shadows	1330 3682 947		Celery Grapes-untrained Soil-vinyard-develop	3579 11271 6203 2278
Oats Soybeans-n	mowed         28           wed         478           20           notill         972           min         2455		Bitumen Self-Blocking Bricks Shadows	1330 3682 947		Celery Grapes-untrained Soil-vinyard-develop Corn-senesced-green-weeds	3579 11271 6203 3278 1068
Soybeans-	mowed         28           wed         478           20         20           notill         972           min         2455           hear         502		Bitumen Self-Blocking Bricks Shadows	1330 3682 947		Celery Grapes-untrained Soil-vinyard-develop Corn-senesced-green-weeds Lettuce-romaine-4wk	3579 11271 6203 3278 1068
Soybeans-n Soybeans-n Soybeans-n	mowed         28           owed         478           20         20           notill         972           min         2455           lean         593           205		Bitumen Self-Blocking Bricks Shadows	1330 3682 947		Celery Grapes-untrained Soil-vinyard-develop Corn-senesced-green-weeds Lettuce-romaine-4wk Lettuce-romaine-5wk	3579 11271 6203 3278 1068 1927 016
Soybeans-n Soybeans-n Soybeans-d Soybean-cl Wheat	mowed         28           owed         478           20         20           notill         972           min         2455           lean         593           205         1265		Bitumen Self-Blocking Bricks Shadows	1330 3682 947		Celery Grapes-untrained Soil-vinyard-develop Corn-senesced-green-weeds Lettuce-romaine-4wk Lettuce-romaine-5wk Lettuce-romaine-6wk	3579 11271 6203 3278 1068 1927 916
Soybeans-n Soybeans-n Soybean-cl Wheat Woods	mowed         28           wwed         478           20           notill         972           min         2455           lean         593           205         1265           a Driver         28		Bitumen Self-Blocking Bricks Shadows	1330 3682 947		Celery Grapes-untrained Soil-vinyard-develop Corn-senesced-green-weeds Lettuce-romaine-4wk Lettuce-romaine-5wk Lettuce-romaine-7wk Viruerd-untering d	3579 11271 6203 3278 1068 1927 916 1070 7268
Soybeans-n Soybeans-n Soybean-cl Wheat Bldg-Grass-Tre	mowed         28           owed         478           20         20           notill         972           min         2455           lean         593           205         1265           e-Drives         386           ower         92	•	Bitumen Self-Blocking Bricks Shadows	1330 3682 947		Celery Grapes-untrained Soil-vinyard-develop Corn-senesced-green-weeds Lettuce-romaine-4wk Lettuce-romaine-5wk Lettuce-romaine-6wk Lettuce-romaine-7wk Vinyard-untrained Vinyard-untrained ter Wis	3579 11271 6203 3278 1068 1927 916 1070 7268
Oats Oats Soybeans-n Soybeans-cl Wheat Wheat Bldg-Grass-Tre Stone-steel t	mowed         28           owed         478           20           notill         972           min         2455           lean         593           205         1265           e-Drives         386           owers         93		Bitumen Self-Blocking Bricks Shadows	1330 3682 947		Celery Grapes-untrained Soil-vinyard-develop Corn-senesced-green-weeds Lettuce-romaine-4wk Lettuce-romaine-5wk Lettuce-romaine-6wk Lettuce-romaine-7wk Vinyard-untrained Vinyard-vertical-trellis	3579 11271 6203 3278 1068 1927 916 1070 7268 1807

#### TABLE 1. Number of samples of the three test HSIs, Indian Pines, Pavia University and Salinas Valley.

within each patch. Moreover, after dividing the scenes into patches of size  $31 \times 31$ , they have been randomly rotated with angles  $0^{\circ}$ ,  $90^{\circ}$ ,  $180^{\circ}$  and  $270^{\circ}$  and have then been divided into training and test sets in order to measure the rotational effect. No data augmentation has been performed during the training of the models.

The training percentage has been varied to test the accuracy performance of the developed method, in particular 3%, 5%, 10% and 5% of the IP dataset (which is spectrally more complex), and 1%, 3%, 5% and 10% of PU and SV datasets, respectively. The accuracy of each method was evaluated by three quantitative metrics: overall accuracy (OA), average accuracy (AA), and the Cohen's kappa (K) coefficient [129].

• The second experiment extends the comparison between the standard and rotation equivariant CNN.

In particular, we have compared the standard CNN2D, considering the IP, PU and SV datasets with rotated and non-rotated training-test samples, and the presented method with rotated inputs. The main goal is to observe the degradation in terms of OA of the standard CNN2D when rotated samples are fed to the model.

Also in this experiment, after dividing the scenes into patches of size  $31 \times 31$ , they have been randomly rotated with angles 0°, 90°, 180° and 270° and then divided into training and test sets (using 3%, 5%, 10% and 5% of IP and 1%, 3%, 5% and 10% of the PU and SV datasets as training percentages) in order to measure the rotational effect over the standard CNN2D. Again, no data augmentation has been performed during the training of the models.

TABLE 2. Overall accuracy varying the training percentage for Indian

Pines, University of Pavia and Salinas Valley datasets.

• The third experiment performs a detailed comparison between the new rotation-equivariant network with CHFs for remote sensing HSI data classification, the standard CNN2D model and the regularized FE model proposed by Chen *et al.* [130]. Chen *et al.* obtain the representation of the deep features of the HSI scenes by implementing a standard CNN2D, performing the final classification through a logistic regression classifier, which is fed with the extracted feature vectors.

The experiment uses IP and PU scenes, where both the spatial size of the input HSI patches  $(27 \times 27)$  and the size of the training and test sets has been adjusted to those indicated in [130] (i.e. 1765 training samples and 6223 test samples of IP scene, and 3930 training samples and 33940 test samples for PU image). Moreover, following the second experiment, the standard CNN2D model and reCNN2D with CHFs have been tested with rotated training-test samples to observe the OA effect. Once more, the input patches have been randomly rotated between considering angles  $0^{\circ}$ ,  $90^{\circ}$ ,  $180^{\circ}$  and  $270^{\circ}$ .

• As a last comparison, the fourth experiment compares the performance between the new rotation-equivariant CNN2D model, the standard CNN2D and the feature enhancement fully CNN2D model FEFCN) proposed by Li *et al.* in [131]. This network comprises several FE stages of standard convolution, deconvolution and pooling layers. Obtained feature vectors are processed by an ELM [71] for classification purposes.

The experiment follows the training details described in [131], where 10% and 5% of training data is used for the IP and PU data sets respectively, using patches of 48 × 48. For the standard CNN2D and reCNN2D, the HSI data has been randomly rotated with angles of  $0^{\circ}$ ,  $90^{\circ}$ ,  $180^{\circ}$  and  $270^{\circ}$ .

## 1) FIRST EXPERIMENT: GENERAL STATE-OF-THE-ART COMPARISON

The obtained results of the first experiment can be observed in Table 2. With the exception of two isolated cases (PU and SV scenes with 1% of training samples), reCNN2D achieves the best classification result. Looking at the spectral classifiers (which are not affected by the random rotation of spatial information), they reach typical OA results, improving as more training samples are provided, where SVM and CNN1D exhibit the best classification results. However, the graphic results depicted in the classification maps of Fig. 6 show the traditional *salt and pepper* noise due to the incorrect classification of those pixels located in the interior areas of the classes.

On the contrary, those classifiers that introduce contextual information can overcome this limitation, reducing the uncertainty introduced by the spectral variability by taking advantage of the information provided by the neighbors pixels. These classifiers are based on a CNN2D model, and they have been denoted as *CNN2D\_wR* and *reCNN2D*,

INDIAN DINES									
T. Size	RF	MLR	SVM	MLP	CNN1D	CNN2D wR	reCNN2D		
3%	65.65	66.09	70.48	68.65	67.52	41.95	78.93		
5%	68.33	70.44	75.57	73.38	72.97	47.56	85.20		
10%	73.43	75.67	81.45	79.82	80.25	63.63	93.96		
15%	75.62	78.16	83.97	82.77	83.71	77.39	96.20		
	PAVIA UNIVERSITY								
T. Size	RF	MLR	SVM	MLP	CNN1D	CNN2D_wR	reCNN2D		
1%	81.10	84.97	88.67	85.19	90.04	74.64	85.21		
3%	86.04	88.39	91.54	91.41	93.01	86.76	95.23		
5%	87.73	88.99	92.68	92.84	93.70	88.52	97.25		
10%	89.92	89.73	93.97	94.21	94.73	95.42	98.89		
SALINAS VALLEY									
T. Size	RF	MLR	SVM	MLP	CNN1D	CNN2D_wR	reCNN2D		
1%	86.42	88.91	89.41	88.76	90.54	59.69	85.20		
3%	88.95	91.07	92.26	91.40	93.04	72.77	93.53		
5%	89.55	91.75	92.85	92.31	93.86	75.55	95.50		
10%	90.03	92.35	93.65	93.68	94.84	79.73	98.18		

respectively. The first one comprises the standard 2D convolution filters, receiving as input data, rotated and non-rotated spatial patches (hence the suffix wR, i.e. with rotation), while the second one implements the new methodology, receiving also rotated and non-rotated spatial patches.

Focusing on IP, one can observe that the reCNN2D achieves the best overall accuracy (OA, 78.93%) with least training data (1%). Comparing the traditional CNN2D wR with reCNN2D we can observe how the last one performs 36.98 percentage points better than the CNN2D wR model. It clearly appears more robust to rotational variations when few training samples are provided, understanding the concept of "robustness" as the ability of the neural model to cope with complex data (in particular, rotated data with few training samples) during classification, without its classification performance being adversely affected. This behavior is repeated with the rest of the percentages (for instance 37.64 and 30.33 percentage points better with 5% and 10% of raining samples), although as more training is added, the distance is reduced (18.81 percentage points better with 15%). This makes sense, since as more samples are introduced, networks are able to cover the data variability, learning better during the training stage and generalizing better over the unseen test samples.

Considering PU and SV datasets, SVM is the best classifier with only 1% of training data, while the new rotation-equivariant CNN2D is the third one (behind the CNN1D). However, for the remaining percentages, reCNN2D reaches the first position, exhibiting the best OA result. Moreover, reCNN2D exhibits a better generalization power than CNN2D with 1%, 3%, 5%, 10% and 15% of training data. While standard CNN2D hardly reaches the spectral classifiers performance, reCNN2D is able to overcome the OA values without difficulty. Regarding, the classification maps in Fig. 6, we can observe an important salt and pepper noise within spectral methods. In particular in SV scene, it is very interesting to note how the CNN2D-based network suffers more than pixel-wise models in the lettuce region, which is quite hard to identify due the



FIGURE 6. Classification maps using 10% of training data considering Indian Pines scene and 5% of training data considering Pavia University and Salinas valley datasets. From left to right, ground-truth, RF, MLR, SVM, MLP, CNN1D, traditional CNN2D and reCNN2D classifier results.

high similarity between the spectral signatures of the lettuces at different stages of maturation. The same behaviour is observed between Vineyard-untrained and Grapes-untrained classes. In this sense, the problem of rotation is compounded by the lack of spectral information, thus the lack of spectral information between very similar spectra and the rotation of patches (in particular, the orientation of those patches with edges between different classes) make the CNN2D classifier negatively affected. On the contrary, reCNN2D provides a clearer map, indetifying these regions better than CNN2D\_wR model.

## 2) SECOND EXPERIMENT: EVALUATING THE BEHAVIOR OF

CONVOLUTIONAL MODELS WITH INPUT ROTATED PATCHES This experiment is focused on studying how traditional convolution kernel is negatively affected by the introduction of rotated patches in both training and testing, while new reCNN2D CHF-based architecture is more robust and generalizes better than standard CNN2D.

By rotating both the training and test patches, we are introducing some "spatial variability". In this sense, traditional convolution filters will learn to identify the features associated with some orientation, compelling them to reserve

VOLUME 8, 2020

many parameters that in the end are redundant, as they are rotated copies of the same filter. On the contrary, reCNN2D should reveal a better performance, since its filters can obtain rotation-equivariant responses, independently of the input degree of transformation, so it should generalize better in those unseen samples of the test set.

Obtained results can be graphically observed in Fig. 7. On the upper row, we compare the standard CNN2D model introducing randomly rotated (CNN2D\_wR) and non-rotated (CNN2D nR) patches within the training and test sets. The obtained results of CNN2D\_nR are as expected [87], exhibiting a low performance when few training samples are available during the training stage, specially with IP and SV datasets, where first one the has an important spectral mixture of the pixels due to the low spatial resolution, while the second contains two problematic regions due to the similar spectral signatures of the lettuces at different stages of maturation and the vineyards crops. However, the *CNN2D\_nR* improves its classification performance as more training samples are provided, reaching an OA higher than 95%. Despite these results, the standard CNN2D performance degrades significantly when rotated patches are introduced. Focusing on IP scene, the CNN2D wR OA falls around 30 percentage



FIGURE 7. Overall accuracy comparison between standard CNN2D with and without rotated training and test input patches (denoted as CNN2D\_wR and CNN2D\_nR, respectively), and reCNN2D with rotated training and test input patches. On the upper row, comparison between CNN2D\_wR and CNN2D\_nR. On the bottom row, comparison between CNN2D\_wR and reCNN2D.

points. This fall is slightly less in the other two HSI scenes, because they have more training samples than IP, so they can cover more information than IP, although the classification performance reduction is still remarkable.

On the bottom row, we compare the  $CNN2D_wR$  and the new network (denoted as reCNN2D), including rotated patches within the raining and test sets. In this sense, these images graphically depict the OA provided in Table 2. In PU the difference between  $CNN2D_wR$  and reCNN2D is slightly smaller than in IP and SV scenes. The main reason resides in the large number of structures, which are oriented and located in different areas of the scene, providing a direct way for the traditional filters to cover the features. However, IP and SV are more complex scenes not only due to their spectral complexity, but also because of the lack of structure. In this context, the  $CNN2D_wR$  suffers a dramatic OA reduction compared to the rotation-equivariant reCNN2D model.

On the other hand, if we compare the *CNN2D\_nR* (without rotated inputs) and *reCNN2D* (with rotated inputs) we can observe that the obtained results are quite similar as more training samples are introduced, where *reCNN2D* obtains the best OA percentages when there are very few samples (for instance in IP and SV scene).

## 3) THIRD EXPERIMENT: COMPARISON AMONG CNN2D MODELS

The third experiment compares several CNN models. In particular, we compare the CNN2D-LR proposed by Chen *et al.* 

 TABLE 3. Comparative with several CNN models: CNN2D-LR [130]

 (considering non-rotated input patches within training/test sets),

 standard CNN2D and reCNN2D (both networks work with rotated input patches within training/test sets.

	IN	DIAN PINES		PAVIA UNIVERSITY			
Class	CNN2D-LR_nR	CNN2D_wR	reCNN2D	CNN2D-LR_nR	CNN2D_wR	reCNN2D	
1	99.65	74.29	98.57	97.11	85.11	98.44	
2	90.64	57.77	93.32	87.66	64.82	98.38	
3	99.11	81.44	96.32	99.69	88.59	99.29	
4	100.0	90.81	99.59	98.49	81.86	96.62	
5	98.48	83.87	93.53	100.0	99.60	99.62	
6	97.95	91.55	97.55	98.00	81.49	99.76	
7	100.0	88.57	91.43	99.89	76.57	99.81	
8	100.0	99.63	99.83	99.70	94.35	98.94	
9	100.0	80.00	100.0	97.11	97.76	98.26	
10	95.33	69.97	95.16	-	-	-	
11	78.21	55.81	92.78	-	-	-	
12	99.39	81.80	96.57	-	-	-	
13	100.0	99.40	99.80	-	-	-	
14	97.71	90.84	98.73	-	-	-	
15	99.31	69.97	96.23	-	-	-	
16	99.22	95.13	96.67	-	-	-	
OA	89.99	72.42	95.31	94.04	76.35	98.59	
AA	97.19	81.93	96.63	97.52	85.57	98.79	
K(x100)	87.95	68.54	94.55	92.43	72.57	98.07	
		-			-	-	

in [130] (called *CNN2D-LR\_nR*, as it receives non-rotated patches), the standard CNN2D and *reCNN2D*. The same experimental configuration as described in [130] have been used. Table 3 provides the obtained results over IP and PU datasets. The first column provides the results obtained by the *CNN2D-LR\_nR* model in [130], which has been trained with 1765 training samples from IP and 3930 from PU, using patches of  $27 \times 27$ . Following the same training procedure, the second and third columns show the results obtained by the *CNN2D\_wR* with the same architecture as Fig. 4, implemented by traditional convolutional kernels, and the new method respectively. For this experiment, patches have been

 TABLE 4. Comparative with several CNN models: FCEFCN [131]

 (considering non-rotated data), standard CNN2D and reCNN2D (both with rotated input data).

		INDIAN PINES		PAVIA UNIVERSITY			
Class	FCEFCN	CNN2D_wR	reCNN2D	FCEFCN	CNN2D_wR	reCNN2D	
1	94.29	40.49	90.81	94.97	87.99	98.09	
2	82.36	60.80	97.35	96.44	92.80	99.72	
3	77.14	68.37	97.93	84.69	79.27	97.89	
4	65.97	52.44	98.18	97.39	70.04	85.06	
5	91.14	50.71	96.11	99.14	97.76	96.36	
6	91.46	76.38	98.17	94.77	95.57	99.79	
7	90.91	40.80	95.65	88.90	69.46	99.08	
8	97.49	91.88	99.84	84.11	92.16	98.08	
9	50.00	30.56	94.38	100.0	84.02	83.89	
10	78.24	52.65	97.00	-	-	-	
11	78.35	71.61	98.94	-	-	-	
12	75.50	52.15	94.68	-	-	-	
13	91.88	67.68	98.50	-	-	-	
14	93.23	89.83	99.11	-	-	-	
15	76.39	54.03	97.50	-	-	-	
16	98.72	41.31	80.79	-	-	-	
OA	83.12	67.70	97.78	94.35	89.27	97.72	
AA	83.32	58.86	95.93	93.38	85.45	95.33	
K(x100)	90.63	62.92	97.47	92.49	86.12	96.97	

randomly rotated with angles  $0^{\circ}$ ,  $90^{\circ}$ ,  $180^{\circ}$  and  $270^{\circ}$  and then divided into training and test sets.

As we can observe, for both data sets the new neural model provides the best OA results. In fact, its OA results are 5.32 and 4.55 percentage points better than the second best model, the  $CNN2D-LR_nR$ , and 22.89 and 22.24 points better than the  $CNN2D_wR$  model.

## 4) FOURTH EXPERIMENT: COMPARISON BETWEEN ROTATION-EQUIVARIANCE CNN AND FULLY-CNN MODELS

The last experiment compares the efficiency in terms of OA, AA, and Kappa values between the feature enhancement fully CNN (FEFCNN) [131] model, the standard CNN2D model with rotated inputs (CNN2D\_wR) and the newly designed rotation-equivariant approach reCNN2D. The obtained results are shown in Table 4 for both IP and PU data sets considering the experimental configuration provided in [131]. In this sense, the comparison have been conducted by utilizing the 10% and 5% of available labeled training samples randomly taken from IP and PU datasets. The FEFCNN network takes the input of overlapped patches of size  $48 \times 48$ pixels with a stride of 15 for both IP and PU data sets. The second and third columns of Table 4 reports the results obtained by the CNN2D\_wR with the backbone architecture shown in Fig. 4, which incorporates traditional convolutional kernels, and the new method respectively. In order to explore the rotational effect the extracted patches have been randomly rotated  $0^{\circ}$ ,  $90^{\circ}$ ,  $180^{\circ}$  and  $270^{\circ}$  angles and then the transform samples are divided into training and test sets. It can be seen that, for both data sets, the new rotation-equivariant network for HSI classification provides the best OA, AA, and Kappa values showing the robustness under rotational variations.

#### **IV. CONCLUSION**

This article presents a new rotation-equivariant convolutional neural network reCNN2D based on CHFs for classyfying HSI remote sensing data. Obtained results with three popular real HSI data sets varying the rotation in images in a systematic way, illustrate the performance of the new method with a small amount of training data, avoiding the overfitting problem exhibited by traditional CNN models. Moreover, comparison with several CNN2D models shows that the new approach exhibits a better robustness to data rotational variance, performing a better classification and providing a better generalization without requiring data augmenting techniques.

#### REFERENCES

- A. F. H. Goetz, G. Vane, J. E. Solomon, and B. N. Rock, "Imaging spectrometry for earth remote sensing," *Science*, vol. 228, no. 4704, pp. 1147–1153, 1985.
- [2] G. Wang and Q. Weng, *Remote Sensing of Natural Resources*. Boca Raton, FL, USA: CRC Press, 2013.
- [3] P. Treitz and P. Howarth, "Hyperspectral remote sensing for estimating biophysical parameters of forest ecosystems," *Prog. Phys. Geography*, vol. 23, no. 3, pp. 359–390, Sep. 1999.
- [4] P. H. Sampson, P. J. Zarco-Tejada, G. H. Mohammed, J. R. Miller, and T. L. Noland, "Hyperspectral remote sensing of forest condition: Estimating chlorophyll content in tolerant hardwoods," *Forest Sci.*, vol. 49, no. 3, pp. 381–391, 2003.
- [5] E. Reyes, R. J. Abad, J. Diaz, and V. Manian, "Change detection assessment in a tropical forest using multispectral and hyperspectral images," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2017, pp. 624–627.
- [6] F. A. Kruse, J. W. Boardman, and J. F. Huntington, "Comparison of airborne hyperspectral data and eo-1 hyperion for mineral mapping," *IEEE Trans. Geosci. Remote Sens.*, vol. 41, no. 6, pp. 1388–1400, Jun. 2003.
- [7] C. A. Bishop, J. G. Liu, and P. J. Mason, "Hyperspectral remote sensing for mineral exploration in Pulang, Yunnan Province, China," *Int. J. Remote Sens.*, vol. 32, no. 9, pp. 2409–2426, May 2011.
- [8] F. D. Van der Meer, H. M. Van der Werff, F. J. Van Ruitenbeek, C. A. Hecker, W. H. Bakker, M. F. Noomen, M. Van Der Meijde, E. J. M. Carranza, J. B. De Smeth, and T. Woldai, "Multi-and hyperspectral geologic remote sensing: A review," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 14, no. 1, pp. 112–128, Feb. 2012.
- [9] S. Koponen, "Lake water quality classification with airborne hyperspectral spectrometer and simulated MERIS data," *Remote Sens. Environ.*, vol. 79, no. 1, pp. 51–59, Jan. 2002.
- [10] V. E. Brando and A. G. Dekker, "Satellite hyperspectral remote sensing for estimating estuarine and coastal water quality," *IEEE Trans. Geosci. Remote Sens.*, vol. 41, no. 6, pp. 1378–1387, Jun. 2003.
- [11] L. G. Olmanson, P. L. Brezonik, and M. E. Bauer, "Airborne hyperspectral remote sensing to assess spatial distribution of water quality characteristics in large rivers: The mississippi river and its tributaries in minnesota," *Remote Sens. Environ.*, vol. 130, pp. 254–265, Mar. 2013.
- [12] D. Haboudane, J. R. Miller, N. Tremblay, P. J. Zarco-Tejada, and L. Dextraze, "Integrated narrow-band vegetation indices for prediction of crop chlorophyll content for application to precision agriculture," *Remote Sens. Environ.*, vol. 81, nos. 2–3, pp. 416–426, Aug. 2002.
- [13] H. Cetin, J. T. Pafford, and T. G. Mueller, "Precision agriculture using hyperspectral remote sensing and GIS," in *Proc. 2nd Int. Conf. Recent Adv. Space Technol. (RAST)*, 2005, pp. 70–77.
- [14] M. Teke, H. S. Deveci, O. Haliloğlu, S. Z. Gurbuz, and U. Sakarya, "A short survey of hyperspectral remote sensing applications in agriculture," in *Proc. 6th Int. Conf. Recent Adv. Space Technol. (RAST)*, Jun. 2013.
- [15] S. G. Bajwa and L. F. Tian, "Soil fertility characterization in agricultural fields using hyperspectral remote sensing," *Trans. ASAE*, vol. 48, no. 6, pp. 2399–2406, 2005.
- [16] É. Arsenault and F. Bonn, "Evaluation of soil erosion protective cover by crop residues using vegetation indices and spectral mixture analysis of multispectral and hyperspectral data," *Catena*, vol. 62, nos. 2–3, pp. 157–172, Aug. 2005.
- [17] T. Schmid, M. Rodríguez-Rastrero, P. Escribano, A. Palacios-Orueta, E. Ben-Dor, A. Plaza, R. Milewski, M. Huesca, A. Bracken, V. Cicuéndez, and M. Pelayo, "Characterization of soil erosion indicators using hyperspectral data from a mediterranean rainfed cultivated region," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 2, pp. 845–860, Feb. 2016.
- [18] M. Zhang, Z. Qin, X. Liu, and S. L. Ustin, "Detection of stress in tomatoes induced by late blight disease in California, USA, using hyperspectral remote sensing," *Int. J. Appl. Earth Observ. Geoinformation*, vol. 4, no. 4, pp. 295–310, Nov. 2003.

- [19] J. G. P. W. Clevers and L. Kooistra, "Using hyperspectral remote sensing data for retrieving canopy chlorophyll and nitrogen content," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 5, no. 2, pp. 574–583, Apr. 2012.
- [20] S. L. MacDonald, M. Staid, M. Staid, and M. L. Cooper, "Remote hyperspectral imaging of grapevine leafroll-associated virus 3 in cabernet sauvignon vineyards," *Comput. Electron. Agricult.*, vol. 130, pp. 109–117, Nov. 2016.
- [21] U. Heiden, K. Segl, S. Roessner, and H. Kaufmann, "Determination of robust spectral features for identification of urban surface materials in hyperspectral remote sensing data," *Remote Sens. Environ.*, vol. 111, no. 4, pp. 537–552, Dec. 2007.
- [22] U. Heiden, W. Heldens, S. Roessner, K. Segl, T. Esch, and A. Mueller, "Urban structure type characterization using hyperspectral remote sensing and height information," *Landscape Urban Planning*, vol. 105, no. 4, pp. 361–375, Apr. 2012.
- [23] C. Weber, R. Aguejdad, X. Briottet, J. Avala, S. Fabre, J. Demuynck, E. Zenou, Y. Deville, M. S. Karoui, F. Z. Benhalouche, and S. Gadal, "Hyperspectral imagery for environmental urban planning," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2018, pp. 1628–1631.
- [24] S. Bhaskaran, B. Forster, B. Datt, and T. Neal, "Hail storm vulnerability assessment by using hyperspectral remote sensing and GIS techniques," in *Proc. Scanning Present Resolving Future, IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, vol. 4, Jul. 2001, pp. 1826–1828.
- [25] J. Bustamante, D. Aragonés, I. Afán, C. J. Luque, A. Pérez-Vázquez, E. M. Castellanos, and R. Díaz-Delgado, "Hyperspectral sensors as a management tool to prevent the invasion of the exotic cordgrass spartina densiflora in the Doñana wetlands," *Remote Sens.*, vol. 8, no. 12, p. 1001, Dec. 2016.
- [26] R. D. M. Scafutto, C. R. de Souza Filho, and W. J. de Oliveira, "Hyperspectral remote sensing detection of petroleum hydrocarbons in mixtures with mineral substrates: Implications for onshore exploration and monitoring," *ISPRS J. Photogramm. Remote Sens.*, vol. 128, pp. 146–157, Jun. 2017.
- [27] R. F. Kokaly, B. W. Rockwell, S. L. Haire, and T. V. V. King, "Characterization of post-fire surface cover, soils, and burn severity at the Cerro Grande Fire, New Mexico, using hyperspectral and multispectral remote sensing," *Remote Sens. Environ.*, vol. 106, no. 3, pp. 305–325, Feb. 2007.
- [28] H. Zhang, H.-J. Song, and B.-C. Yu, "Application of hyper spectral remote sensing for urban forestry monitoring in natural disaster zones," in *Proc. Int. Conf. Comput. Manage. (CAMAN)*, May 2011, pp. 1–4.
- [29] Y. Wan, X. Hu, Y. Zhong, A. Ma, L. Wei, and L. Zhang, "Tailings reservoir disaster and environmental monitoring using the UAV-ground hyperspectral joint observation and processing: A case of study in xinjiang, the belt and road," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.* (IGARSS), Jul. 2019, pp. 9713–9716.
- [30] R. O. Green, M. L. Eastwood, C. M. Sarture, T. G. Chrien, M. Aronsson, B. J. Chippendale, J. A. Faust, B. E. Pavri, C. J. Chovit, M. Solis, and M. R. Olah, "Imaging spectroscopy and the airborne visible/infrared imaging spectrometer (AVIRIS)," *Remote Sens. Environ.*, vol. 65, no. 3, pp. 227–248, Sep. 1998.
- [31] C.-I. Chang, Hyperspectral Imaging: Techniques for Spectral Detection and Classification. New York, NY, USA: Springer, 2003.
- [32] P. Fisher, "The pixel: A snare and a delusion," Int. J. Remote Sens., vol. 18, no. 3, pp. 679–685, Feb. 1997.
- [33] J. M. 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, Jan. 2017.
- [34] M. E. Paoletti, J. M. Haut, J. Plaza, and A. Plaza, "Estudio comparativo de Técnicas de clasificación de imágenes hiperespectrales," *Revista Iberoamericana de Automática e Informática Industrial*, vol. 16, no. 2, pp. 129–137, 2019.
- [35] F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 8, pp. 1778–1790, Aug. 2004.
- [36] M. E. Paoletti, J. M. Haut, X. Tao, J. P. Miguel, and A. Plaza, "A new GPU implementation of support vector machines for fast hyperspectral image classification," *Remote Sens.*, vol. 12, no. 8, p. 1257, Apr. 2020.
- [37] J.-M. Yang, P.-T. Yu, and B.-C. Kuo, "A nonparametric feature extraction and its application to nearest neighbor classification for hyperspectral image data," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 3, pp. 1279–1293, Mar. 2010.

- [38] C. Cariou and K. Chehdi, "A new k-nearest neighbor density-based clustering method and its application to hyperspectral images," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2016, pp. 6161–6164.
- [39] J. Li, J. M. Bioucas-Dias, and A. Plaza, "Semisupervised hyperspectral image classification using soft sparse multinomial logistic regression," *IEEE Geosci. Remote Sens. Lett.*, vol. 10, no. 2, pp. 318–322, Mar. 2013.
- [40] M. Khodadadzadeh, J. Li, A. Plaza, and J. M. Bioucas-Dias, "A subspacebased multinomial logistic regression for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 12, pp. 2105–2109, Dec. 2014.
- [41] 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.
- [42] J. C.-W. Chan and D. Paelinckx, "Evaluation of random forest and AdaBoost tree-based ensemble classification and spectral band selection for ecotope mapping using airborne hyperspectral imagery," *Remote Sens. Environ.*, vol. 112, no. 6, pp. 2999–3011, Jun. 2008.
- [43] A. Plaza, J. Plaza, A. Paz, and S. Sánchez, "Parallel hyperspectral image and signal processing," *IEEE Signal Process. Mag.*, vol. 28, no. 3, pp. 119–126, May 2011.
- [44] M. Shah, Y. Liu, and H. Hayat, "Rotation-based object-oriented ensemble in land use land cover classification of hyperspectral data," in *Proc. 6th Int. Conf. Innov. Comput. Technol. (INTECH)*, Aug. 2016, pp. 296–301.
- [45] J. M. Bioucas-Dias, A. Plaza, G. Camps-Valls, P. Scheunders, N. 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.
- [46] 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.
- [47] Y. Tarabalka, M. Fauvel, J. Chanussot, and J. A. Benediktsson, "SVMand MRF-based method for accurate classification of hyperspectral images," *IEEE Geosci. Remote Sens. Lett.*, vol. 7, no. 4, pp. 736–740, Oct. 2010.
- [48] B. Zhang, S. Li, X. Jia, L. Gao, and M. Peng, "Adaptive Markov random field approach for classification of hyperspectral imagery," *IEEE Geosci. Remote Sens. Lett.*, vol. 8, no. 5, pp. 973–977, Sep. 2011.
- [49] J. Li, J. M. Bioucas-Dias, and A. Plaza, "Spectral-spatial hyperspectral image segmentation using subspace multinomial logistic regression and Markov random fields," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 3, pp. 809–823, Mar. 2012.
- [50] L. Sun, Z. Wu, J. Liu, L. Xiao, and Z. Wei, "Supervised Spectral– Spatial hyperspectral image classification with weighted Markov random fields," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 3, pp. 1490–1503, Mar. 2015.
- [51] L.-Z. Huo and P. Tang, "Spectral and spatial classification of hyperspectral data using SVMs and Gabor textures," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2011, pp. 1708–1711.
- [52] R. Seifi Majdar and H. Ghassemian, "A probabilistic SVM approach for hyperspectral image classification using spectral and texture features," *Int. J. Remote Sens.*, vol. 38, no. 15, pp. 4265–4284, Aug. 2017.
- [53] L. He, J. Li, A. Plaza, and Y. Li, "Discriminative low-rank Gabor filtering for Spectral–Spatial hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 3, pp. 1381–1395, Mar. 2017.
- [54] X. Kang, S. Li, and J. A. Benediktsson, "Spectral–Spatial hyperspectral image classification with edge-preserving filtering," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 5, pp. 2666–2677, May 2014.
- [55] Y. Guo, H. Cao, S. Han, Y. Sun, and Y. Bai, "Spectral-spatial HyperspectralImage classification with K-Nearest neighbor and guided filter," *IEEE Access*, vol. 6, pp. 18582–18591, 2018.
- [56] L. Wang, J. Zhang, P. Liu, K.-K. R. Choo, and F. Huang, "Spectral-spatial multi-feature-based deep learning for hyperspectral remote sensing image classification," *Soft Comput.*, vol. 21, no. 1, pp. 213–221, Jan. 2017.
- [57] J. Liao and L. Wang, "Hyperspectral image classification based on fusion of curvature filter and domain transform recursive filter," *Remote Sens.*, vol. 11, no. 7, p. 833, Apr. 2019.
- [58] J. Liao and L. Wang, "Adaptive hyperspectral image classification based on the fusion of manifolds filter and spatial correlation features," *IEEE Access*, vol. 8, pp. 90390–90409, 2020.
- [59] M. Pesaresi and J. A. Benediktsson, "A new approach for the morphological segmentation of high-resolution satellite imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 39, no. 2, pp. 309–320, 2001.

- [60] M. D. Mura, J. A. Benediktsson, J. Chanussot, and L. Bruzzone, "The evolution of the morphological profile: From panchromatic to hyperspectral images," in Optical Remote Sensing: Advances in Signal Processing and Exploitation Techniques, S. Prasad, L. M. Bruce, and J. Chanussot, Eds. Berlin, Germany: Springer, 2011, pp. 123-146.
- [61] J. A. Benediktsson, J. A. Palmason, and J. R. Sveinsson, "Classification of hyperspectral data from urban areas based on extended morphological profiles," IEEE Trans. Geosci. Remote Sens., vol. 43, no. 3, pp. 480-491, Mar. 2005.
- [62] A. Plaza, P. Martínez, J. Plaza, and R. Perez, "Dimensionality reduction and classification of hyperspectral image data using sequences of extended morphological transformations," IEEE Trans. Geosci. Remote Sens., vol. 43, no. 3, pp. 466-479, Mar. 2005.
- [63] J. Plaza, A. Plaza, and C. Barra, "Multi-channel morphological profiles for classification of hyperspectral images using support vector machines," Sensors, vol. 9, no. 1, pp. 196-218, Jan. 2009.
- [64] M. Fauvel, J. A. Benediktsson, J. Chanussot, and J. R. Sveinsson, "Spectral and spatial classification of hyperspectral data using SVMs and morphological profiles," IEEE Trans. Geosci. Remote Sens., vol. 46, no. 11, pp. 3804–3814, Nov. 2008.
- [65] X. Huang, X. Guan, J. A. Benediktsson, L. Zhang, J. Li, A. Plaza, and M. D. Mura, "Multiple morphological profiles from multicomponentbase images for hyperspectral image classification," IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 7, no. 12, pp. 4653-4669, Dec. 2014.
- [66] M. Dalla Mura, J. Atli Benediktsson, B. Waske, and L. Bruzzone, "Extended profiles with morphological attribute filters for the analysis of hyperspectral data," Int. J. Remote Sens., vol. 31, no. 22, pp. 5975-5991, Dec. 2010.
- [67] N. Falco, J. A. Benediktsson, and L. Bruzzone, "Spectral and spatial classification of hyperspectral images based on ICA and reduced morphological attribute profiles," IEEE Trans. Geosci. Remote Sens., vol. 53, no. 11, pp. 6223-6240, Nov. 2015.
- [68] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," IEEE Trans. Pattern Anal. Mach. Intell., vol. 24, no. 7, pp. 971–987. Jul. 2002.
- [69] W. Li, C. Chen, H. Su, and Q. Du, "Local binary patterns and extreme learning machine for hyperspectral imagery classification," IEEE Trans. Geosci. Remote Sens., vol. 53, no. 7, pp. 3681-3693, Jul. 2015.
- [70] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," Neurocomputing, vol. 70, nos. 1-3, pp. 489-501, Dec. 2006.
- [71] J. M. Haut, M. E. Paoletti, J. Plaza, and A. Plaza, "Fast dimensionality reduction and classification of hyperspectral images with extreme learning machines," J. Real-Time Image Process., vol. 15, no. 3, pp. 439-462, Oct 2018
- [72] Z. Ye, R. Dong, L. Bai, C. Jin, and Y. Nian, "Hyperspectral image classification based on segmented local binary patterns," Sens. Imag., vol. 21, no. 1, pp. 1-16, Dec. 2020.
- [73] P. Sidike, C. Chen, V. Asari, Y. Xu, and W. Li, "Classification of hyperspectral image using multiscale spatial texture features," in Proc. 8th Workshop Hyperspectral Image Signal Process., Evol. Remote Sens. (WHISPERS), Aug. 2016, pp. 1-4.
- [74] Y. Tarabalka, J. A. Benediktsson, and J. Chanussot, "Spectral-spatial classification of hyperspectral imagery based on partitional clustering techniques," IEEE Trans. Geosci. Remote Sens., vol. 47, no. 8, pp. 2973–2987, Aug. 2009.
- [75] Y. Tarabalka, J. Chanussot, and J. A. Benediktsson, "Segmentation and classification of hyperspectral images using watershed transformation," Pattern Recognit., vol. 43, no. 7, pp. 2367-2379, Jul. 2010.
- [76] Z. Zhang, E. Pasolli, M. M. Crawford, and J. C. Tilton, "An active learning framework for hyperspectral image classification using hierarchical segmentation," IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 9, no. 2, pp. 640-654, Feb. 2016.
- [77] Y. Tarabalka, J. Atli Benediktsson, J. Chanussot, and J. C. Tilton, "Multiple spectral-spatial classification approach for hyperspectral data," IEEE Trans. Geosci. Remote Sens., vol. 48, no. 11, pp. 4122-4132, Nov. 2010.
- [78] Y. Gu, J. Chanussot, X. Jia, and J. A. Benediktsson, "Multiple kernel learning for hyperspectral image classification: A review," IEEE Trans. Geosci. Remote Sens., vol. 55, no. 11, pp. 6547-6565, Nov. 2017.
- [79] G. Camps-Valls, L. Gomez-Chova, J. Munoz-Mari, J. Vila-Frances, and J. Calpe-Maravilla, "Composite kernels for hyperspectral image classification," IEEE Geosci. Remote Sens. Lett., vol. 3, no. 1, pp. 93-97, Jan. 2006.

- [80] J. Li, P. R. Marpu, A. Plaza, J. M. Bioucas-Dias, and J. A. Benediktsson, "Generalized composite kernel framework for hyperspectral image classification," IEEE Trans. Geosci. Remote Sens., vol. 51, no. 9, pp. 4816-4829, Sep. 2013.
- [81] G. Camps-Valls, N. Shervashidze, and K. M. Borgwardt, "Spatio-spectral remote sensing image classification with graph kernels," IEEE Geosci. Remote Sens. Lett., vol. 7, no. 4, pp. 741-745, Oct. 2010.
- [82] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, pp. 436-444, May 2015.
- [83] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA, USA: MIT Press, 2016.
- Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based [84] classification of hyperspectral data," IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 7, no. 6, pp. 2094-2107, Jun. 2014.
- [85] 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.
- [86] 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., vol. 145, pp. 120-147, Nov. 2018.
- [87] M. E. Paoletti, J. M. Haut, J. Plaza, and A. Plaza, "Deep learning classifiers for hyperspectral imaging: A review," ISPRS J. Photogramm. Remote Sens., vol. 158, pp. 279-317, Dec. 2019.
- [88] M. J. Khan, H. S. Khan, A. Yousaf, K. Khurshid, and A. Abbas, "Modern trends in hyperspectral image analysis: A review," IEEE Access, vol. 6, pp. 14118-14129, 2018.
- [89] H. Greenspan, S. Belongie, R. Goodman, P. Perona, S. Rakshit, and C. H. Anderson, "Overcomplete steerable pyramid filters and rotation invariance," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 1994, pp. 222-228.
- [90] M. N. Do and M. Vetterli, "Rotation invariant texture characterization and retrieval using steerable wavelet-domain hidden Markov models," IEEE Trans. Multimedia, vol. 4, no. 4, pp. 517–527, Dec. 2002. [91] B. Fasel and D. Gatica-Perez, "Rotation-invariant neoperceptron," in
- Proc. 18th Int. Conf. Pattern Recognit. (ICPR), vol. 3, 2006, pp. 336–339.
- [92] K. Huang and S. Aviyente, "Rotation invariant texture classification with ridgelet transform and Fourier transform," in Proc. Int. Conf. Image Process., Oct. 2006, pp. 2141-2144.
- [93] D. Marcos, M. Volpi, and D. Tuia, "Learning rotation invariant convolutional filters for texture classification," in Proc. 23rd Int. Conf. Pattern Recognit. (ICPR), Dec. 2016, pp. 2012-2017.
- [94] D. E. Worrall, S. J. Garbin, D. Turmukhambetov, and G. J. Brostow, "Harmonic networks: Deep translation and rotation equivariance," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 5028-5037.
- [95] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," 2018, arXiv:1803.07728. [Online]. Available: http://arxiv.org/abs/1803.07728
- [96] C. K. Chui, S.-B. Lin, and D.-X. Zhou, "Deep neural networks for rotation-invariance approximation and learning," 2019. arXiv:1904.01814. [Online]. Available: http://arxiv.org/abs/1904.01814
- [97] C. Tao, Y. Tang, C. Fan, and Z. Zou, "Hyperspectral imagery classification based on rotation-invariant Spectral-Spatial feature," IEEE Geosci. Remote Sens. Lett., vol. 11, no. 5, pp. 980-984, May 2014.
- [98] S. Prasad, D. Labate, M. Cui, and Y. Zhang, "Morphologically decoupled structured sparsity for rotation-invariant hyperspectral image analysis," IEEE Trans. Geosci. Remote Sens., vol. 55, no. 8, pp. 4355-4366, Aug. 2017.
- [99] Y. Li, Q. Li, Y. Liu, and W. Xie, "A spatial-spectral SIFT for hyperspectral image matching and classification," Pattern Recognit. Lett., vol. 127, pp. 18-26, Nov. 2019.
- [100] D. Hong, X. Wu, P. Ghamisi, J. Chanussot, N. Yokoya, and X. X. Zhu, "Invariant attribute profiles: A spatial-frequency joint feature extractor for hyperspectral image classification," IEEE Trans. Geosci. Remote Sens., vol. 58, no. 6, pp. 3791-3808, Jun. 2020.
- [101] M. E. Paoletti, J. M. Haut, J. Plaza, and A. Plaza, "Neighboring region dropout for hyperspectral image classification," IEEE Geosci. Remote Sens. Lett., vol. 17, no. 6, pp. 1032-1036, Jun. 2020.
- [102] J. Acquarelli, E. Marchiori, L. M. C. Buydens, T. Tran, and T. van Laarhoven, "Spectral-spatial classification of hyperspectral images: Three tricks and a new supervised learning setting," 2017, arXiv:1711.05512. [Online]. Available: http://arxiv.org/abs/1711.05512
- [103] W. Li, C. Chen, M. Zhang, H. Li, and Q. Du, "Data augmentation for hyperspectral image classification with deep CNN," IEEE Geosci. Remote Sens. Lett., vol. 16, no. 4, pp. 593-597, Apr. 2019.

- [104] J. M. Haut, M. E. Paoletti, J. Plaza, A. Plaza, and L. Plaza, "Hyperspectral image classification using random occlusion data augmentation," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 11, pp. 1751–1755, Nov. 2019.
- [105] S. Dieleman, K. W. Willett, and J. Dambre, "Rotation-invariant convolutional neural networks for galaxy morphology prediction," *Monthly Notices Roy. Astronomical Soc.*, vol. 450, no. 2, pp. 1441–1459, 2015.
- [106] T. Cohen and M. Welling, "Group equivariant convolutional networks," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 2990–2999.
  [107] T. S. Cohen and M. Welling, "Steerable CNNs," 2016,
- [107] T. S. Cohen and M. Welling, "Steerable CNNs," 2016, arXiv:1612.08498. [Online]. Available: http://arxiv.org/abs/1612.08498
- [108] M. E. Paoletti, J. M. Haut, J. Plaza, and A. Plaza, "Neural ordinary differential equations for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 3, pp. 1718–1734, Mar. 2020.
- [109] R. Kumar Srivastava, K. Greff, and J. Schmidhuber, "Highway networks," 2015, arXiv:1505.00387. [Online]. Available: http://arxiv.org/abs/1505.00387
- [110] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, arXiv:1502.03167. [Online]. Available: http://arxiv.org/abs/1502.03167
- [111] S. Santurkar, D. Tsipras, A. Ilyas, and A. Madry, "How does batch normalization help optimization?" in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 2483–2493.
- [112] K. Jarrett, K. Kavukcuoglu, M. A. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object recognition?" in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 2146–2153.
  [113] V. Nair and G. E. Hinton, "Rectified linear units improve restricted
- [113] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. ICML*, Jan. 2010, pp. 807–814. [Online]. Available: https://icml.cc/Conferences/2010/papers/432.pdf
- [114] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. 14th Int. Conf. Artif. Intell. Statist. (AISTATS)*, G. J. Gordon and D. B. Dunson, Eds., 2011, pp. 315–323.
- [115] C. Zhou, "Computation of optical flow using a neural network," in *Proc. IEEE Int. Conf. Neural Netw.*, vol. 2, Jul. 1988, pp. 71–78.
  [116] M. D. Zeiler, and R. Fergus, "Visualizing and understanding convolu-
- [116] M. D. Zeiler, and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham, Switzerland: Springer, Sep. 2014, pp. 818–833.
- [117] R. Gens and P. M. Domingos, "Deep symmetry networks," in Advances in Neural Information Processing Systems 27, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 2537–2545. [Online]. Available: http://papers.nips.cc/paper/5424-deep-symmetry-networks.pdf
- [118] S. Dieleman, J. De Fauw, and K. Kavukcuoglu, "Exploiting cyclic symmetry in convolutional neural networks," 2016, arXiv:1602.02660. [Online]. Available: http://arxiv.org/abs/1602.02660
- [119] M. Weiler, F. A. Hamprecht, and M. Storath, "Learning steerable filters for rotation equivariant CNNs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 849–858.
- [120] R. Ghosh and A. K. Gupta, "Scale steerable filters for locally scaleinvariant convolutional neural networks," 2019, arXiv:1906.03861.
   [Online]. Available: http://arxiv.org/abs/1906.03861
- [121] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, and M. Kudlur, "Tensorflow: A system for large-scale machine learning," in *Proc. 12th USENIX Symp. Operating Syst. Design Implement. (OSDI)*, 2016, pp. 265–283.
- [122] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, and A. Desmaison, "Pytorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 8026–8037.
- [123] 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," in *Proc. Optoelectron. Technol. Remote Sens. Space*, J. Seeley and S. Bowyer, Eds., Apr. 1988, pp. 134–141.
- [124] J. M. Haut, M. E. 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*), 2017, pp. 1063–2321.
- [125] B. Waske, S. van der Linden, J. A. 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.
- [126] R. Collobert and S. Bengio, "Links between Perceptrons, MLPs and SVMs," in *Proc. 21st Int. Conf. Mach. Learn. (ICML)*. New York, NY, USA: ACM Press, 2004, pp. 177–184. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1015330.1015415

- [127] W. Hu, Y. Huang, L. Wei, F. Zhang, and H. Li, "Deep convolutional neural networks for hyperspectral image classification," *J. Sensors*, vol. 2015, pp. 1–12, 2015.
- [128] W. Zhao, Z. Guo, J. Yue, X. Zhang, and L. Luo, "On combining multiscale deep learning features for the classification of hyperspectral remote sensing imagery," *Int. J. Remote Sens.*, vol. 36, no. 13, pp. 3368–3379, Jul. 2015.
- [129] J. Cohen, "A coefficient of agreement for nominal scales," *Educ. Psychol. Meas.*, vol. 20, no. 1, pp. 37–46, Apr. 1960.
- [130] 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.
- [131] J. Li, X. Zhao, Y. Li, Q. Du, B. Xi, and J. Hu, "Classification of hyperspectral imagery using a new fully convolutional neural network," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 2, pp. 292–296, Feb. 2018.



**MERCEDES E. PAOLETTI** (Member, IEEE) received the B.Sc. and M.Sc. degrees in computer engineering from the University of Extremadura, Cáceres, Spain, in 2014 and 2016, respectively, and the Ph.D. degree with an 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, Univer-

sity 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. She has been a Reviewer of the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING and the IEEE GEOSCIENCE AND REMOTE SENSING LETTERS. She was a recipient of the 2019 Outstanding Paper Award recognition in WHISPERS 2019 congress.



**JUAN M. HAUT** (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees in computer engineering from the Hyperspectral Computing Laboratory (HyperComp), Department of Technology of Computers and Communications, University of Extremadura, in 2011 and 2014, respectively, and the Ph.D. degree in information technology with an University Teacher Training Programme from the Spanish Ministry of Education, in 2019.

He is currently a member of the HyperComp Laboratory, Department of Technology of Computers and Communications, University of Extremadura. His research interests include remote sensing data processing and high-dimensional data analysis, applying machine (deep) learning, and cloud computing approaches. He has authored/coauthored more than 30 JCR journal articles (19 in IEEE journals) and 20 peer-reviewed conference proceeding papers. Some of his contributions have been recognized as hot-topic publication for their impact on the scientific community.

Dr. Haut was a recipient of the Outstanding Ph.D. Award at the University of Extremadura, in 2019. He was also a recipient of the Outstanding Paper Award in WHISPERS 2019 congress. From his experience as a Reviewer, it is worth mentioning his active collaboration in more than ten scientific journals, such as 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, being awarded with the Best Reviewers recognition of the IEEE GEOSCIENCE AND REMOTE SENSING LETTERS in 2018. Furthermore, he has guest-edited three Special Issues on Hyperspectral Remote Sensing for different journals. He is also an Associate Editor of the IEEE GEOSCIENCE AND REMOTE SENSING LETTERS and the IEEE JOURNAL ON MINIATURIZATION FOR AIR AND SPACE SYSTEMS.



**SWALPA KUMAR ROY** (Student Member, IEEE) received the bachelor's degree in computer science and engineering from the West Bengal University of Technology, Kolkata, India, in 2012, and the master's degree in computer science and engineering from the Indian Institute of Engineering Science and Technology, Shibpur, Howrah, India, in 2015. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, University of Calcutta, Kolkata.

He was a Project Linked Person with the Computer Vision and Pattern Recognition Unit, Indian Statistical Institute, Kolkata, from July 2015 to March 2016. He is currently an Assistant Professor with the Department of Computer Science and Engineering, Jalpaiguri Government Engineering College, Jalpaiguri, India. His research interests include computer vision, deep learning, remote sensing, texture feature description, and fractal image coding.



**ELIGIUS M. T. HENDRIX** is currently a European Researcher and a Professor with 30 years of experience in mathematical modeling and optimization questions. He is also affiliated with Wageningen University, Wageningen, The Netherlands, and the Universidad de Málaga, Málaga, Spain. He has published more than 70 journal articles. His research interest includes how to use the mathematical structure of an optimization application in order to derive specific solution methods

and algorithms. His current research interest includes how to adapt algorithms such that they can exploit modern computer structures.

...