Deep Learning for Classification of Hyperspectral Data: A Comparative Review

Nicolas Audebert, Bertrand Le Saux, *Member, IEEE* and Sébastien Lefèvre

Abstract—In recent years, deep learning techniques revolutionized the way remote sensing data are processed. Classification of hyperspectral data is no exception to the rule, but has intrinsic specificities which make application of deep learning less straightforward than with other optical data. This article presents a state of the art of previous machine learning approaches, reviews the various deep learning approaches currently proposed for hyperspectral classification, and identifies the problems and difficulties which arise to implement deep neural networks for this task. In particular, the issues of spatial and spectral resolution, data volume, and transfer of models from multimedia images to hyperspectral data are addressed. Additionally, a comparative study of various families of network architectures is provided and a software toolbox is publicly released to allow experimenting with these methods.[1] This article is intended for both data scientists with interest in hyperspectral data and remote sensing experts eager to apply deep learning techniques to their own dataset.

I. INTRODUCTION

Thanks to recent advances in deep learning for image processing and pattern recognition, remote sensing data classification progressed tremendously in the last few years. In particular, standard optical imagery (Red-Green-Blue –RGB– and Infra-Red –IR–) benefited from using deep convolutional neural networks (CNN) for tasks such as classification, object detection or semantic segmentation [1], [2], [3]. This was made possible by the transfer of models developed in computer vision, which focuses mostly on images encoded on three channels. However, remote sensing relies often on multispectral imagery (coming from satellites such as Sentinel-2 or Landsat, or from airborne sensors) which allows to capture simultaneously the radiance at several wavelength bands. Hyperspectral imaging (HSI) data are a subset of multispectral data for which the wavelength resolution is fine, wavelength bands are contiguous and their range is particularly high. It makes possible a precise analysis of soils and materials [4], [5].

Indeed, the high spectral resolution allows to characterize precisely the electromagnetic spectrum of an object. However, most hyperspectral sensors have a low spatial resolution so deep learning techniques designed for computer vision are not easily transferable to hyperspectral data cubes since the spectral dimension prevails over the spatial neighborhood in most cases. If compared with today's optical images, the

volume of data is similar but the structure is completely different. Moreover, the low spatial resolution actually limits the number of samples available for training statistical models. This also makes more difficult the annotation process which is required in supervised learning, since objects smaller than the spatial resolution are mixed with their neighborhood. These two points are the main challenges to take up in order to use deep learning for hyperspectral image processing.

This article aims at bridging the gap between data scientists and hyperspectral remote sensing experts. Thus, it is more focused than previous reviews on deep learning [6] while presenting hyperspectral peculiarities from a deep learning point of view, different from [7]. We first summarize some principles of hyperspectral imaging and list some reference datasets available for public use. We then review some standard machine learning techniques used for classification before focusing on recent works with deep learning, where the comparison of existing networks is supported by experimental analysis. We finally conclude with an emphasis on the emerging research axes.

II. HYPERSPECTRAL IMAGING: PRINCIPLES AND RESOURCES

A. Hyperspectral imaging principles in a nutshell

Hyperspectral sensors measure the intensity of the radiant flux for a given surface and a given wavelength, i.e. a physical quantity in watts per squared meter steradian ($W/(sr.m^2)$). Precisely, per each surface unit (which corresponds to a pixel of the image) the sensor captures light emitted and reflected by the object as a spectrum of several hundreds of channels, which defines a spectral response curve.

In the context of Earth Observation, signals coming from the Earth surface are changed by atmospheric perturbations such as clouds, water vapor atmospheric aerosols, etc. So, for remote sensing of surfaces and land cover, the reflectance is preferably used, defined as the ratio between the emitted flux of the surface and the incidental flux. This ratio gives the reflecting effectiveness of a given object for each light wavelength band. Reflectance is an intrinsic property of the materials, independently of the environment, and thus is highly discriminative for classification purposes.

To compensate for the atmospheric perturbations, several atmospheric correction methods have been developed [8], [9], [10] in order to obtain measures able to characterize the land cover [11]. They take the intensity images as input and produce reflectance images, by getting rid of the light diffusion effects and radiative phenomena of the atmosphere. They are based on elaborate physics models and take into account acquisition

N. Audebert and B. Le Saux are with DTIS, ONERA, University Paris Saclay, F-91123 Palaiseau - France. E-mails: nicolas.audebert@onera.fr, bertrand.le_saux@onera.fr.

S. Lefèvre is with Univ. Bretagne Sud, UMR 6074, IRISA, F-56000 Vannes, France. E-mail: sebastien.lefevre@irisa.fr

[1] https://github.com/nshaud/DeepHyperX

parameters such as the sunlight levels or the local digital surface model to deal with multiple-reflection phenomena.

In practice, hyperspectral images are (w, h, B) tensors, i.e. three-dimensional cubes with two spatial dimensions (width w and height h) and a spectral one (with B bands). Compared to volumetric data, e.g. seismic data cubes, this so-called hypercube is anisotropic: the three dimensions do not represent the same physical displacement. However, all values in the hypercube are expressed in the same unit, either light intensities or reflectances, which makes linear operations on a 3D subset of the cube mathematically and physically valid. This property will come in handy when dealing with convolutions and filtering operations on the hypercube.

B. Reference datasets

Several public datasets captured using an hyperspectral sensor have been made available to the scientific community. They usually come along with annotations to evaluate the classification performances. Pavia, Indian Pines and Data Fusion Contest 2018 are presented in details in the following. Other standard ones include Salinas (captured with the AVIRIS sensor over the Salinas Valley, Cal., USA and composed of classes such as vegetable cultures, vineyeard and bare soils), Kennedy Space Center (also with AVIRIS over the Kennedy Space Center, with classes such as wetlands and various types of wild vegetation) and Botswana (captured with the Hyperion sensor from the EO'1 satellite, with 14 classes of vegetation and swamps over the Okavango delta). While characteristics of all are listed and compared in Table I, a few ones are now presented in details.

1) Pavia: Pavia is a dataset captured using the ROSIS sensor with a ground sample distance (GSD) of 1.3 m over the city of Pavia, Italy (cf. Fig. 1). It is divided in two parts: Pavia University (103 bands, 610×340px) and Pavia Center (102 bands, 1096×715px). 9 classes of interest are annotated covering 50% of the whole surface. They comprise various urban materials (such as bricks, asphalt, metals), water and vegetation.

This has been for long one of the main reference datasets because it is one of the largest labeled HSI data and because it allows to evaluate the use of HSI for potential applications. However, some preprocessing might be necessary in order to remove some pixels with no spectral information and a few errors occur in the ground-truth.
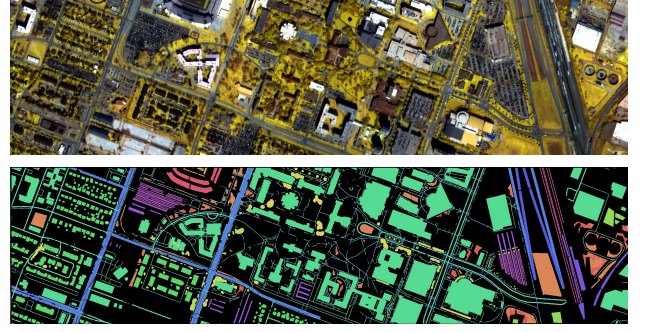

Fig. 1: Pavia University (natural composite image).

Fig. 2: Data Fusion Contest 2018 [12] over Houston: composite image with bands 48, 32 and 16 (top) and ground-truth (bottom row).

2) Data Fusion Contest 2018 (DFC2018): The DFC2018 hyperspectral data (cf. Fig. 2) was acquired over Central Houston, Texas, USA, using an airborne sensor. It covers a [380–1,050] nm spectral range over 48 contiguous bands at 1 m GSD. 20 classes of interest are defined and include not only urban categories (buildings and roads of various types, railways, cars, trains, etc.) but also various vegetation types (stressed, healthy, deciduous or evergreen trees) and specific materials. This dataset is part of the 2018 Data Fusion Contest release, along with very-high-resolution imagery and multispectral LiDAR [12], [13].

3) Indian Pines: Indian Pines is a dataset captured using the AVIRIS sensor. The scene covers agricultural areas in North-Western Indiana, USA, with a ground sampling distance (GSD) of 20 m/px, resulting in a 145×145px image with 224 spectral bands. Most parts of the image represent fields with various crops while the rest denotes forests and dense vegetation. 16 classes (cf. Fig. 3 and 4) are labeled (e.g. corn, grass, soybean, woods, etc.),

Fig. 3: Indian Pines (natural composite image).

some of them being very rare (less than 100 samples for alfalfa or oats). Water absorption bands (104→108, 150→163 and 220) are usually removed before processing. In spite of its limited size, it is one of the main reference datasets of the community. Though, rare classes are usually not taken into account when evaluating classification algorithms.
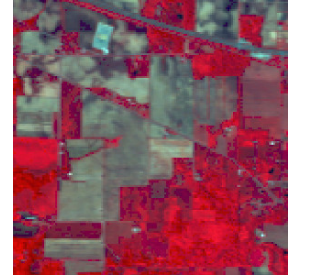
4) Dataset summary: The various dataset statistics and informations are compiled in Table I. This highlights that the main issue with applying machine learning approaches to hyperspectral data lies in the small number of available samples. Existing datasets are small with respect to standard optical imagery. Moreover, due to sensor diversity and post-processing methods, it is not possible to train algorithms simultaneously on different datasets.

Nevertheless these datasets are available and have been shared among researchers for years thanks to the good will
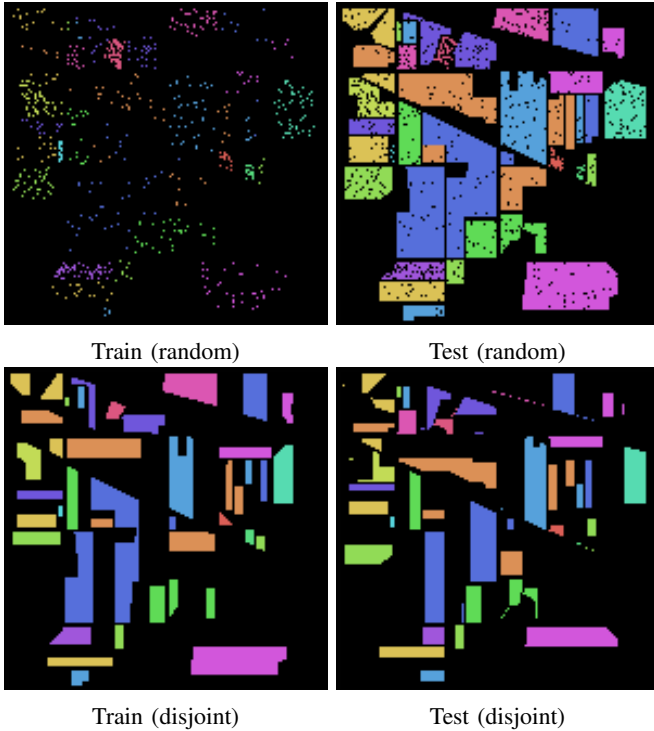
Train (random)	Test (random)
Train (disjoint)	Test (disjoint)

Fig. 4: Examples of train/test splits on the Indian Pines dataset.

TABLE I: Main public labeled datasets in hyperspectral imaging.

Dataset	Pixels	Bands	Range	GSD	Labels	Classes	Mode
Pavia (U & C)	991,040	103	0.43-0.85 μm	1.3 m	50,232	9	Aerial
Indian Pines	21,025	224	0.4-2.5 μm	20 m	10,249	16	Aerial
Salinas	111,104	227	0.4-2.5 μm	3.7 m	54,129	16	Aerial
KSC	314,368	176	0.4-2.5 μm	18 m	5,211	13	Aerial
Botswana	377,856	145	0.4-2.5 μm	30 m	3,248	14	Satellite
DFC 2018	5,014,744	48	0.38-1.05 μm	1 m	547,807	20	Aerial

of some research groups over the World [2]. Moreover, IEEE GRSS is providing the community with the GRSS Data and Algorithm Standard Evaluation (DASE) website [3]. On DASE, researchers can access the data for Indian Pines, Pavia and DFC2018, and submit classification maps which are evaluated on-line. For each dataset, a leader-board allows to compare the state-of-the-art methods as soon as they are tested.

III. HYPERSPECTRAL IMAGE ANALYSIS ISSUES AND STANDARD APPROACHES

This section briefly recalls standard issues and current approaches for hyperspectral data processing. Especially, supervised statistical learning approaches for classification are detailed since they are obvious reference baselines and inspiration for deep learning-based methods.

A. Pre-processing and normalization

Working with hyperspectral images often implies pre-processing the data. Besides the aforementioned atmospheric

and geometric corrections, band selection and normalization are often also applied. Those normalizations will impact how classifiers are able to separate spectral features in the ways described below.

1) Band selection: Depending on the sensor, some spectral bands might be difficult to process or contain outliers which modify the spectrum dynamics. For example, we often remove bands related to water absorption, bands with a low signal-to-noise ratio and saturated values. Not only this improves the robustness of the classifiers by alleviating the noise present in the data, this also helps fight against the well-known curse of dimensionality that provokes decreasing performances of statistical classification models when the dimensions of the data increases. Band selection can also be used by dropping uninformative bands, e.g. using Principal Component Analysis (PCA) [14] or mutual information [15]. However, band selection should be done carefully. Unsupervised dimension reduction can sometimes lead to worse performance than using the raw data since it might remove information that is not useful for compression but was discriminant for classification [16]

2) Statistical normalization: It is a common practice in the machine learning community to normalize the data beforehand to rely on common assumptions for which classifiers are known to behave well, such as zero-mean and unit-variance. Standard strategies often allow to significantly improve processing with statistical approaches. We denote by X_i individual spectra and by I the whole image. These strategies then are:

- Using the spectral angle, the normalized variant of the spectrum with a unit Euclidean norm: $X = X/\|X\|$; The angle between two spectra is a common similarity measure used for classification, notably in the popular Spectral Angle Mapper (SAM) classifier [17].
- Normalizing first and second-order moments (so that to obtain a zero mean and unit variance). This can be done for each band independently, which works especially well with classifiers that expect all features to have similar amplitudes, such as Support Vector Machines. However, this squashes the dynamics in the spectral dimension. Alternatively, the normalization can be done globally on the whole image: $I = \frac{I - m_I}{\sqrt{\sigma_I}}$;
- Converting the dynamics to $[0, 1]$ using $I = \frac{I - min(I)}{max(I) - min(I)}$. This is helpful mostly for numerical optimization that rarely behave well with very large values and relates more to the implementation than the theoretical standpoint. Once again, this can be applied for each band, at the risk of making relative amplitudes disappear and squashing dynamics, or globally on the whole image.

Finally, in order to minimize the influence of outliers and obtain balanced image dynamics, it is also popular to clip on image values over a given threshold: either values over the last decile or outside the range $m_I \pm 2 \times \sqrt{\sigma_I}$.

B. Spectral classification

The most straightforward approach for classifying hyperspectral data is to consider it as a set of 1D spectra. This

makes sense given its fine spectral resolution but low spatial resolution. Each pixel corresponds to a spectral signature, that is a discrete signal to which a statistical model can be fit. In the sequel, we restrict our considerations to machine learning approaches with little or not expert processing, but those previously cited.

1) Unmixing: In an ideal case, a pixel in an hyperspectral image corresponds to the reflectance of a material observed over a surface unit. However, the actual spatial resolution of HSI implies that often a pixel corresponds to a surface made of several various materials which produce a spectra mixture. Formally, if we denote by $S_1, \ldots, S_n$ the pure spectra (end-members) of the set of materials in the scene, then for a pixel of coordinates (i, j), the locally observed spectrum is a function F of S_i:

$$\phi_{i,j} = F(S_1, \ldots, S_n) \simeq \sum_{k=1}^{n} \lambda_k S_k .$$

Under the hypothesis of a plane surface, we may assume F is a mere linear combination where weight coefficients λ_k correspond to the proportion of material k in the observed area.

One way to perform hyperspectral data classification is unmixing [18], that is finding the individual materials in the observed area by computing their abundance maps. Reference spectra of pure materials are called end-members [4] and constitute a decomposition basis of mixed spectra. Abundance maps correspond to the proportional contributions of the end-members to each pixel, that are the decomposition coefficients of the mixture. Usually, given the pure spectra S_k and image Φ, it is possible to invert the linear system to obtain the coefficients λ_k for each point, and thus abundance maps. Such approaches rely on linear algebra and numerical methods for solving inverse problems. However, learning-based approaches are also used, for instance clustering methods to find unknown end-members.

2) Dimensionality reduction: Much works are dedicated to reducing the dimension of spectra. Indeed, given the spatial resolution, neighbor intensities are highly correlated so a spectral signature contains a lot of redundant information. The main issue consists in extracting the discriminative information to reduce the set of relevant bands [19].Among the various pre-processing methods for feature generation which have been used, we can cite feature extraction algorithms such as the well-known principal component analysis (PCA) [20] or more recently random feature selection [21]. An other approach consists in computing some indices which integrate physical priors about the band response, such as the Normalized Difference Vegetation Index (NDVI) or the Normalized Difference Water Index (NDWI). This differs from band selection since the resulting features are not reflectances or intensities anymore. Instead, they are a new representation of the data in a space that will be suitable for classification.

Classification is then processed in a standard way, by using common statistical models: decision trees and random

---

⁴A mineralogy term form pure minerals, by opposition to most minerals which exist as solid solutions.

forests [22], support vector machines (SVM) [23], [24], etc. Approaches such as manifold learning also fit this framework [25]. The goal of dimension reduction is to tackle the curse of dimensionality and to simplify the representation space to make the learning stage easier.

C. Spatial-spectral classification

If a spectral-only approach might work, it is not satisfying since it does not benefit from the spatial structure of hyperspectral images. Indeed, it is likely that neighboring pixels may share some structural relationships (e.g. buildings usually have polygon shapes while vegetation has a fractal-like appearance). Taking the spatial aspect into account during the analysis improves the model robustness and efficiency thanks to these structural dependencies. Three main approaches can be distinguished based on when the spatial aspect is considered in the classification process.

1) Spatial regularization: A popular technique consists to classify individual spectra first, then to regularize the resulting classification with a spatially-structured model such as Markov Random Fields (MRF) or Conditional Random Fields (CRF) [26]. Spatial regularization is then a supervised post-processing.

2) Pre-segmentation: An alternate approach consists in performing spatial regularization as an unsupervised pre-processing. Various methods [27], [28], [29] propose to first segment the hyperspectral image, then to aggregate spectrum-wise features for each segmented region in order to enforce local consistence. Hierarchical segmentations, e.g. tree-like segmentations, are often used to derive local morphological features, such as in the popular morphological attribute profile approach [30].

3) Joint learning: The last approach learns simultaneously spatial and spectral features by using specific kernels. It takes roots in the inspiring works of [31] to compute end-members by benefiting from the correlation between spatially close pixels and [32] by exploiting a mixture of spatial and spectral classifiers. More recent approaches focus on statistical models able to learn directly over local neighborhoods (fixed or adaptive) how to extract combined spectral and spatial features. In particular, [33] introduced the possibility to design spatial-spectral kernels for SVMs able to handle hyperspectral data. This technique will then be largely adopted in later works [34], [35], [36]. With a similar objective, [37] proposes a method to choose automatically the filters which lead to the most efficient features for hyperspectral data classification from a random-filter bank.

The main limitation of traditional shallow learning methods stems from the feature engineering required to improve the classifier's performances. Indeed, spectra from different classes have to be separated in the feature space which can be challenging to achieve. In comparison, deep learning is focused on representation learning, i.e. automatically designing a feature space that is tailored to the objective task. This reduces significantly the need for feature engineering and hopefully should improve performances since both representation and classification will be jointly optimized.

IV. Deep Learning for Hyperspectral Data

Recent works use deep learning techniques for classifying hyperspectral images. The review which follows is organized so as to identify the main families of methods.

A. Pre-processing and normalization

Pre-processing and normalization processes used for deep learning are similar to the ones used for standard machine learning. However, it is worth noting that most works do not use band selection or saturated spectrum removal but on the contrary rely on the robustness of neural networks.

For unmixing, a now standard, unsupervised approach [38], [39] consists in a network with two stacked auto-encoders: the first one is used for denoising while the second one does the actual unmixing by enforcing a sparsity constraint.

B. Spectral classification

1) Supervised learning: The most straightforward evolution from shallow machine learning to deep learning is using a deep fully-connected network instead of a standard classifier (SVM or Random Forest). The principle remains the same, but the network may thematically model the task in a finer way and with a better discrimination capacity. This has been implemented since the 2000s [40], [41] with small networks, and brought up to date recently by [42] with unidimensional CNN which learn a filter collection to be applied on individual spectra. However, processing sequential data can also be done using Recurrent Neural Networks (RNN). RNN use memory to retrieve past information and are often used to process time series. [43] suggests to use these RNN to classify hyperspectral data by assuming that these models can efficiently model both long-range and short-range dependencies in the spectral domain. A similar approach using RNN treating hyperspectral pixels as a sequence of reflectances was introduced in [44], including a pseudo-labeling scheme for semi-supervised learning. Finally, an approach using both the recurrent and convolutional aspects has been proposed by [45], in which the filtered features are finally processed by recurrent layers.

2) Unsupervised learning: One of the most important benefits of deep learning for processing hyperspectral data is the introduction of auto-encoders. Indeed, the band selection problem and more generally dimensionality reduction can be considered as a data compression issue. Within this perspective, auto-encoders allow to learn a smart compression with minimal information loss, for example more efficient than a standard PCA. Thus [46] and later [47] proposed dimension reduction through cascade of auto-encoders for denoising followed by classification with a simple perceptron.

C. Spatial-spectral approaches

1) 1D or 2D Convolutional Neural Networks: A more recent approach grounds in convolutional networks so popular in multimedia vision: Convolutional Neural Networks (CNN). In computer vision, most CNN are designed using a first part which is convolutional and performs the feature extraction and
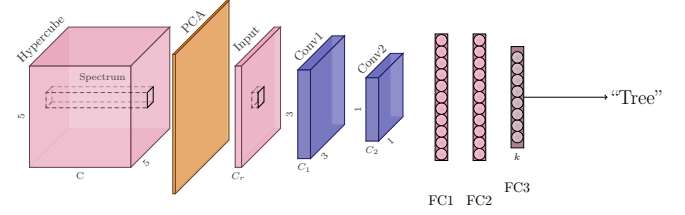

Fig. 5: 2D CNN for classification of hyperspectral data proposed in [48].

representation learning, and a second part which is fully connected and performs the classification. However, the number of filters is proportional to the number of input channels, e.g. for a first convolutional layer with a kernel 5×5 and n output channels, there will be $5 \times 5 \times n \times 3$ for an RGB image (3 channels) but $5 \times 5 \times n \times 100$ for common hyperspectral images (with 100 bands). Therefore, a popular approach to transpose deep convolutional networks to hyperspectral imaging consists in reducing the spectral dimension in order to close the gap between hyperspectral and RGB images.

a) Supervised learning: Thus, several works proposed a CNN for hyperspectral data classification. [48] uses a PCA to project the hyperspectral data into a 3-channel tensor to then perform the classification using a standard 2D CNN architecture, as shown in Figure 5. The architecture alternates convolutions and dimension reduction (either by PCA or by sampling) followed by a multi-layer perceptron for the final classification step, as shown in Figure 5, with 2D convolutions in red and 1D fully connected layers in blue. As an alternative, [50] flattens the spatial dimensions to produce a 2D image with a different shape, instead of an hypercube, and then applies a traditional 2D CNN on the resulting image. One drawback of these methods is that they try to make hyperspectral images similar to RGB ones, i.e. to force hyperspectral data into the multimedia computer vision framework. However, the specific properties of hyperspectral imaging might be wasted when doing so, especially when using unsupervised and uncontrolled dimension reduction.

Trying to go around this problem, [51], [52] propose another approach. Instead of dealing with the hypercube as a whole, they introduce two CNN: a 1D CNN that extracts a spectral feature along the radiometric dimension, and a 2D CNN that learns spatial features as previously described. The features from the two models are then concatenated and fed to a classifier to perform a spatial-spectral classification as usual.

b) Unsupervised learning: It is worth noting that this kind of models has also been extended for the semi-supervised and unsupervised setups. [53] introduced 2D multi-scale convolutional auto-encoders for simultaneous embedding learning of hyperspectral data including both spatial and spectral information and classification, while [54] proposed the use of CNN for unsupervised feature extraction: it performs reduction dimension of spectra by using also the spatial information using a regularizing sparse constraint.

Similarly to standard approaches, a first approach to spatial-spectral classification consists in representing a pixel by a

Fig. 6: 2D+1D CNN for classification of hyperspectral data with a residual learning mechanism as proposed in [49].

descriptor with two parts :

- a spectral feature computed on the radiometric spectrum;
- a spatial feature computed on the local neighborhood of the considered pixel.

A standard descriptor is a vector obtained from the concatenation of the complete spectrum and the spatial feature obtained by PCA on a $w \times h$-sized neighborhood around the pixel in all bands, from which the K principal components are kept (usually $w = h \simeq 8$ and $K = 3$). This descriptor is then processed by a deep classifier, often unsupervised such as Deep Belief Networks in [55], [56], Restricted Boltzmann Machines in [57], [58], or cascades of auto-encoders in [59], [60], [61], [62].

Overall, while the previous approaches introduced deep learning into the hyperspectral imaging framework, they did not fully leveraged the representation learning ability of end-to-end deep networks. Indeed, *ad hoc* processing of the hypercube, either by splitting the dimensions or by unsupervised dimension reduction, could be replaced by learned counterparts.

2) 2D+1D CNN:

a) Supervised learning: Indeed, the importance of spectral information leads to new approaches dealing globally with the hyperspectral cube. The idea is to process directly the hypercube using an end-to-end deep learning process. To answer the problem of the spectral dimension reduction, several works tried to design CNN alternating spatial convolutions with spectral ones to regularly reduce the size of the feature maps.

In particular [63] introduces a CNN which consider both the spatial and spectral neighborhoods of a given pixel, i.e. that takes a 3D patch as an input. The first layers reduce the spectral dimension using a $1 \times 1 \times n$ kernel, then the spatial ones with a $k \times k \times 1$ kernel, and so on. Eventually, two fully-connected layers perform the final classification step. This allows them to compute feature maps where both spectral and spatial representations are learned in an alternate way.

In a similar idea, [64] suggests an alternative approach that performs spatial-spectral convolutions in the first layer to perform spectral dimension reduction, similarly to what could

be expected from PCA, albeit supervised and including spatial knowledge. Deeper layers form a traditional 2D CNN that performs as usual.

b) Unsupervised learning.: On the unsupervised side, [65] introduce a 2D CNN with a residual learning paradigm [66] that is able to learn an efficient low-dimension representation of the hyperspectral pixels and their neighborhood.

Finally, [49] proposes a Fully Convolutional Network (FCN) which handles N bands. The first layer extracts a multi-scale spatial-spectral feature using a module inspired by the Inception architecture [67]. The model applies in the raw data several convolutions with an increasing kernel size in the spatial dimensions, i.e. $1 \times 1 \times N$, $3 \times 3 \times N$ and $3 \times 3 \times N$ where N is the number of bands. This reduces the spectral dimension and also performs a multi-scale filtering of the data. These resulting activation maps are then fed to a succession of non-linearities and 1D convolutions that project the feature maps in the final classification space. This architecture is illustrated in Figure 6, with 1D convolution in pink, 2D convolutions in blue and pooling in orange. It is worth noting that thanks to its Fully Convolutional architecture, this network generates predictions for all pixels in the input patch, and not only the central one. This means that it is more efficient at inference time when sliding over the whole image.

3) 3D CNN: If spatial-spectral methods already reach particularly satisfying results, especially with respect to the spatial regularity, they require a high level of engineering in their design which is not fully compatible with the "data-to-output" motto that defines deep learning. A promising approach [68] handles directly the hyperspectral cube with 3D CNN which work simultaneously on the three dimensions using 3D convolutions. This conceptually simpler approach slightly improves the classification performances with respect to 2D+1D models. Many architectures have been proposed to handle 3D convolutional neural networks for hyperspectral data, mostly to investigate well-known techniques from deep learning for computer vision, such as multi-scale feature extraction [69] and semi-supervision [70]. Instead of producing 2D feature maps, these 3D CNN produce 3D feature cubes

that are well-suited for pattern recognition in a volume and seem at least theoretically more relevant for hyperspectral image classification. [71] especially showed that 3D CNN for classification of hyperspectral images obtained higher performances than their 2D counterparts.

Indeed, with respect to spectral or 2D+1D CNN, 3D CNN combine those two pattern recognition strategies into one filter, requiring less parameters and layers. They can learn to recognize more complex 3D patterns of reflectances: co-located spectral signatures, various differences of absorption between bands, etc. However, all directions are not equivalent in the hyperspectral data cube, so these patterns cannot be processed as mere volumetric data. As for 2D+1D CNN, it implies extra care when designing the network like using anisotropic filters.

D. Comparative Study of Deep Learning Architectures for Hyperspectral Data

1) Deep Learning for Hyperspectral Toolbox: Despite the abundant literature on data driven hyperspectral image processing, there are only few available tools that can be used to compare the state of the art methods in a common framework. To this end, we provide DeepHyperX [5], a deep learning toolbox based on the PyTorch framework that can be used to benchmark neural networks on various public hyperspectral datasets. It includes many supervised approaches, ranging from linear SVM to state of the art 3D CNN. Indeed, many models from the literature are implemented in the toolbox, including 1D CNN for spectral classification (cf. Fig. 8, with 1D convolutions in pink, pooling in orange and fully connected layers in blue) and the most recent 2D and 3D CNN as suggested in [63], [71], [49], [68] (cf. Fig. 7 for a 3D convolutional network architecture example, with 3D convolutions in green, 3D pooling in orange and 1D fully connected layers in blue). The models can be trained and evaluated on several datasets from the literature, e.g. Pavia Center/University, Indian Pines, Kennedy Space Center or DFC2018, described in Sec. II-B. Many hyper-parameters can be tuned to assess the impact of spatial features, the size of the training set or the optimization strategy. This toolbox allows us to evaluate how the methods from the state of the art compare in different use cases.

Technically, this toolbox is written in Python and consists in an interface around the PyTorch and scikit-learn libraries. Deep networks are implemented using PyTorch that can leverage both CPU and GPU based on the user needs, while we rely on scikit-learn for the SVM implementations. Several public classification datasets are pre-configured for easy investigation of deep models. The modular architecture of the toolbox allow users to easily add new remote sensing datasets and new deep architectures.

2) Best practices: Designing and optimizing deep networks can seem arcane, especially considering the abundant literature on this topic that has been published since 2010. However, a principled approach can drastically ease the process of experimenting deep learning on novel applications. A deep learning

experiment can be divided in three broad stages: building a model, optimizing the model and running inferences. We are assuming here that the model will be used for classification using a classification loss function such as the cross-entropy.

Building a model can often be summed up as choosing a model from the literature. Most of the time, it is better to rely on a well-validated architecture from the state of the art than spending time designing one from scratch. Based on the dataset and the application, choosing a model can be done on the following criterion:

- If the dataset is an image that presents spatial correlations, a 2D approach will outperform pixel-wise classifiers in most cases. For hyperspectral images, 3D CNN will be able to leverage correlations in all three dimensions of the data.
- Bigger models mean more parameters to optimize requiring in turn more training samples.
- Large convolutional kernels tend to be slower, especially in 3D. Most implementations are optimized for small 2D kernels.
- Fully Convolutional Networks are more efficient since they can predict several pixels at a time. Moreover, the absence of fully connected layers means that they will have a lot less parameters, therefore being easier to train.
- Non-saturating activation functions such as ReLU alleviate vanishing gradients and help build deeper networks while being faster to compute than sigmoid or $\tanh$ alternatives.

During training, the algorithm of choice for optimizing the network's weights is the backpropagation algorithm [72]. This technique relies on the stochastic gradient descent (SGD) or one of its variants. Many versions have been proposed in the literature, using more or less hyperparameters. The fundamental hyperparameter is the learning rate α which controls the magnitude of the weight update after each iteration. A too high α will cause the loss to diverge or to oscillate around the local minimum without reaching it, while a too small α will be very slow to converge. In practice, it is recommended to train with the highest α that makes not the loss diverge at first, and then slowly decreasing it during the training. For example, our toolbox uses an adaptive policy that divides α by one order of magnitude when the validation error plateaus. An alternative is to use an SGD variant with an adaptive learning rate, such as the popular [73].

Dealing with the backpropagation with a large number of weights can be complex. Optimizing deep networks involves working with many parameters. One fundamental best practice that is often overlooked is the virtue of initialization. Weights are randomly initialized at the start of the training, but various strategies have been developed so that the SGD starting point has better convergence properties. The initialization policy from [74] is especially suited for CNN including ReLU activation. Also, due to their large number of weights, the fully connected layers are often prone to overfitting. Dropout [75] significantly alleviates this phenomenon. Moreover, deeper networks often benefit from a larger batch size and the use of Batch Normalization [76], which smooths the loss landscape.

Training sample preparation also includes a few practices that can be applied for better performances. First, shuffling the dataset after each epoch helps avoiding recurring patterns in the SGD and overall makes the optimization smoother. Data augmentation is especially useful to introduce equivariances. For example, in image data, horizontal and vertical symmetries can be applied by flipping the training patches randomly during the training. This increases the diversity of the examples and the robustness of the model. Moreover, many datasets present large class imbalance, where one or a few classes dominate the labels. One simple solution is to weight according to the loss function in order to penalize more the less-occurring classes. The inverse-median frequency class weighting is commonly used to do so, e.g. in semantic segmentation. This is equivalent to showing more examples from the rarer classes to the model.

It is fundamental to be careful when tuning the optimization hyperparameters. Their choice should be based on a validation set that is not the same as the test set, or the test results will be optimistic. If this is not possible, a cross-validation over several train/test splits helps to assess how robust the hyperparameters are in order to avoid overfitting.

Finally, during inference it is recommended to use the network that reached the best validation score and not necessarily the last epoch weights. This implies saving regular checkpoints during the training.

We tried our best in our toolbox to apply these best practices while letting advanced users use their own parameters where needed.

3) Experiments: In this section, we compare several deep architectures from the literature for hyperspectral image classification in a remote sensing context. To the best of our knowledge, there have been none principled analysis of the various deep convolutional networks introduced in the past. Indeed, works from the literature perform experiments using slightly different setups:

- Most papers divide the datasets in a train and test splits by randomly sampling over the whole image. A few papers (e.g. [43], [77]) use the standard train/test split from the IEEE GRSS DASE initiative.
- Some authors consider only a subset of the classes. This is especially prominent for Indian Pines, where the classes with less than 100 samples are often excluded, e.g. in [42], [49].
- Even when the train/test splits are done the same way, the number of samples in the train set might vary. Some authors use 20% of all the training set, while others use a fixed amount of samples for each (e.g. 200 samples for each class).
- Some authors further divide the training set into a proper training set and a validation set for hyperparameters tuning, while others perform the tuning directly on the test set.

In this work, we argue that randomly sampling the training samples over the whole image is not a realistic use case. Moreover, we affirm that it is a poor indication of generalization power. Indeed, neighboring pixels will be highly correlated, which means that the test set will be very close to the train set.

To demonstrate this, we consider a nearest-neighbor baseline using randomly sampled training pixels, and another using co-located training pixels well-separated from the test set.

Also, in the case of 2D and 3D approaches, especially Convolutional Neural Networks, the receptive field of the network might involuntarily includes test samples in the training set. Indeed, the first convolutional layer also sees the neighbors of the central pixel which might be in the test set if the sampling has not been carefully checked. An example of this is illustrated in Figure 9.

Following the standards from the machine learning for remote sensing community, we perform our experiments by using well-defined train/test splits where the samples are extracted from significantly disjoint parts of the image. In the case of 3D CNN, it ensures that no pixel from the test set will be surreptitiously introduced in the train set. To do so, we use the train/test splits for Indian Pines, Pavia University and the DFC2018 as defined by the IEEE GRSS on the DASE benchmarking website. The ground truth is divided based on the connected components instead of the pixels, which allows the evaluation to actually measure how the model generalizes to new geo-entities. Hyperparameters are tuned using 5% of the train set as a separated validation set.

We use our toolbox to compare various reimplementations of the state of the art. Models have been implemented as close as possible to the original papers [6] We list below the models compared and the changes we performed, if any:

- 1D CNN from [42]. As the optimizer is not specified in the original paper, we use the standard SGD with momentum.
- 1D RNN [43]. The authors experiment both with the standard $tanh$ non-linearity and a parameterized variant. We use the former.
- 2D+1D CNN [63]. No modification.
- 3D CNN [68]. No modification.

The 2D+1D and 3D CNN are trained using patches of 5×5 pixels. The models are compared with two baselines: an SVM –obtained by grid search– and a fully-connected 1D neural network with three layers interlaced with Dropout [75] using the ReLU non-linearity [78]. To compensate the class unbalance in the three datasets, we use inverse median frequency for all models. We also use vertical and horizontal symmetries as data augmentation for the 2D and 3D CNN. We report in Table II the overall accuracy and Cohen's kappa on the three datasets. Experiments are repeated 5 times on Pavia University and Indian Pines, though only once on the DFC2018 given its much larger size.

As could be expected, we obtain results significantly lower than those reported in the literature since we use strongly disjoint training and testing sets. Table III reports the results of our implementation, the original score reported by the authors

---

[6]It is worth noting that despite the authors' best efforts, reproducing exactly the results using only the papers and without the original implementations is very difficult. Many hyperparameters and implementation tricks are omitted in the manuscripts, and can only be guessed when trying to reproduce the results. Although this is not specific to hyperspectral data, providing code along with papers has been critical to the success of deep learning in many other areas such as computer vision and natural language processing.
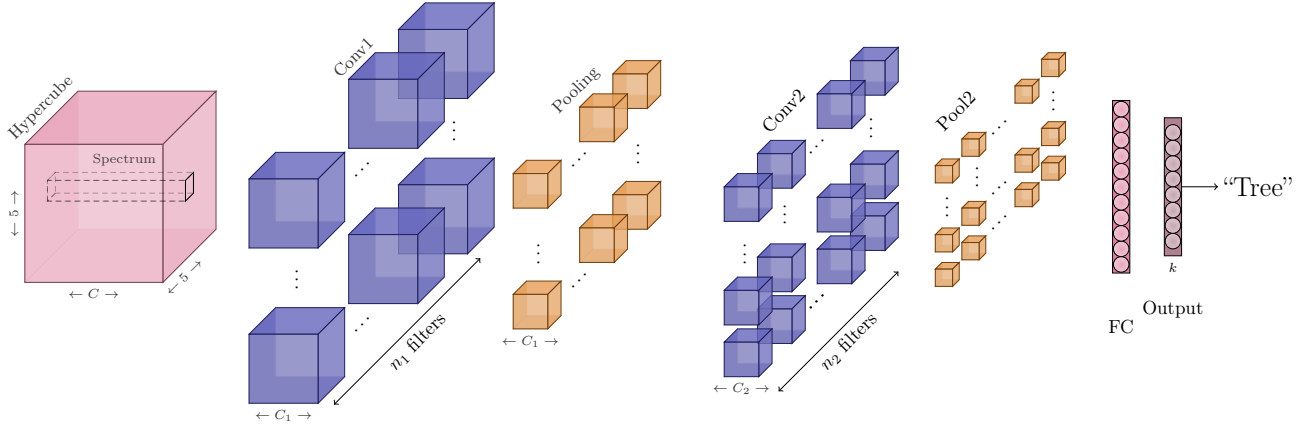
Fig. 7: 3D CNN for classification of hyperspectral data available in the toolbox. It reproduces the architecture proposed in [71] and alternates 3D convolutions and 3D max-pooling layers.
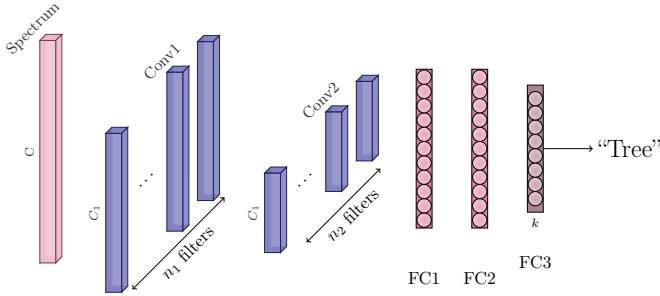
—

Fig. 8: 1D CNN for spectral classification of hyperspectral data available in the toolbox.
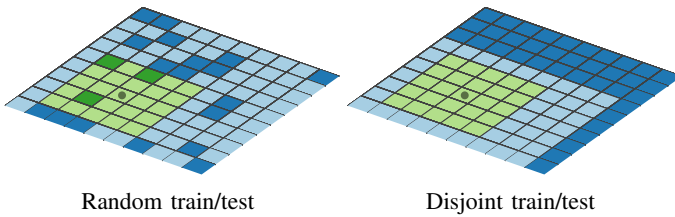

Fig. 9: Train , Test , Receptive field , Test pixel in receptive field , ●: center pixel. The 2D receptive field of a CNN can involuntarily include samples from the test set, making the network overfit and biasing the evaluation.

of each model and the gap between evaluating either on a uniform random or a disjoint separated test set. Except for the RNN from [43] for which our reimplementation underperforms, all of our reimplementations closely approach the results reported in the original papers. However, experimenting on a disjoint test set, which is the realistic use case, shows that the accuracies are actually much lower than expected, as reported in Table II. It is interesting to see that Indian Pines exhibit a very different behavior compared to Pavia University and DFC2018. Indeed, including spatial information using

2D+1D and 3D CNN in Indian Pines actually decreases our classification accuracies. Our hypothesis stems from the fact that Indian Pines has a very low spatial resolution (20 m/px) compared to the two other datasets. This leads to each pixel being actually a mixture of the ground materials. As this dataset is focused on crop classification, this means that each pixel will be an average of the crop reflectance over 400m^2, and bringing information from neighboring pixels does not really improve the discriminative power of the model. On the higher resolution Pavia University and DFC2018, 3D spatial-spectral CNN significantly boosts the model's accuracy, respectively by 3% and 2% compared to the 1D CNN, which shows the capacity of 3D CNN to efficiently combine spatial and spectral patterns with 3D filters. The DFC2018 is a very challenging dataset because of its large number of classes with high inter-class similarity. In our experiments, the 1D NN suffered from grave overfitting and performs worse than a linear SVM trained by SGD. This is because the test set from the DFC2018 has a large spatial extent and pixels have very different statistical properties compared to the train set. This overfitting is not a problem in Indian Pines and Pavia University, where the train and test sets are very similar and on the same scene, but will become much more important on real large scenes such as Houston from the DFC2018. Interestingly, the 2D+1D CNN seems to fail to catch the spatial information, maybe because of the low dimensions of the spatial kernels. Higher resolution datasets such as the DFC2018 would probably benefit from larger receptive fields to model long-distance spatial relationships between pixels.

E. Perspectives

Several potential research axes appear for the future. A first one, which follows the works described previously, is the implementation of fully-3D, end-to-end networks for hyperspectral data. Indeed, they are theoretically able to perform the reduction dimension in both the spatial domain and the spectral domain. However, due to the small spatial resolution, a direct transfer of 2D CNN which mainly rely on the spatial context

Method	Indian Pines		Pavia University		Data Fusion Contest 2018	
	Accuracy	κ	Accuracy	κ	Accuracy	κ
SVM	81.43	0.788	69.56	0.592	42.51	0.39
1D NN	**83.13**±0.84	**0.807**±0.009	76.90±0.86	0.711±0.010	41.08	0.37
1D CNN [42]	*82.99*±0.93	*0.806*±0.011	81.18±1.96	0.759±0.023	*47.01*	*0.44*
RNN [43]	79.70±0.91	0.769±0.011	67.71±1.25	0.599±0.014	41.53	0.38
2D+1D CNN [63]	74.31±0.73	0.707±0.008	*83.80*±1.29	*0.792*±0.016	46.28	0.43
3D CNN [68]	75.47±0.85	0.719±0.010	**84.32**±0.72	**0.799**±0.009	**49.26**	**0.46**

TABLE II: Experimental results of various models from our toolbox on the Indian Pines, Pavia University and DFC 2018 datasets using the DASE train/test split. Best results are in **bold** and second are in *italics*.

Dataset	Indian Pines		Pavia University	
Model	Random	Disjoint	Random	Disjoint
Nearest-neighbor	75.63	67.27	89.99	57.77
1D CNN [42] (original)	90.16	-	92.56	-
1D CNN (ours)	89.34	82.99	90.59	81.18
RNN [43] (original)	85.7	-	-	80.70
RNN (ours)	79.70	62.23	-	67.71
2D+1D CNN [63] (original)	-	-	94.6	-
2D+1D CNN (ours)	-	-	92.39	83.80
3D CNN [68] (original)	99.07	-	99.39	-
3D CNN (ours)	96.87	75.47	96.71	84.32

TABLE III: Experimental results with respect to methodological discrepancies between various implementations and evaluation strategies on the Indian Pines and Pavia University datasets.

of a full spectrum is not the most promising alternative. On the contrary, the transfer of 3D fully-convolutional networks could extract meaningful local features and tackle this issue.

However, there is also a need for new, larger and more complex (e.g., with more classes) reference hyperspectral datasets for the classification task. Indeed, the community has developed standard approaches which already reach satisfying results on most of them, so it is not anymore possible to distinguish between the new proposed approaches.

A second promising research field is the study of unsupervised and semi-supervised models to overcome the sparsely available annotations of hyperspectral data. For example, 3D spatial-spectral auto-encoders might learn representations of hyperspectral scenes independently of the acquisition context, which can be used for subsequent tasks such as clustering or classification.

Finally, a last coming research field is data synthesis, which would allow to simulate realistic hyperspectral scenes. In the recent years, generative models such as Gaussian Mixture Models or Generative Adversarial Networks (GAN) have been applied to generate new training samples. Initially, the idea was that generative models could approximate the distribution of the deep features in the embedded space. Then, the models infers new samples that could be used to train larger classifiers. [79] (using GMM) and [80] (using GAN) are two examples of such works. However, thanks to the theoretical improvements to the GAN framework, deep generative models are now able to synthesize from scratch hyperspectral pixels [81] and even small hyperspectral cubes using 3D CNN [82]. This can of course be used for data augmentation, although the benefit of adding new data estimated from a generative model does not seem huge at this point. Nonetheless, generative models will likely be a great asset to estimate image transformations that usually are costly to implement or require specific expert knowledge, such as atmospheric correction, transfer function estimation between sensors or image denoising.

V. CONCLUSION

Deep learning already proved to be efficient for hyperspectral data. 2D or 3D convolutional networks allow to combine spatial and spectral information intuitively and the first published works show state of the art performances, often without expert knowledge about the physics or the sensor.

Today, the main challenge is the scarce availability of massively annotated datasets. Yet, data volume is the key of the success of statistical approaches. A promising path is the development of unsupervised approaches and data augmentation by synthesis in order to overcome this issue and unlock the full potential of deep learning in hyperspectral imaging.

REFERENCES

[1] N. Audebert, B. Le Saux, and S. Lefèvre, "Semantic Segmentation of Earth Observation Data Using Multimodal and Multi-scale Deep Networks," in *Computer Vision – ACCV 2016*. Springer, Cham, Nov. 2016, pp. 180–196.

[2] M. Volpi and D. Tuia, "Dense Semantic Labeling of Subdecimeter Resolution Images With Convolutional Neural Networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 2, pp. 881–893, 2017.

[3] D. Marmanis, J. D. Wegner, S. Galliani, K. Schindler, M. Datcu, and U. Stilla, "Semantic Segmentation of Aerial Images with an Ensemble of CNNs," *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 3, pp. 473–480, 2016.

[4] J. Bioucas-Dias, A. Plaza, G. Camps-Valls, N. Nasrabadi, P. Scheunders, and J. Chanussot, "Hyperspectral remote sensing data analysis and future challenges," *IEEE Geoscience and Remote Sensing Magazine*, vol. 1, no. 2, pp. 6–36, June 2013.

[5] M. Cubero-Castan, J. Chanussot, V. Achard, X. Briottet, and M. Shimoni, "A Physics-Based Unmixing Method to Estimate Subpixel Temperatures on Mixed Pixels," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 4, pp. 1894–1906, Apr. 2015.

[6] X. X. Zhu, D. Tuia, L. Mou, G. Xia, L. Zhang, F. Xu, and F. Fraundorfer, "Deep learning in remote sensing: A comprehensive review and list of resources," *IEEE Geoscience and Remote Sensing Magazine*, vol. 5, no. 4, pp. 8–36, Dec 2017.

[7] P. Ghamisi, N. Yokoya, J. Li, W. Liao, S. Liu, J. Plaza, B. Rasti, and A. Plaza, "Advances in hyperspectral image and signal processing: A comprehensive overview of the state of the art," *IEEE Geoscience and Remote Sensing Magazine*, vol. 5, no. 4, pp. 37–78, Dec 2017.

[8] P. Y. Deschamps and T. Phulpin, "Atmospheric correction of infrared measurements of sea surface temperature using channels at 3.7, 11 and 12 mm," *Boundary-Layer Meteorology*, vol. 18, no. 2, pp. 131–143, Mar. 1980.

[9] H. Rahman and G. Dedieu, "SMAC: a simplified method for the atmospheric correction of satellite measurements in the solar spectrum," *International Journal of Remote Sensing*, vol. 15, no. 1, pp. 123–143, Jan. 1994.

[10] P. Chavez, "Image-based atmospheric corrections: Revisited and improved," *Photogrammetric engineering and remote sensing*, vol. 62, no. 9, pp. 1025–1036, 1996.

[11] B.-C. Gao, M. J. Montes, C. O. Davis, and A. F. Goetz, "Atmospheric correction algorithms for hyperspectral remote sensing data of land and ocean," *Remote Sensing of Environment*, vol. 113, pp. S17–S24, 2009.

[12] IEEE GRSS. (2018) 2018 IEEE GRSS Data Fusion Contest. [Online]. Available: http://www.grss-ieee.org/community/technical-committees/data-fusion

[13] B. Le Saux, N. Yokoya, R. Haensch, and S. Prasad, "2018 IEEE GRSS Data Fusion Contest: Multimodal land use classification [technical committees]," *IEEE Geoscience and Remote Sensing Magazine*, vol. 6, no. 1, pp. 52–54, March 2018.

[14] M. D. Farrell and R. M. Mersereau, "On the impact of PCA dimension reduction for hyperspectral detection of difficult targets," *IEEE Geoscience and Remote Sensing Letters*, vol. 2, no. 2, pp. 192–195, Apr. 2005.

[15] B. Guo, S. R. Gunn, R. I. Damper, and J. D. B. Nelson, "Band Selection for Hyperspectral Image Classification Using Mutual Information," *IEEE Geoscience and Remote Sensing Letters*, vol. 3, no. 4, pp. 522–526, Oct. 2006.

[16] Q. Du, "Band selection and its impact on target detection and classification in hyperspectral image analysis," in *IEEE Workshop on Advances in Techniques for Analysis of Remotely Sensed Data, 2003*, Oct 2003, pp. 374–377.

[17] R. Yuhas, A. Goetz, and J. Boardman, "Discrimination among semi-arid landscape endmembers using the Spectral Angle Mapper (SAM) algorithm," in *Summaries of the Third Annual JPL Airborne Geoscience Workshop. Volume 1: AVIRIS Workshop*, Jun. 1992. [Online]. Available: https://ntrs.nasa.gov/search.jsp?R=19940012238

[18] L. Parra, C. Spence, P. Sajda, A. Ziehe, and K.-R. Müller, "Unmixing hyperspectral data," in *Proceedings of the 12th International Conference on Neural Information Processing Systems*. MIT Press, 1999, pp. 942–948.

[19] A. Le Bris, N. Chehata, X. Briottet, and N. Paparoditis, "Extraction of optimal spectral bands using hierarchical band merging out of hyperspectral data," *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XL-3/W3, pp. 459–465, Aug. 2015.

[20] C. Rodarmel and J. Shan, "Principal component analysis for hyperspectral image classification," *Surveying and Land Information Science*, vol. 62, no. 2, p. 115, 2002.

[21] B. B. Damodaran, N. Courty, and S. Lefèvre, "Sparse Hilbert Schmidt Independence Criterion and Surrogate-Kernel-Based Feature Selection for Hyperspectral Image Classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 4, pp. 2385–2398, 2017.

[22] J. Ham, Y. Chen, M. M. Crawford, and J. Ghosh, "Investigation of the random forest framework for classification of hyperspectral data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 3, pp. 492–501, March 2005.

[23] F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, no. 8, pp. 1778–1790, Aug 2004.

[24] J. A. Gualtieri and R. F. Cromp, "Support vector machines for hyperspectral remote sensing classification," in *Proc. SPIE 27th AIPR Workshop: Advances in Computer-Assisted Recognition*, vol. 3584, 1999, pp. 1 – 12.

[25] L. Chapel, T. Burger, N. Courty, and S. Lefèvre, "PerTurbo Manifold Learning Algorithm for Weakly Labeled Hyperspectral Image Classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 4, pp. 1070–1078, Apr. 2014.

[26] J. Wu, Z. Jiang, H. Zhang, B. Cai, and Q. Wei, "Semi-supervised Conditional Random Field for hyperspectral remote sensing image classification," in *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, Jul. 2016, pp. 2614–2617.

[27] Y. Tarabalka, J. Chanussot, and J. A. Benediktsson, "Segmentation and classification of hyperspectral images using watershed transformation," *Pattern Recognition*, vol. 43, no. 7, pp. 2367–2379, 2010.

[28] M. Fauvel, Y. Tarabalka, J. A. Benediktsson, J. Chanussot, and J. C. Tilton, "Advances in Spectral-Spatial Classification of Hyperspectral Images," *Proceedings of the IEEE*, vol. 101, no. 3, pp. 652–675, Mar. 2013.

[29] E. Aptoula, M. D. Mura, and S. Lefèvre, "Vector Attribute Profiles for Hyperspectral Image Classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 6, pp. 3208–3220, Jun. 2016.

[30] M. D. Mura, J. A. Benediktsson, B. Waske, and L. Bruzzone, "Extended profiles with morphological attribute filters for the analysis of hyperspectral data," *International Journal of Remote Sensing*, vol. 31, no. 22, pp. 5975–5991, 2010. [Online]. Available: https://doi.org/10.1080/01431161.2010.512425

[31] A. Plaza, P. Martinez, R. Perez, and J. Plaza, "Spatial/spectral endmember extraction by multidimensional morphological operations," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 40, no. 9, pp. 2025–2041, Sep. 2002.

[32] F. Dell'Acqua, P. Gamba, A. Ferrari, J. A. Palmason, J. A. Benediktsson, and K. Arnason, "Exploiting spectral and spatial information in hyperspectral urban data with high resolution," *IEEE Geoscience and Remote Sensing Letters*, vol. 1, no. 4, pp. 322–326, Oct. 2004.

[33] G. Camps-Valls, L. Gomez-Chova, J. Muñoz-Marí, J. Vila-Francés, and J. Calpe-Maravilla, "Composite kernels for hyperspectral image classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 3, no. 1, pp. 93–97, 2006.

[34] Y. Tarabalka, J. A. Benediktsson, and J. Chanussot, "Spectral–spatial classification of hyperspectral imagery based on partitional clustering techniques," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 8, pp. 2973–2987, 2009.

[35] M. Fauvel, J. Chanussot, and J. A. Benediktsson, "A Spatial-spectral Kernel-based Approach for the Classification of Remote-sensing Images," *Pattern Recogn.*, vol. 45, no. 1, pp. 381–392, Jan. 2012.

[36] Y. Cui, L. Chapel, and S. Lefèvre, "Scalable Bag of Subpaths Kernel for Learning on Hierarchical Image Representations and Multi-Source Remote Sensing Data Classification," *Remote Sensing*, vol. 9, no. 3, p. 196, Feb. 2017.

[37] D. Tuia, R. Flamary, and N. Courty, "Multiclass feature learning for hyperspectral image classification: Sparse and hierarchical solutions," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 105, pp. 272–285, 2015.

[38] R. Guo, W. Wang, and H. Qi, "Hyperspectral image unmixing using autoencoder cascade," in *7th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, 2015, pp. 1–4.

[39] S. Ozkan, B. Kaya, E. Esen, and G. B. Akar, "Endnet: Sparse autoencoder network for endmember extraction and hyperspectral unmixing," *arXiv:1708.01894*, 2017.

[40] P. K. Goel, S. O. Prasher, R. M. Patel, J. A. Landry, R. B. Bonnell, and A. A. Viau, "Classification of hyperspectral data by decision trees and artificial neural networks to identify weed stress and nitrogen status of corn," *Computers and Electronics in Agriculture*, vol. 39, no. 2, pp. 67–93, 2003.

[41] F. Ratle, G. Camps-Valls, and J. Weston, "Semisupervised Neural Networks for Efficient Hyperspectral Image Classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 48, no. 5, pp. 2271–2282, 2010.

[42] W. Hu, Y. Huang, L. Wei, F. Zhang, and H. Li, "Deep Convolutional Neural Networks for Hyperspectral Image Classification," *Journal of Sensors*, vol. 2015, p. e258619, Jul. 2015.

[43] L. Mou, P. Ghamisi, and X. X. Zhu, "Deep Recurrent Neural Networks for Hyperspectral Image Classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 7, pp. 3639–3655, Jul. 2017.

[44] H. Wu and S. Prasad, "Semi-Supervised Deep Learning Using Pseudo Labels for Hyperspectral Image Classification," *IEEE Transactions on Image Processing*, vol. 27, no. 3, pp. 1259–1270, Mar. 2018.

[45] ——, "Convolutional Recurrent Neural Networks forHyperspectral Data Classification," *Remote Sensing*, vol. 9, no. 3, p. 298, Mar. 2017.

[46] C. Xing, L. Ma, and X. Yang, "Stacked Denoise Autoencoder Based Feature Extraction and Classification for Hyperspectral Images," *Journal of Sensors*, vol. 2016, p. e3632943, Nov. 2015.

[47] Q. Fu, X. Yu, X. Wei, and Z. Xue, "Semi-supervised classification of hyperspectral imagery based on stacked autoencoders," in *Proc. SPIE 10033, Eighth International Conference on Digital Image Processing (ICDIP)*, vol. 10033, 2016, pp. 100 332B–100 332B–5.

[48] K. Makantasis, K. Karantzalos, A. Doulamis, and N. Doulamis, "Deep supervised learning for hyperspectral data classification through convolutional neural networks," in *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, Jul. 2015, pp. 4959–4962.

[49] H. Lee and H. Kwon, "Going deeper with contextual cnn for hyperspectral image classification," *IEEE Transactions on Image Processing*, vol. 26, no. 10, pp. 4843–4855, Oct 2017.

[50] V. Slavkovikj, S. Verstockt, W. De Neve, S. Van Hoecke, and R. Van de Walle, "Hyperspectral Image Classification with Convolutional Neural Networks," in *Proceedings of the 23rd ACM International Conference on Multimedia*, ser. MM '15. New York, NY, USA: ACM, 2015, pp. 1159–1162.

[51] W. Zhao and S. Du, "Spectral-Spatial Feature Extraction for Hyperspectral Image Classification: A Dimension Reduction and Deep Learning Approach," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 8, pp. 4544–4554, 2016.

[52] J. Yue, W. Zhao, S. Mao, and H. Liu, "Spectral-spatial classification of hyperspectral images using deep convolutional neural networks," *Remote Sensing Letters*, vol. 6, no. 6, pp. 468–477, 2015.

[53] 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," *International Journal of Remote Sensing*, vol. 36, no. 13, pp. 3368–3379, 2015.

[54] A. Romero, C. Gatta, and G. Camps-Valls, "Unsupervised Deep Feature Extraction for Remote Sensing Image Classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 3, pp. 1349–1362, Mar. 2016.

[55] T. Li, J. Zhang, and Y. Zhang, "Classification of hyperspectral image based on deep belief networks," in *2014 IEEE International Conference on Image Processing (ICIP)*, Oct. 2014, pp. 5132–5136.

[56] Y. Chen, X. Zhao, and X. Jia, "Spectral-Spatial Classification of Hyperspectral Data Based on Deep Belief Network," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 8, no. 6, pp. 2381–2392, Jun. 2015.

[57] Z. Lin, Y. Chen, X. Zhao, and G. Wang, "Spectral-spatial classification of hyperspectral image using autoencoders," in *2013 9th International Conference on Information, Communications Signal Processing*, 2013, pp. 1–5.

[58] M. E. Midhun, S. R. Nair, V. T. N. Prabhakar, and S. S. Kumar, "Deep Model for Classification of Hyperspectral Image Using Restricted Boltzmann Machine," in *Proceedings of the 2014 International Conference on Interdisciplinary Advances in Applied Computing*, ser. ICONIAAC '14. New York, NY, USA: ACM, 2014, pp. 35:1–35:7.

[59] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep Learning-Based Classification of Hyperspectral Data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 6, pp. 2094–2107, Jun. 2014.

[60] X. Ma, H. Wang, and J. Geng, "Spectral-Spatial Classification of Hyperspectral Image Based on Deep Auto-Encoder," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 9, no. 9, pp. 4073–4085, Sep. 2016.

[61] C. Tao, H. Pan, Y. Li, and Z. Zou, "Unsupervised Spectral-Spatial Feature Learning With Stacked Sparse Autoencoder for Hyperspectral Imagery Classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 12, pp. 2438–2442, 2015.

[62] 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 Computing*, vol. 21, no. 1, pp. 213–221, Jan. 2017.

[63] A. Ben Hamida, A. Benoit, P. Lambert, and A. Chokri Ben, "Deep Learning Approach for Remote Sensing Image Analysis," in *Big Data from Space (BiDS'16)*, Santa Cruz de Tenerife, Spain, Mar. 2016, p. 133.

[64] Y. Luo, J. Zou, C. Yao, T. Li, and G. Bai, "HSI-CNN: A Novel Convolution Neural Network for Hyperspectral Image," *arXiv:1802.10478 [cs]*, Feb. 2018.

[65] L. Mou, P. Ghamisi, and X. X. Zhu, "Unsupervised Spectral–Spatial Feature Learning via Deep Residual Conv–Deconv Network for Hyperspectral Image Classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 1, pp. 391–406, Jan. 2018. [Online]. Available: http://ieeexplore.ieee.org/document/8082108/

[66] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, United States, Jun. 2016, pp. 770–778.

[67] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, pp. 1–9. [Online]. Available: http://www.cv-foundation.org/openaccess/content_cvpr_2015/html/Szegedy_Going_Deeper_With_2015_CVPR_paper.html

[68] Y. Li, H. Zhang, and Q. Shen, "Spectral–Spatial Classification of Hyperspectral Imagery with 3D Convolutional Neural Network," *Remote Sensing*, vol. 9, no. 1, p. 67, Jan. 2017.

[69] M. He, B. Li, and H. Chen, "Multi-scale 3d deep convolutional neural network for hyperspectral image classification," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, Sept 2017, pp. 3904–3908.

[70] B. Liu, X. Yu, P. Zhang, X. Tan, A. Yu, and Z. Xue, "A semi-supervised convolutional neural network for hyperspectral image classification," *Remote Sensing Letters*, vol. 8, pp. 839–848, 2017.

[71] 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 Transactions on Geoscience and Remote Sensing*, vol. 54, no. 10, pp. 6232–6251, Oct. 2016.

[72] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation Applied to Handwritten Zip Code Recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, Dec. 1989.

[73] D. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015, arXiv: 1412.6980. [Online]. Available: http://arxiv.org/abs/1412.6980

[74] K. He, X. Zhang, S. Ren, and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1026–1034.

[75] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[76] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in *Proceedings of the 32nd International Conference on Machine Learning*, 2015, pp. 448–456. [Online]. Available: http://jmlr.org/proceedings/papers/v37/ioffe15.html

[77] R. Kemker, R. Luu, and C. Kanan, "Low-Shot Learning for the Semantic Segmentation of Remote Sensing Imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 10, pp. 6214–6223, Oct. 2018.

[78] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[79] A. Davari, E. Aptoula, B. Yanikoglu, A. Maier, and C. Riess, "GMM-Based Synthetic Samples for Classification of Hyperspectral Images With Limited Training Data," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 6, pp. 942–946, Jun. 2018.

[80] Z. He, H. Liu, Y. Wang, J. Hu, Z. He, H. Liu, Y. Wang, and J. Hu, "Generative Adversarial Networks-Based Semi-Supervised Learning for Hyperspectral Image Classification," *Remote Sensing*, vol. 9, no. 10, p. 1042, Oct. 2017. [Online]. Available: https://www.mdpi.com/2072-4292/9/10/1042

[81] N. Audebert, B. Le Saux, and S. Lefèvre, "Generative adversarial networks for realistic synthesis of hyperspectral samples," in *2018 IEEE*

International Geoscience and Remote Sensing Symposium (IGARSS), Jul. 2018, pp. 5091–5094.

[82] L. Zhu, Y. Chen, P. Ghamisi, and J. A. Benediktsson, "Generative Adversarial Networks for Hyperspectral Image Classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 9, pp. 5046–5063, Sep. 2018.

APPENDIX

ABOUT THE AUTHORS

Dr. Nicolas Audebert (M.Eng 2015 Supélec, M.Sc 2015 Université Paris-Sud, PhD 2018 Univ. Bretagne-Sud) is a research scientist working on computer vision and deep learning for remote sensing. His PhD in Computer Science, prepared at ONERA, the French Aerospace Laboratory and the IRISA research institute, focused on the use of deep neural networks for scene understanding of airborne and satellite optical images and Earth Observation. He was awarded the ISPRS Best Benchmarking Contribution at GEOBIA'16 and 2nd Best Student Paper at JURSE'17 prizes.

Dr. Bertrand Le Saux (M.Eng 1999, M.Sc 1999 both at INP Grenoble, PhD 2003 Univ. Versailles / Inria Rocquencourt) is a research scientist in the Information Processing and Systems Department at ONERA, the French Aerospace Laboratory. His research objective is visual understanding by means of data-driven techniques. Currently, his interests focus on developing machine learning and deep learning methods for remote sensing, (flying) robotics and 3D vision. He currently serves as the chair of the IEEE GRSS Image Analysis and Data Fusion Technical Committee.

Prof. Sébastien Lefèvre (M.Sc. 1999 at TU Compiègne, Ph.D. 2002 at University of Tours, French HDR degree 2009 at University of Strasbourg) is Full Professor in Computer Science in University Bretagne Sud since 2010, where he conducts his researches within the Institute for Research in Computer Science and Random Systems (IRISA). He is leading the OBELIX team dedicated to image analysis and machine learning for remote sensing and Earth Observation (http://www.irisa.fr/obelix). He has coauthored more than 140 papers in image analysis and pattern recognition. His current research interests are mainly related to hierarchical image analysis and deep learning applied to remote sensing of environment. He was co-chair of GEOBIA 2016 and is co-chair of JURSE 2019.