ISPRS Journal of Photogrammetry and Remote Sensing 158 (2019) 279–317

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

journal homepage: www.elsevier.com/locate/isprsjprs

Deep learning classifiers for hyperspectral imaging: A review

M.E. Paoletti*, J.M. Haut, J. Plaza, A. Plaza

Hyperspectral Computing Laboratory (HyperComp), Department of Computer Technology and Communications, Escuela Politecnica de Caceres, University of Extremadura, Avenida de la Universidad s/n, E-10003 Caceres, Spain

ARTICLE INFO

Keywords:
Deep learning (DL)
Hyperspectral imaging (HSI)
Earth observation (EO)
Classification

ABSTRACT

Advances in computing technology have fostered the development of new and powerful deep learning (DL) techniques, which have demonstrated promising results in a wide range of applications. Particularly, DL methods have been successfully used to classify remotely sensed data collected by Earth Observation (EO) instruments. Hyperspectral imaging (HSI) is a hot topic in remote sensing data analysis due to the vast amount of information comprised by this kind of images, which allows for a better characterization and exploitation of the Earth surface by combining rich spectral and spatial information. However, HSI poses major challenges for supervised classification methods due to the high dimensionality of the data and the limited availability of training samples. These issues, together with the high intraclass variability (and interclass similarity) –often present in HSI data– may hamper the effectiveness of classifiers. In order to solve these limitations, several DL-based architectures have been recently developed, exhibiting great potential in HSI data interpretation. This paper provides a comprehensive review of the current-state-of-the-art in DL for HSI classification, analyzing the strengths and weaknesses of the most widely used classifiers in the literature. For each discussed method, we provide quantitative results using several well-known and widely used HSI scenes, thus providing an exhaustive comparison of the discussed techniques. The paper concludes with some remarks and hints about future challenges in the application of DL techniques to HSI classification. The source codes of the methods discussed in this paper are available from: https://github.com/mhaut/hyperspectral_deeplearning_review.

1. Introduction

Imaging spectroscopy, also called hyperspectral imaging (HSI), studies how the light interacts with the observed materials, measuring the amount of light that is emitted, reflected or transmitted from a certain object or target. Imaging spectrometers (also called HSI sensors) usually operate in the 0.4 to 2.5 μm spectral region, capturing the visible and solar-reflected infrared spectrum (i.e., the near-infrared or NIR, and the short-wavelength infrared or SWIR) from the observed materials. However, as opposed to broad-band sensing systems that under-sample the available spectral information, narrow-band HSI systems are able to produce, for each captured target, a distinctive spectral signature composed by reflectance measurements at hundreds of different wavelength channels (Goetz et al., 1985). The exploitation of spectral signatures as unique *fingerprints* makes imaging spectrometry an interesting and powerful tool for the categorization of the surface of the Earth, reaching promising results in a wide range of applications (Huadong et al., 2001; Transon et al., 2017; Khan et al., 2018; Transon et al., 2018). In the current literature, a great number of works focus on the use of HSI data for resource management. For instance, in agricultural applications (Teke et al., 2013) there are several works focused on the analysis of environmental stress in crops and associated diseases (Strachan et al., 2002; Feng et al., 2017), crops variability (Yang et al., 2004; Rußwurm and Körner, 2017), soil erosion stages (Bannari et al., 2006; Chabrillat et al., 2014) or precision agriculture (Haboudane et al., 2004; Rodríguez-Pérez et al., 2007; Mahesh et al., 2015), among many others. In forestry and environmental management, relevant works have been presented on analyzing the status and health of forests (Coops et al., 2003; Shang and Chisholm, 2014), invasive species detection (Ustin et al., 2002a; Große-Stoltenberg et al., 2016), and infestations in plantation forestry (Narumalani et al., 2009; Peerbhay et al., 2015). Also, in water and maritime resources management (Younos and Parece, 2015), several studies have focused on water quality analysis (Koponen et al., 2002; Olmanson et al., 2013; El-Magd and El-Zeiny, 2014) and precipitations (Zhou et al., 2011) or sea ice detection (Han et al., 2017). In geological exploration and mineralogy, HSI data have been used for detection and mapping of mineral deposits (Resmini et al., 1997; Kokaly et al., 2013; Kokaly et al., 2016; Mazhari et al., 2017; Scafutto et al., 2017; Aslett et al., 2018; Dumke et al., 2018; Acosta et al., 2019) or soil composition analysis (Shi et al.,

* Corresponding author.
E-mail addresses: mpaoletti@unex.es (M.E. Paoletti), juanmariohaut@unex.es (J.M. Haut), jplaza@unex.es (J. Plaza), aplaza@unex.es (A. Plaza).

https://doi.org/10.1016/j.isprsjprs.2019.09.006
Received 6 November 2018; Received in revised form 9 September 2019; Accepted 9 September 2019
Available online 19 November 2019

2014). Other areas in which the use of HSI provided relevant results include urban planning (Abbate et al., 2003; Lulla et al., 2009; Heldens et al., 2011; Man et al., 2015; Anand et al., 2017), disaster prediction (Ustin et al., 2002b; Roberts et al., 2003; Transon et al., 2018; Veraverbeke et al., 2018), military and defense applications (Richter, 2005; Briottet et al., 2006; Ardouin et al., 2007; El-Sharkawy and Elbasuney, 2019) and archaeological analyses (Savage et al., 2012).

Several efforts have been made over the past decades to produce high-quality HSI data for Earth Observation (EO) (Lucas et al., 2004; Ghamisi et al., 2017b), developing a wide range of imaging spectrometers placed on aerial/satellite platforms, and recently also on stationary or hand-held platforms. These sensors combine the power of digital imaging and spectroscopy to extract, for every location in an image plane, the corresponding spectral signature, using thousands of narrow and continuous bands and acquiring complete HSI data cubes by raster-scanning the scene while the platform moves across the surface (i.e. pushbroom sensors), covering large observation areas. As result, the captured area or scene is recorded in different wavebands, creating a huge data cube $\mathbf{X} \in \mathbb{N}^{n_1 \times n_2 \times n_{bands}}$, composed by $(n_1 \times n_2)$ spectral vectors or HSI pixels, where each $\mathbf{x}_i \in \mathbb{N}^{n_{bands}}$ records the spectral signature of the observed material.

Nowadays, several instruments are routinely capturing great volumes of HSI data, with some of them exhibiting high acquisition rates, i.e. being able to capture gigabytes (GBs) or even terabytes (TBs) of data per hour (Vane et al., 1989; Kruse et al., 2000). In this regard, Table 1 provides the specifications of some of the best-known spectrometers currently available. Moreover, advances in computing technologies have achieved great improvements in the data acquisition, storage and processing procedures, allowing also the launch of a number of HSI-EO missions –such as the NASA Hyperspectral Infrared Imager (HyspIRI) (Roberts et al., 2012), the Environmental Mapping and Analysis Program (EnMAP) (Kaufmann et al., 2008) or the Precursore IperSpettrale della Missione Applicativa (PRISMA) program (Galeazzi et al., 2008)– as well as the practical application of remotely sensed HSI data in real scenarios (Tuia and Camps-Valls, 2009; Zhang and Du, 2012), providing a general idea about the importance and utility of HSI-based remote sensing.

The specialized literature about remotely sensed HSI data covers a wide range of processing techniques that can efficiently extract the information contained in the HSI cube. The most popular ones include: (i) spectral unmixing (Bioucas-Dias et al., 2012; Heylen et al., 2014; Shi and Wang, 2014; Sánchez et al., 2015; Zhong et al., 2016a), (ii) resolution enhancement (Eismann and Hardie, 2005; Mookambiga and Gomathi, 2016; Yi et al., 2017; Yi et al., 2018), (iii) image restoration and denoising (Xu and Gong, 2008; Chen and Qian, 2011; Zhang et al., 2014; Wei et al., 2017b), (iv) anomaly detection (Stein et al., 2002; Xu et al., 2016; Kang et al., 2017), (v) dimensionality reduction (Bruce et al., 2002; Haut et al., 2018d) and (vi) data classification (Fauvel et al., 2013; Camps-Valls et al., 2014; Ghamisi et al., 2017a). In this work, we particularly focus on the topic of HSI data classification, which has received remarkable attention due its important role in land use and land cover applications (Cheng et al., 2017a), and which is currently one of the most popular techniques for HSI data exploitation (Chang, 2007).

A wide variety of HSI data classification methodologies rely on machine learning (ML) techniques (Kotsiantis et al., 2006; Kotsiantis et al., 2007), which are already collected in an extensive list of detailed reviews, such as Plaza et al. (2009), Zhang and Du (2012), Ablin and Sulochana (2013), Fauvel et al. (2013), Camps-Valls et al. (2014), Li and Du (2016), Chutia et al. (2016), Ghamisi et al. (2017b), Chen et al. (2014b), or even more recently in Li et al. (2019a), Audebert et al. (2019), Signoroni et al. (2019), among others. However, ML is a field in constant evolution, where new and improved methods are designed from time to time. In this sense, from the early 2000s, the ML field has experimented a significant revolution thanks to the development of new deep learning (DL) models (Schmidhuber, 2015), which have been supported by advances in computer technology. These models have become an inspiration for the development of new and improved HSI data classifiers, marking a clear trend since 2017 (Petersson et al., 2016; Ghamisi et al., 2017a; Zhu et al., 2017). In this sense, the aim of this work is to delve deeper into those classification techniques based on DL techniques, providing an updated review about the most popular models and widely used architectures to perform remotely sensed HSI data classification.

The remainder of the paper is organized as follows. In Section 2, we introduce the problem of HSI data classification, providing a brief framework for ML and DL methods, introducing the general benefits of DL models and their limitations, coupled with the challenges that must be faced when working with remotely sensed HSI data. Section 3 introduces some general DL concepts, while Section 4 reviews the principal DNN architectures employed for HSI data classification. Section 5 introduces some widely-used techniques to overcome DL and HSI limitations. Section 6 presents some popular programming frameworks for the development of DL models. Section 7 provides an experimental evaluation of the discussed methods using several well-known HSI data sets. Our experimental assessment includes a detailed discussion of the results obtained in terms of accuracy and performance. Section 8 concludes the paper with a discussion on future trends, including ongoing computational developments such as the use of parallelization and distribution techniques via graphical processing units (GPUs) and cloud computing environments.

Table 1
Some of the most widely-known HSI sensors, highlighting several of their spectral-spatial characteristics. In particular, we outline the spectral features, the number of bands, range (μm), and spectral resolution (nm), taking into account also the spatial ground sample distance measured in meters per pixel (mpp).

	Sensor	Bands	Range	Width	GSD
Airborne	AVIRIS (Green et al., 1998)	224	0.36–2.45	10	20
	AVIRIS-NG (Bue et al., 2015)	600	0.38–2.51	5	0.3–4.0
	CASI (Babey and Anger, 1989)	144	0.36–1.05	2.4	2.5
	HYDICE (Rickard et al., 1993)	210	0.40–2.50	10.2	1–7
	HYMAP (Cocks et al., 1998)	126	0.45–2.50	15	5
	PRISM (Mouroulis et al., 2014)	248	0.35–1.05	3.5	2.5
	ROSIS (Kunkel et al., 1988)	115	0.43–0.86	4	1.3
Satellite	EnMAP (Guanter et al., 2015)	228	0.42–2.40	5.25–12.5	30
	DESIS (Eckardt et al., 2015)	180	0.40–1.00	3.30	30
	HYPERION (Pearlman et al., 2003)	220	0.40–2.50	10	30
	PRISMA (Pignatti et al., 2013)	237	0.40–2.50	≤12	30
	SHALOM (Feingersh and Dor, 2015)	241	0.40–2.50	10	10

2. Hyperspectral data classification: backgrounds and challenges

2.1. From traditional machine learning methods to deep learning models

Any classification problem can be mathematically formulated as an optimization one, where a mapping function (with or without certain parameters θ) $f_c(\cdot,\theta)$ receives an input data sample $\mathcal{X}$ and obtains the corresponding label category, $\mathcal{Y}$, by applying several transformations over the original input, i.e. $f_c: \mathcal{X} \rightarrow \mathcal{Y}$, with the aim of minimizing the gap between the desired output and the obtained one. In this regard, the purpose of classifying HSI data is to categorize those pixels $\mathbf{x}_i \in \mathbb{N}^{n_{bands}}$ (spectral vectors) contained in the HSI scene $\mathbf{X} \in \mathbb{N}^{n_1 \times n_2 \times n_{bands}}$ into a set of unique and mutually exclusive land cover classes (He et al., 2017a), obtaining the classification map $\mathbf{Y} \in \mathbb{N}^{n_1 \times n_2} \subset \{1, ..., n_{classes}\}$. Moreover, it is usual to binarize each category, performing the so-called one-hot encoding $\mathbf{Y} \in \mathbb{N}^{n_1 \times n_2 \times n_{classes}}$, so the mapping function $\mathbf{Y} = f_c(\mathbf{X}, \theta)$ assigns a vector label $\mathbf{y}_i \in \mathbb{N}^{n_{classes}}$ to each spectral pixel, $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{(n_1 \cdot n_2)}$.

In the literature, there is a vast amount of works about HSI data classification. Usually, these methods have been inspired by those algorithms and techniques developed in the fields of computer vision and pattern recognition, exhibiting a wide variety of methodologies and learning procedures. As a result, they can be divided in many groups depending on multiple factors, from unsupervised methods (for instance: k-means (Haut et al., 2017b), k-nearest neighbors -KNN- (Cariou and Chehdi, 2015) or iterative self-organizing data analysis technique -ISODATA- (Wang et al., 2014)) to supervised ones (support vector machines -SVMs- (Melgani and Bruzzone, 2004) or random forests -RFs- (Ham et al., 2005)), from statistical classifiers (such as multinomial logistic regression -MLR- (Haut et al., 2017a)) to deterministic methods (for instance, extreme learning machines -ELMs- (Li et al., 2018a)), from parametric algorithms (such as the maximum likelihood -ML- (Kuching, 2007)) to non-parametric ones (such as the evidential reasoning -ER- (Sanz, 2001)), from spectral-based methodologies (traditional distance metrics based classifiers (Du and Chang, 2001; Keshava, 2004), spectral angle mapper -SAM- (Camps-Valls, 2016; Calin et al., 2018), etc.) to spatial or spectral-spatial ones (sparse coding -SC- (Charles et al., 2011; Yang et al., 2014), morphological profiles -MP- (Fauvel et al., 2008; Huang and Zhang, 2013; Bhardwaj and Patra, 2018), among others). In this regard, several taxonomies have been proposed in order to categorize the available methods. For instance, Lu and Weng (2007) offered an interesting and complete taxonomy of thirteen categories, considering six different criteria, while Chutia et al. (2016) presented a simpler taxonomy of six different groups depending on the classification procedure. Also, Ghamisi et al. (2017a) provided a complex taxonomy with eight criteria and twenty categories, although none of them are exclusively dedicated to DL methods.

In fact, DL is a subfield of ML inspired by the structure and functions of the biological brain (Bengio, 2009; LeCun et al., 2015; Goodfellow et al., 2016), so those DL-HSI classifiers are often framed within the field of artificial neural networks (ANNs) (Plaza et al., 2011b), which are characterized by their flexible architecture, composed by groups (layers) of connected computational units (neurons). ANNs work on the basis that the global classification problem defined by f_c is split into several hierarchically ordered sub-mapping functions $\mathbf{Y} = f_c(\mathbf{X}, \theta) \approx \hat{f}(f^{(L)}(\cdots(f^{(1)}(\mathbf{X}, \theta^{(1)})\cdots), \theta^{(L)}))$, being L the number of layers that compose the network, $\mathbf{X}$ the original input data and $\hat{f}$ the final classifier (performed by a classification layer in end-to-end models or by any standard ML classifier). This is supported by the assumption that approximating a high number of small steps is better than solving a small number of large steps, implementing a "divide & conquer" strategy. In this context, each $f^{(l)}$ is of the general form defined by Eq. (1):

$$\mathbf{X}^{(l)} = f^{(l)}(\mathbf{X}^{(l-1)}, \mathbf{W}^{(l)}, b^{(l)}), \tag{1}$$

where weights $\mathbf{W}^{(l)}$ and biases $b^{(l)}$ are the parameters $\theta^{(l)}$ of the sub-mapping function $f^{(l)}$, and $\mathbf{X}^{(l-1)}$ and $\mathbf{X}^{(l)}$ are the input and output data, respectively. Moreover, ANNs are inspired by the neural connections that conform the biological brain's structure and the pulses that travel through synaptic connections to transmit information. In this sense, each $f^{(l)}$ is in fact composed by a set of neurons, which apply their corresponding synaptic weights over the input data, and whose responses are filtered, determining the neural activations which will be forwarded to the following $f^{(l+1)}$.

This hierarchical structure of stacked functions has fostered the rise of deep and very deep ANN models, as described in the outstanding and comprehensive overview presented in Zhang et al. (2016b). These models will be referred to hereinafter as DNNs and VDNNs. In this regard, although the limits between one type of network and another have not been established (Schmidhuber, 2015), there is an agreement among the experts to establish a distinction between shallow and deep architectures (Bengio et al., 2007b), whereby single-hidden layer structures are considered as shallow ANNs, architectures with two or more hidden layers are considered as DNNs, and models with dozens of layers are categorized as VDNNs (Srivastava et al., 2015). For instance, Hinton et al. (2006) presented a neural model with three hidden layers as one of the first deep architectures; Krizhevsky et al. (2012) considered their model with more than 5 layers as a deep network, and Simonyan and Zisserman (2014) introduced a VDNN with 16–19 layers. Following this trend, extremely deep neural networks (EDNN, also known as ultra-deep nets) have been introduced as architectures with more than 50 layers, reaching even thousands of layers (He et al., 2016). In this context, the stack of functions allows to extract data representations at different levels, which are processed by the successive neural layers. In fact, any ANN works as a feature extractor (FE), regardless of its depth, where each sub-mapping function encodes different characteristics from the input data. In general, these models' architecture allows for the learning of generic features at the early stages, a piece of knowledge that is traditionally considered as less dependent on the application, while the final layers are able to learn pieces of knowledge that are more related with the application at hand. This allows for the extraction of highly abstract data representations, which are directly obtained and refined by the classification problem itself, being modeled by each $\theta^{(l)}$ of the architecture. This also allows a higher flexibility in comparison with those methods that are subordinated to *hand-crafted features*, which should manually design the desired features, employing some well-known FE methods such as the scale invariant feature transform -SIFT- (Al-khafaji et al., 2018), histogram of oriented gradients -HOG-, local binary patterns -LBP- (Li et al., 2015b), or speeded-up robust features -SURF-, among others. This last procedure imposes several restrictions, in particular, the obtained features are very specific and usually exhibit limited levels of invariance and abstraction. Also, they are critically dependent on the user's knowledge, making hard to guarantee their setup (Yang et al., 2016).

In turn, the structure and functions of ANNs makes them universal approximators (Cybenko, 1989; Hornik, 1991), allowing them to learn any data system's behavior without any prior or additional information about the statistical distribution of the input data. In this sense, ANNs have attracted the attention of a large number of researchers in the area of HSI data classification (Benediktsson et al., 1993; Yang, 1999), and nowadays also in their DL version (Chen and Wang, 2014), due to the benefits that DNN models exhibit when compared to traditional ML methods (Collobert and Bengio, 2004):

1. The ability to extract hidden and sophisticated structures (both, linear and non-linear features) contained in the raw data. Such ability is intrinsically related, on the one hand, to the capacity to model their own internal representation (rather than having it prespecified, as handcrafted features by kernel functions (Camps-Valls et al., 2006)) and, on the other hand, to their ability for generalizing any kind of knowledge.

2. They are extremely flexible in the types of data they can support. In particular, they can take advantage of the spectral and spatial domains of HSI data, in both separate and coupled fashion.
3. Also, they offer a large flexibility on architectures, in terms of the type of layers, blocks or units, and their depth.
4. Moreover, their learning procedure can be adapted to a great variety of learning strategies, from unsupervised to supervided techniques, going through intermediate strategies.
5. Finally, advances in processing techniques such as batch partition and high performance computing (HPC) (Plaza et al., 2009; Lee et al., 2011; Bioucas-Dias et al., 2013), in particular on parallel and distributed architectures (Plaza and Chang, 2008; Plaza et al., 2011a), have allowed DNN models to scale better when dealing with large amounts of data.

These characteristics make DNNs very powerful and popular models for HSI data classification. However, as traditional ML approaches, DNNs are not exempt from certain limitations, which are highly related to the characteristics of HSI data.

2.2. Hyperspectral data challenges and deep learning limitations

ANN classifiers in general (and DL-based models in particular) need to face some challenges related to the processing of high-spectral dimensional data sets such as HSI data cubes. In fact, although the rich spectral information contained in each pixel $\mathbf{x}_i \in \mathbb{N}^{n_{bands}}$ is very useful to perform an accurate discrimination process, its large dimensionality brings new challenges, not only in terms of computation time and storage, but also due to the so-called peaking paradox (Theodoridis and Koutroumbas, 2003; Kallepalli et al., 2014; Sima and Dougherty, 2008). This paradox establishes that the use of additional features (i.e., spectral bands) brings complexity into the classifier, increasing the number of statistical parameters that define the land cover classes, and which must be estimated in advance. Following the previous notation, if we formulate the classification process as the approximation of a function $f_c: \mathbb{N}^{n_{bands}} \rightarrow \mathbb{N}^{n_{classes}}$ that identifies, for each spectral pixel $\mathbf{x}_i \in \mathbf{X}$, its corresponding label vector $f_c(\mathbf{x}_i) = \mathbf{y}_i$,we can infer that the corresponding estimation errors will increase when more parameters/features are taken into account, hampering the final classification performance (Landgrebe, 2005). This leads to the *curse of dimensionality* problem (Bellman, 2015) that greatly affects supervised classification methods, in which the size of the training set may not be sufficient to accurately derive the statistical parameters, thus leading the classifier to quickly overfit (Hughes phenomenon (Hughes, 1968)).

Coupled with their high dimensionality, HSI data presents several artefacts that make the classification process a difficult task. Similar to very high-resolution (VHR) images, HSI data also suffers a high intraclass variability, resulting from uncontrolled changes in the reflectance captured by the spectrometer (normally because of changes in atmospheric conditions, occlusions due to the presence of clouds, and variations in illumination, among other environmental interferers). Also, the instrumental noise produced by the spectrometer may degrade the data acquisition process, corrupting spectral bands to different degrees (Rasti et al., 2018), or even making several bands unusable due to saturation/cutoff or calibration errors (Pearlman et al., 2003). Also, there is a tendency in HSI instruments to include significant redundancy across adjacent spectral bands, which leads to the presence of redundant information that may hinder computational efficiency of analysis algorithms. Regarding the spatial information, pixels in HSI data often cover large spatial regions on the surface of the Earth in images with low/medium spatial resolution, so they tend to generate mixed spectral signatures, leading to high interclass similarity in border regions. In the end, these challenges create potential ambiguities and uncertainties (Varshney and Arora, 2004; Gomez et al., 2015) that must be faced by classification algorithms in order to extract representative features from the images.

Another important issue is the problematic lack of labelled data. In fact, despite the launch and start-up of the HSI-EO missions described on Section 1, the number of operational spaceborne spectrometers that are continuously acquiring images is still low in comparison with multispectral remote sensing sensors such as Landsat or the Sentinel missions, and in general the captured data are not publicly offered. Moreover, airborne spectrometers cover much smaller areas than those sensors allocated on satellite platforms, so the amount of HSI datasets is quite limited. In addition, the task of labelling each pixel contained in the HSI dataset is arduous and time-consuming, as it generally requires a human expert, further limiting the number of available HSI datasets for classification tasks.

These challenges greatly worsen the limitations already exhibited by DNN models (Nogueira et al., 2017), which are related to the complexity of the classifiers, such as the number of parameters required by deep models. In the following, we enumerate some of the aforementioned issues:

1. The training of DNNs is complex, since the optimization and the tuning of parameters in deep models is a non-convex and NP-complete problem (Blum and Rivest, 1989), much harder to train and without guaranteeing the convergence of the optimization process (Chen and Wang, 2014; Nguyen and Hein, 2018). Also, the increase in the number of parameters in deeper architectures often leads to multiple local minima (Bach, 2017).
2. Resulting from the large amount of parameters that must be managed in a deep model, there is a high computational burden involved, requiring computationally expensive and memory-intensive methods (Cheng et al., 2017b)
3. Also, due to the number of parameters that must be fine-tuned, supervised deep models consume great amounts of training data, and they tend to overfit when there are few training parameters (Erhan et al., 2010). In this context, the high-dimensional nature of HSI data, coupled with the limited availability of training samples, makes DNNs quite ineffective in generalizing the distribution of HSI data, requiring excessive adjustments at the training stage, while the performance on the test data is generally poor.
4. Moreover, simply stacking of layers by itself does not achieve the desirable improvement in precision results. In fact, forward propagation suffers from an important degradation of the data (He et al., 2016), while the backpropagation mechanism presents difficulties in propagating the activations and gradient signal to all layers as the network's depth increase (Srivastava et al., 2015). The gradient (which is necessary to update the model's parameters) fades slightly as it passes through each DNN layer. This degradation becomes quite severe in VDNNs, resulting in its practical disappearance or vanishing. These problems elongate the model's objective function until the model cannot properly change its weights at each iteration.
5. The "black box" nature of the training procedure is also a disadvantage, being the model's internal dynamics very hard to interpret (Benítez et al., 1997; Lipton, 2016). This may hinder the design and implementation of optimization decisions, although several efforts have been done to visualize the parameters of DNN models (Shwartz-Ziv and Tishby, 2017), and to enhance the extraction of more significant and interpretable filters.

The combination of the aforementioned challenges introduced by HSI data and the limitations of deep models force developers to carefully select and implement those models that best suit HSI data, choosing the architectures, learning strategies and improvement tricks that best fit the data while maintaining computational efficiency. In the following sections, these points will be covered in detail, providing a list of current models and methods that have been successfully applied to HSI data classification.

3. Deep neural networks: flexible and configurable models

Standard ANN models for HSI data classification exhibit a rather limited performing, usually conducting supervised learning of purely spectral features in a fully-connected architecture. On the contrary, DNNs offer a great variety of models, allowing for the inclusion of different layers, the exploitation of features in both the spectral and spatial domains, and the adoption of different learning strategies. In the following, these concepts will be briefly introduced.

3.1. Type of features

The type of features obtained from HSI data $\mathbf{X} \in \mathbb{N}^{n_1 \times n_2 \times n_{bands}}$ are one of the factors that impose several restrictions in the performance of the classifier, being crucial to the discrimination between the different classes. In particular, HSI data are characterized by their two spatial components: $n_1 \times n_2$, and by their large spectral domain, n_{band}, allowing the exploitation of both types of features (Landgrebe, 2002). Although there are also many traditional ML methods that allow for the exploitation of these two types offeatures, DNN models stand out for their versatility, adapting both their input and their internal operation to the use of such features through the implementation of different types of layers.

Focusing on traditional pixel-wise DNN classifiers, these methods exploit the ability of HSI data for detecting and uniquely characterizing the captured surface materials in certain land cover classes, learning existing relationships between the spectral signatures associated to each HSI pixel and the information that is contained in them (Chen et al., 2014b). In this sense, spectral-based DNN models learn spectral feature representations from $\mathbf{X}$, processing each pixel vector $\mathbf{x}_i \in \mathbf{X}$ in a way that is completely isolated from the rest of the pixels in the image (Romero et al., 2016), under the assumption that each $\mathbf{x}_i$ contains a perfect and pure signature of a single surface material, without any mixing of different land cover materials (Fisher, 1997). The performance and final accuracy of these classifiers is strongly related to the available training samples, usually requiring a large number of them to properly learn the parameters of the classifier (Hu et al., 2015) and to deal the spectral intraclass variability and interclass similarity –with the aim of avoiding the misclassification of the samples (traditional “salt & pepper” noise)– (Huang and Zhang, 2013).

To deal with these limitations, recent research has demonstrated the benefits of exploiting the spatial arrangement of HSI data (Jiménez et al., 2005; Zhang et al., 2012; Huang and Zhang, 2013), enhancing the classification performance of standard pixel-wise HSI classification procedures (Tarabalka et al., 2010; Mei et al., 2016) by analyzing the contextual information around each pixel $\mathbf{x}_i$ (Fauvel et al., 2008; Tarabalka et al., 2009; Bioucas-Dias et al., 2013). With advances in remote sensing technology, the spatial resolution has become gradually better, making HSI data cubes able to represent target zones/objects using finer spectral pixels and increasing the number of captured samples for each type of coverage (which intrinsically increases the intraclass variability), and improving the acquisition and observation of certain spatial patterns present in particular land cover materials. These classifiers operate under the assumption that adjacent pixels commonly belong to the same land cover category (Mura et al., 2010; Ghamisi et al., 2018), providing additional valuable information to the classification task which helps to reduce the intraclass variance and the label uncertainty. In the available literature, the contextual information given by the spatial arrangement of the HSI data cube can be employed by two kind of DNN classifiers: (i) those that only exploit the spatial features, and (ii) those that combine both spatial and spectral features to perform the final classification.

Focusing on spatial-based DNN classifiers, these models usually process some spatial information extracted from the original data cube $\mathbf{X}$, learning only spatial feature representations from the data (Chen et al., 2016). Although some spatial models may employ spatial handcrafted features as input data, such as the minimum noise fraction (MNF) (Zhang et al., 2019a) and covariance matrices (He et al., 2018), Gabor filtering (Chen et al., 2017b; Kang et al., 2018), among others, the most common and simple strategy to perform spatial HSI classification is to feed the network with some features extracted by the principal component analysis (PCA) method (Wold et al., 1987; Jolliffe, 2002; Fernandez et al., 2016), which reduces the spectral redundancy and the number of dimensions while keeping the spatial information intact (Yue et al., 2015; Haut et al., 2019a). In this context, although there is no consensus on the number of bands to be reduced, a DNN model is generally considered to be spatial when it applies PCA to the input data and its architecture allows only spatial features to be extracted (Makantasis et al., 2015; Chen et al., 2016; Haut et al., 2018c).

Although spatial-based DNN models may overcome spectral methods under some circumstances, in particular, in high spatial resolution HSI scenes with clear and distinctive spatial structures (and with spectral signatures that are not mixed) (Chen et al., 2016), the joint exploitation of both spatial and spectral features is more desirable, as it not only comprises the analysis of spectral signatures but also the associated contextual information (Paoletti et al., 2017a; Paoletti et al., 2018a). In this regard, available DNN architectures are able to process both features by including spatial information as concatenated information to the spectral vector (following the traditional ML vector vision (Chen et al., 2014b; Chen et al., 2015)), or by processing the 3-dimensional cube to maintain the original structure and contextual information (Chen et al., 2016; Paoletti et al., 2017a; Paoletti et al., 2018a; Paoletti et al., 2018c).

3.2. Type of layers

As mentioned before, the type of layer has a decisive influence on the architecture of the model, allowing for the processing of different features. Following the previous notation, DNN models divide the global mapping function $f_c(\cdot,\theta)$ into hierarchically stacked submapping functions $f^{(l)}(\cdot,\theta^{(l)})$. In this regard, each $f^{(l)}$ performs a two-step stage, composed by FE and detection, which are in turn implemented by several types of stacked layers, being $\theta^{(l)}$ their parameters.

Contextualizing the evolution of DL methods, at early days, neural models emerged within the fields of pattern recognition and signal processing, inspired by the behaviour of the biological brain and implementing a hierarchical structure where each part of the stack conforms a layer, being neurons (also perceptrons) the basic unit of each layer (Ball et al., 2017). However, with the development of image processing, traditional fully-connected structures became ineffective for the analysis of 2-dimensional and 3-dimensional data cubes (LeCun et al., 2015). To overcome this limitation, a fully-connected structure was adapted to the behaviour of those neurons that compose the biological visual cortex, characterized by a local receptive field in which they are activated or not in the presence of some specific visual *stimuli*, creating a hierarchical structure in which deeper neurons are able to respond to more abstract and higher level features. With this in mind, DNN models can implement several types of layers, where the most common ones are explained below.

3.2.1. Fully-connected layers

Also known as FC layers, they connect every neuron in the l-th layer to every neuron in the subsequent layer $l+1$, as it can be observed on the leftmost model in Fig. 1, where a traditional feed-forward multilayer perceptron (MLP) (Collobert and Bengio, 2004) is represented. These layers apply a linear transformation between the input layer data $\mathbf{X}^{(l-1)}$ and the layer parameters, weights $\mathbf{W}^{(l)}$ and biases $b^{(l)}$, adapting the original mapping function of Eq. (1) as follows:

$$\mathbf{X}^{(l)} = \mathbf{W}^{(l)} \cdot \mathbf{X}^{(l-1)} + b^{(l)} \tag{2}$$

The main drawback of FC layers is the high number of connections, imposing a large number of parameters that must be fine-tuned. In

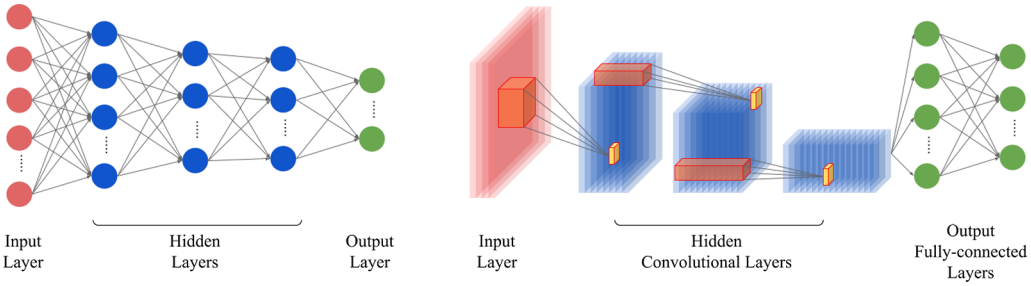


Fig. 1. Comparison between the traditional fully-connected (left) and the convolutional architecture (right) of a DNN model. The first model is represented as a conventional multilayer perceptron (MLP) with 3 hidden fully-connected (FC) layers, while the second model is represented as a convolutional neural network (CNN) with 3 hidden convolution layers too. Focusing on the last one, neurons in the CNN create 3-dimensional blocks with local connectivity over one pre-defined window of each layer input volume, known as receptive field. FC layers can be observed at the architecture tail, conforming the classifier network.

particular, the number of parameters can be calculated as the sum of all the connections between adjacent layers $n_{parameters} = \sum_{i=0}^{L-1} (n_{nodes}^{(l)} \cdot n_{nodes}^{(l+1)} + 1)$, which involves the number of weights and the bias. Also, both the input data that they need and the extracted features are limited to a vector representation of the input data, losing to some extent the potential of the spatial-contextual information (Chen and Wang, 2014).

3.2.2. Convolutional layers

As we can observe in Fig. 1, the CONV layer defines a block of neurons that operate as linear kernels (also called *filter bank*) connected and applied over small pre-defined regions from the input data (input volume hereinafter). The main idea lies on analyzing the statistical properties of the HSI cube $\mathbf{X} \in \mathbb{N}^{n_1 \times n_2 \times n_{bands}}$, which can be considered as a stationary source of spectral pixels in which data features are equally distributed into the entire $\mathbf{X}$ in relation to spatial positions (Field, 1999). This suggests that the learned features at a certain position of $\mathbf{X}$ can be successfully applied to other regions of $\mathbf{X}$, which in the end can be understood as the chance to employ the same features at all locations of the input image.

In this sense, CONV layers can be interpreted as a traditional sliding window method, where $K^{(l)}$ fixed-size filters are overlapped over the input layer data, sliding at certain intervals defined by the stride of the layer $s^{(l)}$. This can be observed in Fig. 2. In contrast with FC layers, CONV layers offer a great versatility, since the size of these chunks or windows is defined by the receptive field of the layer, indicated as $k^{(l)} \times k^{(l)} \times q^{(l)}$, where $k^{(l)}$ is applied over the two spatial axes and $q^{(l)}$ is applied over the spectral axis. This allows the CONV layer to accept 1-D, 2-D and 3-D inputs, and to extract spatial, spectral or spatial-spectral features.

$$\mathbf{X}^{(l)} = (\mathbf{W}^{(l)} * \mathbf{X}^{(l-1)} + b^{(l)})_{K^{(l)} \times k^{(l)} \times k^{(l)} \times q^{(l)}} \tag{3a}$$

$$x_{i,j,t}^{(l)z} = \sum_{\hat{i}=0}^{k^{(l)}-1} \sum_{\hat{j}=0}^{k^{(l)}-1} \sum_{\hat{t}=0}^{q^{(l)}-1} (w_{\hat{i},\hat{j},\hat{t}}^{(l)} \cdot x_{(i \cdot s^{(l)}+\hat{i}),(j \cdot s^{(l)}+\hat{j}),(t \cdot s^{(l)}+\hat{t})}^{(l-1)}) + b^{(l)} \tag{3b}$$

As Eq. (3a) indicates, the l-th CONV layer applies $K^{(l)}$ linear 3D-kernels over the input layer $\mathbf{X}^{(l-1)}$, which performs a dot product between its weights and biases, $\mathbf{W}^{(l)}$ and $\mathbf{b}^{(l)}$, respectively, and small chunks of the input volume data. As a result, an output volume $\mathbf{X}^{(l)}$ composed by $K^{(l)}$ feature volumes is obtained. In particular, Eq. (3b) indicates the general calculation of the feature (i, j, t) for the z-th feature of the output volume, $x_{i,j,t}^{(l)z}$.

CONV layers exhibit some advantages over traditional FC layers (Guo et al., 2016; Li et al., 2017b). In particular, the local connectivity allows to learn spatial correlations among neighboring pixels, introducing some invariance to the location of the feature. Also, the sparse connectivity and the parameter sharing mechanism reduces the number of parameters that must be fined-tuned.

3.2.3. Activation layers

Usually, the data transformations applied by FC and CONV layers are considered the FE stage of the network, defining a linear operation of element-wise matrix multiplication and addition over the data. In this sense, those DNN models without activation layers (or with linear activation ones) are essentially working as linear regressors. A non-linear activation layer must be implemented behind FC and CONV layers in order to learn non-linear representations of the data structure. In fact, the activation layer is considered as the detector stage of DNN models (Goodfellow et al., 2016), and is implemented by a non-linear, element-wise activation function which allows to model a response variable (i.e., a feature score) that varies non-linearly with the output volume of the previous FC/CONV layer, giving as a result an output volume containing the activations of each neuron of the previous layer, $\mathbf{X}^{(l)} = \mathcal{H}(\mathbf{X}^{(l-1)})$. In this regard, $\mathcal{H}(\cdot)$ can be implemented by several activation functions, depending on the desired properties. Fig. 3 gives the graphical visualization of some widely used functions. For instance,

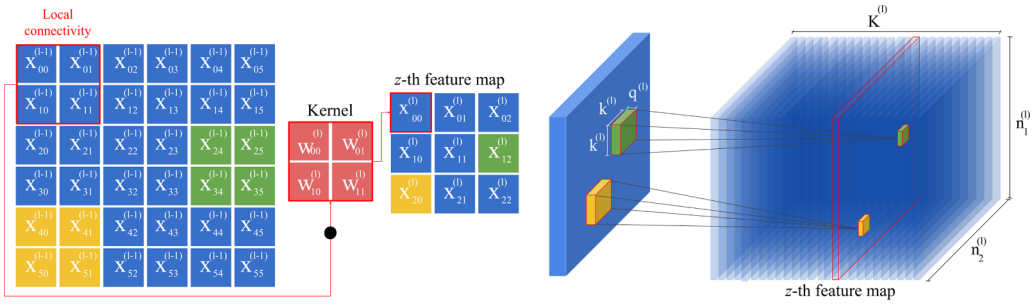


Fig. 2. Graphical visualization of the CONV layer from a 2D point of view (left) and 3D point of view (right). On the left we can observe how the 2D kernel is applied over spatial regions of the input volume $\mathbf{X}^{(l-1)}$ with a stride $s^{(l)} = 2$ (the dark circle symbolizes the dot product between the window from the original data and the kernel). On the right we can observe how the z-th kernel of size $k^{(l)} \times k^{(l)} \times q^{(l)}$ produces, for each region to which it is applied, a scalar value (represented as a smaller rectangle) which is allocated into the z-th feature map, composing an output volume $\mathbf{X}^{(l)}$ of $K^{(l)}$ feature maps.

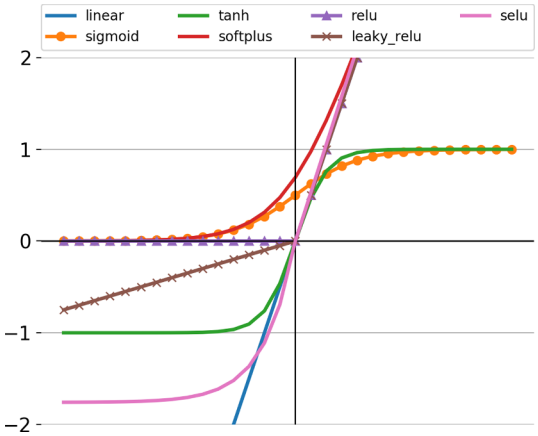


Fig. 3. Graphical visualization of different activation functions that can be implemented in a DNN model, including the linear function $\mathcal{H}(x) = x$, the *leaky* ReLU, and SeLU (Section 5.3.3 contains further details).

the *sigmoid* $\mathcal{H}(x) = \frac{1}{1+e^{-x}}$ presents a smooth and continuously differentiable function, whose values range from 0 to 1 (not inflating the neural activation values). However, as it only produces positive values in the 0–1 range, it becomes hard and slow to optimize. The *tanh function* $\mathcal{H}(x) = \frac{e^{x}-e^{-x}}{e+e^{-x}}$ is very similar to the sigmoid, being less smooth and symmetric over the origin, as its values range from -1 to 1. This makes its gradient steeper than that of the sigmoid.

Although these standard activations can operate properly with shallow architectures, the smallest derivative terms tend to zero when the model's architecture is deep enough, leading to the vanishing gradient problem. The *rectified linear activation function* (ReLU) (Nair and Hinton, 2010) tries to overcome previous limitations by applying a $\max(\cdot)$ function between 0 and the input data x, setting the gradient to 0 if the data are equal or smaller than 0, and to x otherwise, i.e. $\mathcal{H}(x) = \max(0, x)$, with an output range of $[0, +\infty)$ (it is unbounded on the positive side). This alleviates the vanishing gradient problem, as the derivative of the positive x is always 1. Moreover, ReLU conducts a sparsity activation function where not all the neurons are activated at the same time, being more computationally efficient than the sigmoid, for instance. However, if the gradient is set to 0, the influence of the affected neurons is eliminated, so they cannot contribute to improving the learning process (Pedamonti, 2018), leading to the *dying ReLU* problem.

Other interesting functions are the *softplus* (Dugas et al., 2001) and *softmax* activation functions, with equations $\mathcal{H}(x) = \ln(1 + e^{x})$ and $\mathcal{H}(x_i) = \frac{e^{xi}}{\sum_j e^{xj}}$, respectively. The first one produces values in the range $[0, +\infty)$, i.e. it is similar to a smoothed ReLU being differentiable into 0 and where its derivative is the sigmoid function, which makes it computationally slower during the backward step. The second one is inspired by the sigmoid, squeezing the input layer data between 0 and 1 and dividing the obtained outputs by the sum of them. In this sense, the softmax function works as a winner-take-all function that gives the probability of the input data belonging to a particular class, and it is usually employed as the final layer of a DNN model.

Despite the wide range of activation functions available in the current DL literature (Agostinelli et al., 2014; Sonoda and Murata, 2017; Ramachandran et al., 2017), the vast majority of DNN models for the analysis of HSI remote sensing data employ ReLU and softmax as the principal activation functions, with few exceptions (Mei et al., 2016; Paoletti et al., 2018).

3.2.4. Down-sampling layers

Also known as pooling or POOL layers, they are inspired by the spatial processing of CONV layers. Particularly, POOL layers perform a non-linear sub-sampling strategy with the aim of: (i) reducing the spatial dimensions of the extracted feature maps, sumarizing them into a reduced volume, (ii) contributing to the data with certain invariance to small transformations, and (iii) reducing the computation time and the complexity in terms of both, data size/dimensionality and network parameters (Boureau et al., 2010). The POOL layer implements a sample-based discretization process (see Fig. 4), applying some numerical operation over a square window defined by the spatial receptive field $k^{(l)} \times k^{(l)}$ of the layer. The most usual operations are the average-pooling, the sum-pooling or the max-pooling (Scherer et al., 2010), although it should be noted that several alternative methods have been also implemented, such as stochastic pooling (Zeiler and Fergus, 2013), mixed pooling (Yu et al., 2014) or wavelet pooling (Williams and Li, 2018). Also, several works have investigated the replacement of pooling layers by CONV layers with increased stride (Springenberg et al., 2014).

3.3. Learning strategies

In addition to the type of features and layers used, DNN models also allow the implementation of different learning strategies. Following the previous notation, the classification function $f_c(\cdot,\theta)$ can be understood as a particular DNN model. In this sense, the performance of f_c will depend on certain parameters θ that must be correctly fine-tuned. Moreover, depending on how this parameter adjustment is carried out, two main types of learning can be distinguished: *unsupervised* and *supervised* learning

3.3.1. Unsupervised learning

Unsupervised learning performs the classification without *a priori* knowledge about the given data, optimizing parameters θ by the inherent similarities present in the data structure (Xiaoli Jiao, 2007; Romero et al., 2016; Hassanzadeh et al., 2018). In this context, an unsupervised DNN performs a *greedy layer-wise unsupervised pre-training* (Bengio et al., 2013), where each layer performs hierarchical unsupervised FE for inferring the underlying structure of the data, being

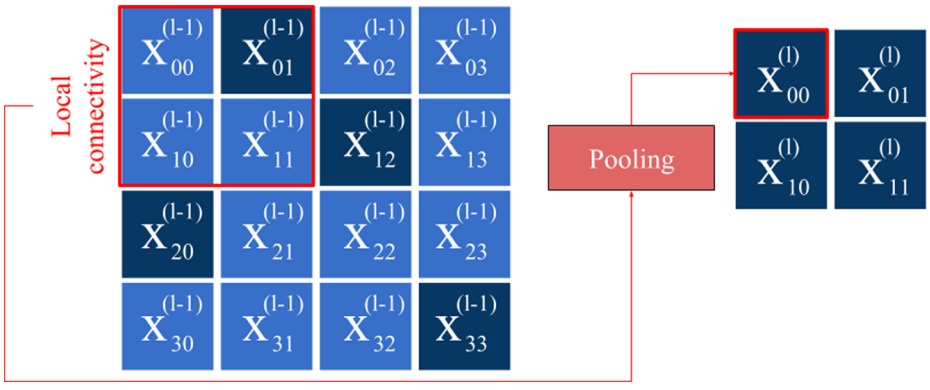


Fig. 4. Graphical visualization of the POOL layer from a 2D point of view. Dark cells indicate the selected value from the pooling operation (for instance, if the max pooling has been implemented, dark cells would represent the higher value of each region from the volume), although the final value also can be obtained as the average or sum value of the entire region. In fact, the pooling layer can be interpreted as a kernel of size $k^{(l)} \times k^{(l)}$.

combined at the end to initialize another deep supervised or generative model (Bengio et al., 2012) that will carry out the final regression or classification task. In fact, unsupervised DNN models are usually employed for clustering (Xie et al., 2016; Tian et al., 2017; Shaham et al., 2018; Min et al., 2018), anomaly detection (Penttilä, 2017; Ma et al., 2018a) and data encoding (Li et al., 2014; Paul and Kumar, 2018; Kang et al., 2018). In particular, there is a wide range of works about unsupervised DL methods for HSI data pre-processing, being widely used to perform dimensionality reduction (DR) (Lin et al., 2013b).

3.3.2. Supervised learning

In contrast to unsupervised learning, supervised learning needs to learn those parameters θ that model the relationship between $\mathbf{x}_i$ and $\mathbf{y}_i$ by performing an inference procedure based on previous knowledge (Sabale and Jadhav, 2015; Qiu et al., 2017). In this way, it is needed to split the original scene $\mathbf{X}$ into those training samples with known identity $\mathcal{D}_{train} = \{\mathbf{x}_i, y_i\}_{i=1}^{n_{labeled}}$ that will be used during the training step to categorize the rest of unlabeled pixels, which will be employed during the inference $\mathcal{D}_{test} = \{\mathbf{x}_i\}_{i=1}^{n_{unlabeled}}$ (Sabalel and Jadhav, 2014). Usually, supervised DNNs models are able to achieve better performance than their unsupervised counterparts, being the most widely used (Chen and Wang, 2014; Sabale and Jadhav, 2015; Qiu et al., 2017; Paoletti et al., 2017a; Paoletti et al., 2018a). However, this learning also imposes a severe training constraint, whereby DNN models need to consume large amounts of labelled data during the training to correctly fine-tune the model parameters (Makantasis et al., 2015).

4. When HSI data meets DL: main classification models

To date, in addition to the traditional MLP (Roodposhti et al., 2019), four DNN models have become the mainstream DL architectures for the analysis of HSI data: autoencoders (AEs), deep belief networks (DBNs), recurrent neural networks (RNNs) and CNNs. In the following we review each model, pointing the most relevant works in the HSI literature, and then paying more attention to state-of-the-art models such CNNs.

4.1. Autoencoders (AEs)

When dealing with HSI data classification tasks, the extraction of accurate features becomes a critical preprocessing step to model the internal structures and relationships of the data, helping to reduce the Hughes effect and the curse of dimensionality. In this sense, auto-associative neural networks (AANNs), also known as autoencoders (AEs) (Hinton and Zemel, 1993; Bishop, 1995; Chen et al., 2014b; Karhunen et al., 2015; Zhang et al., 2016c; Plaut, 2018) have been widely used as deep models to perform unsupervised coding from HSI data. Regarding its operational mode, the AE model does not carry out classification tasks, but reconstructs the input data by reducing $\min\|\mathbf{X} - \mathbf{X}'\|_2$ (the distance between the obtained representation $\mathbf{X}'$ and the original data $\mathbf{X}$). In fact, the main particularly of these networks lies in their ability to project the original input samples into a new space, generating compressed, extended or even equally-dimensioned outputs, with the least possible amount of distortion. This projection is performed by a traditional architecture implemented by encoder and decoder nets, both linked by a bottleneck layer that represents the latent space (Baldi and Hornik, 1989; Hinton and Zemel, 1993), as it can be observed in Fig. 5.

HSI-AEs emerged as typically pixel-wise methods, being usually exploited to carry out dimensionality reduction (DR) and high-level spectral FE due to the existing correlation between adjacent bands (Ahmad et al., 2017). In this regard, the spectral pixel $\mathbf{x}_i \in \mathbb{N}^{n_{bands}}$ is taken as input of the encoder, representing it in a new space $\mathbb{R}^{n_{new}}$ by applying a hierarchical set of $L_{encoder}$ recognition weights or encoder components, as Eq. (4a) illustrates. Then, the obtained code vector or code dictionary $\mathbf{c}_i \in \mathbb{R}^{n_{new}}$ is sent as an input to the decoder, which applies a set of $L_{decoder}$ generative weights over the code vector to recover and/or obtain an approximate reconstruction of the original input vector, $\mathbf{x}_i'$, as Eq. (4b) indicates.

$$\mathbf{c}_i \leftarrow \text{For } \; l \text{ in } L_{encoder}: \quad \mathbf{x}_i^{(l+1)} = \mathcal{H}(\mathbf{W}^{(l)} \cdot \mathbf{x}_i^{(l)} + b^{(l)}) \tag{4a}$$

$$\mathbf{x}_i' \leftarrow \text{For } \; ll \quad \text{in} \quad L_{decoder}: \quad \mathbf{c}_i^{(ll+1)} = \mathcal{H}(\mathbf{W}^{(ll)} \cdot \mathbf{c}_i^{(ll)} + b^{(ll)}) \tag{4b}$$

Several AE models for HSI data analysis have been presented in the literature. Focusing on spectral-based ones, Zhu et al. (2017a) propose an unsupervised tied AE (TAE) for spectral FE, based on the maximum noise fraction (MNF) (Green et al., 1988; Iyer et al., 2017) as pre-processing DR step, and fine-tuning with classification via softmax. Following a simple architecture, Hassanzadeh et al. (2017) combine the multi-manifold spectral clustering (MMSC) (Wang et al., 2010) with the unsupervised contractive AE (CAE) (Rifai et al., 2011) to enhance the HSI data classification by reinforcing the model's learning through a regularizer term, being less sensitive to small variations in the training samples. A pixel-wise stacked AE (SAE) is proposed by Okan et al. (2014), which implements a two-step training strategy with unsupervised representation learning and supervised fine-tuning, before the final supervised classification, performed by a logistic regression layer. Furthermore, Wang et al. (2016) implement a stacked denoising AE (SDAE), which stochastically corrupts the inputs in order to overcome the *identity-function risk* present in deep AEs. Also, in order to reduce the computational complexity of SAEs, Zabalza et al. (2016) propose a segmented SAE (S-SAE) to comprise original features into smaller data segments, being separately processed by smaller and independent SAEs.

Also, recent works combine AEs with spectral-spatial feature extraction methods. For instance, Chen et al. (2014b) present three different AEs and SAEs to generate shallow and deep or high-level features using spectral, spatial and spectral-spatial information, using a logistic regression method to perform the final classification, while Lin et al. (2013b) perform a comparison between spectral and spectral-spatial AEs with shallow and deep architectures. In both cases, the spatial information is obtained via PCA reduction, obtaining n_{new} components and flattening the $d \times d \times n_{new}$ cube that surrounds each pixel into a vector. Mughees et al. (2016) also develop a SAE to perform spectral processing, while spatial analysis is performed by an adaptive boundary adjustment-based segmentation method. As a result, the spectral-based classification map and the spatial-based single band segmented map are combined by a majority voting based method. Wang et al. (2017b) apply guided filtering (He et al., 2013) to exploit the spatial information, flattening it to combine it with spectral information in a multilayer fine-tuning SAE (FSAE). Li et al. (2015a) implement a SAE, which is pre-trained in unsupervised fashion over 3D Gabor features extracted from the HSI data cube, with an MLP performing the final classification. Ma et al. (2016b) combine the FE performed by the SAE with a relative distance prior in the fine-tuning process, in order to enhance the model when the number of available labeled samples is not enough. Also, Ma et al. (2016a) introduce a spatial updated deep auto-encoder (SDAE) to improve the extraction of spectral-spatial information by adding a regularization term in the energy function, and updating the features by integrating contextual information. Paul and Kumar (2018) propose a segmented stacked autoencoder (S-SAE) for spectral-spatial HSI data classification as an improvement of the SAE, reducing its complexity and computational times through the use of mutual information (MI), to perform spectral segmentation, and morphological profiles (MPs) to assimilate the spatial information contained in the HSI cube. Tao et al. (2015) develop two stacked sparse AEs (SSAEs) to extract overcomplete sparse spectral and spatial features, which are stacked and embedded into a linear SVM for classification purposes. Wan et al. (2017) also propose a SSAE to process different types of features, such as spectral-spatial, multifractal and other higher-order statistical ones, while a RF is employed for classification. Zhao et al. (2017a) exploit again the SSAE with RF to extract and classify more abstract and deep-seated

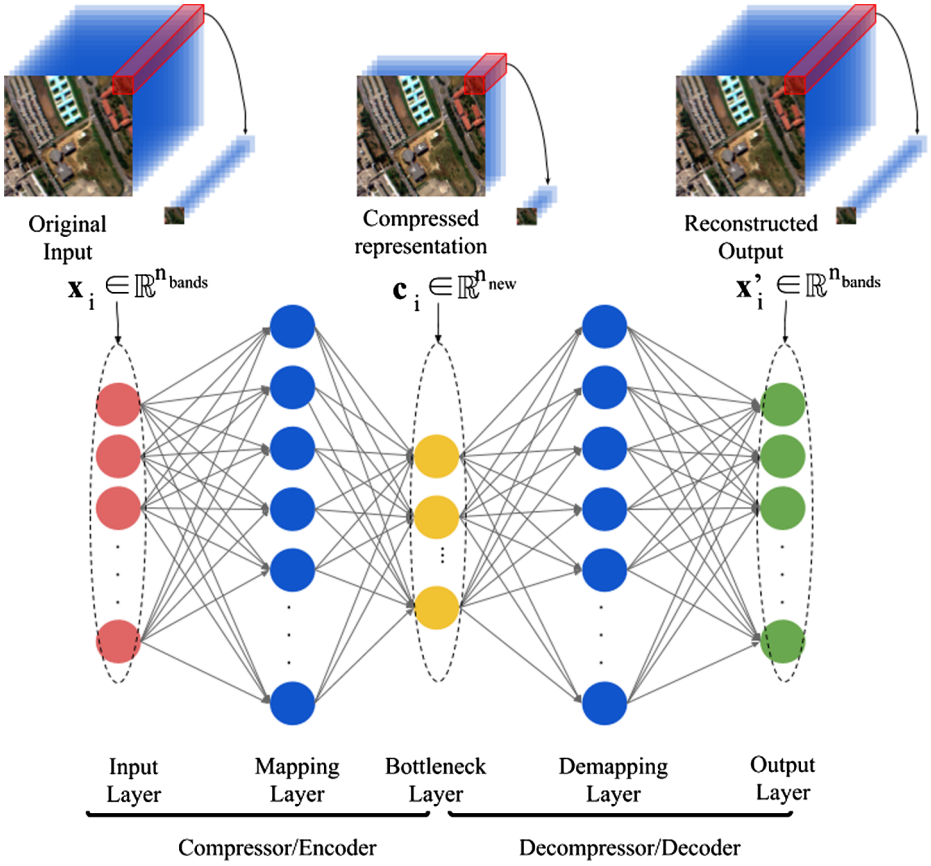


Fig. 5. Traditional representation of a tied autoencoder, composed by two main parts: an encoder and a decoder, linked by a bottleneck layer.

features from spectral, spatial and spectral-spatial sets. In contrast, Xing et al. (2016) develop a SDAE to extract robust spectral features from HSI data, using logistic regression to perform the supervised fine-tuning and classification. Liu et al. (2015) also employ a SDAE to learn spectral feature representations from the input data, while a superpixel technique is employed to generate the spatial constraints for refining the spectral classification results. Recently, Zhou et al. (2019a) have proposed a two-stage AE, called compact and discriminative SAE (CDSAE), where the first one performs the training of a discriminative SAE (DSAE, where each layer performs a local Fisher discriminant regularization) to learn a feature mapping by minimizing the reconstruction error, and the second one performs the classification of the data, updating the DSAE's parameters. Also, the extraction of spectral features using AEs has been combined with neural models such as CNNs (to extract spatial information), as Hao et al. (2018) discussed.

Although AE structures have demonstrated to be a powerful tool, their performance is often hampered by the large number of parameters that must be trained, learned and updated, which requires a large number of samples to perform the fine-tuning process, a demand that cannot be always satisfied. Although several new techniques have been adopted to avoid this problem, such as the use of active learning (AL) (Li, 2015), additional enhancements are needed. Furthermore, the spatial processing step that AEs usually perform on the data implies the use of DR methods followed by a flattening of the data into a vector, neglecting the rich spectral-spatial structural information that HSI data cubes contain (Chen et al., 2014a; Tuia et al., 2015).

4.2. Deep belief networks (DBNs)

DBNs combine probability and graph theory to implement a generative probabilistic graphical model (PGM) with the structure of a directed acyclic graph (DAG) (Ball et al., 2017). In the literature, several works address the implementation of DBNs as a stack of unsupervised networks, such as restricted Boltzmann machines (RBMs) (Smolensky, 1986; Larochelle and Bengio, 2008; Tan et al., 2019) with a greedy learning algorithm as optimizer (Hinton et al., 2006; Hinton and Salakhutdinov, 2006).

In HSI data analysis, DBNs have been employed as a variant of the AE model with greedy layer-wise training to perform FE. In this sense, Li et al. (2014) implement a DBN for feature extraction and classification, stacking spectral-spatial characteristics and using logistic regression for classification. Also, Chen et al. (2015) introduce three DBNs to extract spectral, spatial and spectral-spatial high-level features from HSI data in hierarchical fashion, and performing the final classification task by means of logistic regression. There are also several efforts aimed at improving the performance of this kind of DNN for HSI classification purposes, for instance Le et al. (2015) review the hyper-parameters used by the spectral and spectral-spatial DBNs of Chen et al. (2015), while Zhong et al. (2017a) present a diversified DBN for HSI data classification, which regularizes the pre-training and fine-tuning procedures by a diversity-promoting prior over latent factors to avoid the co-adaptation of the latter. Guofeng et al. (2017) improve the standard training process of DBNs in order to avoid the effect of a gradient disappearance, using PCA and kernel PCA (KPCA). Inspired by DBNs, Zhou et al. (2017) developed a group belief network (GBN), which considers the characteristics of grouped spatial-spectral features from HSI data by modifying the bottom layer of each RBM that composes the model architecture.

Although DBNs are very promising DL methods for HSI data classification, as they often provide good results (and improve their performance with the incorporation of spatial information), they suffer from the same limitation as SAEs: these neural models are designed for processing 1D-signals, so the rich spatial information contained in HSI data cubes must be vectorized to be processed together with the

spectral one, or even separately processed by other techniques in order to be properly exploited. In the end, this kind of spectral-spatial processing cannot fully incorporate the spatial-contextual information present in HSI data cubes.

4.3. Recurrent neural networks (RNNs)

The architecture of RNN models (Williams and Zipser, 1989) is characterized by loops in the connections, where node-activations at each step depend on those of the previous step. This internal structure (similar to a directed graph) makes the RNN an ideal model for learning temporal sequences, exhibiting a dynamic temporal behavior for a given data sequence, with an internal state or memory that allows for the association between the current input data and the previous ones at each step (i.e. remembering the context). This fact enables RNNs as a powerful tool for predicting future events depending on the previously remembered ones, being particularly interesting for remote sensing land-cover analysis, which exhibits many changes in their reflective characteristics over time, hampering the classification task (Rußwurm and Körner, 2017).

RNNs can be categorized into three main groups: vanilla RNNs, long short term memory (LSTM) (Hochreiter and Schmidhuber, 1997) and gated recurrent unit (GRU) (Cho et al., 2014) architectures. The vanilla RNN was the first recurrent model introduced as a DL framework, and its operation is quite intuitive. Given an input data sample $\mathbf{x}_t \in \mathbb{R}^n$ captured at time t, the vanilla RNN computes its corresponding output $\mathbf{y}_t$ as a hidden state at time t, $\mathbf{y}_t = \mathbf{h}_t$, which represents the current memory of the model, as Eq. (5) indicates:

$$\mathbf{h}_t = \begin{cases} 0 & \text{if } t = 0 \\ \mathcal{H}(\mathbf{W}_h \cdot \mathbf{x}_t + \mathbf{U}_h \cdot \mathbf{h}_{t-1} + b_h) & \text{if } t \neq 0 \end{cases} \tag{5}$$

where $\mathcal{H}(\cdot)$ is a non-linear activation function (for instance, the sigmoid), b_h is the bias, $\mathbf{W}_h$, is the weight matrix of the input, and $\mathbf{U}_h$ is the weight matrix of the recurrent connections.

Although vanilla is the easiest RNN model to implement, its simplicity leads to a degradation of information when high dimensional input data are processed. In this sense, the LSTM offers advantages when dealing with the deficiencies of the original RNN by developing a recurrent unit composed by a cell, which remembers values at arbitrary time intervals, and three gates (input, output and forget gates), intended to regulate the flow of information in and out of the cell. A schematic overview is presented in Fig. 6. In this case the model stores, for each input at time t, two states: the original hidden state, $\mathbf{h}_t$, and the cell state, $\mathbf{c}_t$, which removes or adds information to the cell, depending on the gates. In particular, the input gate $\mathbf{i}_t$ determines whether or not a new input is allowed to go inside the cell, the forget gate $\mathbf{f}_t$ deletes the irrelevant or unimportant information, and the output gate $\mathbf{o}_t$ allows the information to affect the network's output at time t. This mechanism allows the LSTM unit to learn which information is important along time, as Eq. (6) indicates:

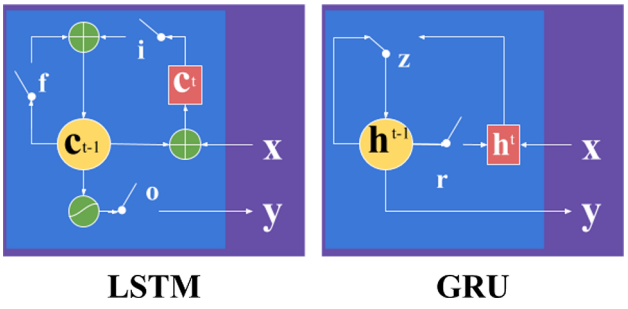


Fig. 6. Architecture of RNN models: comparison between the internal architecture of a LSTM recurrent unit and a GRU one.

$$\begin{aligned}
\mathbf{i}_t &= \mathcal{H}(\mathbf{W}_i \cdot \mathbf{x}_t + \mathbf{U}_i \cdot \mathbf{h}_{t-1} + b_i) \\
\mathbf{f}_t &= \mathcal{H}(\mathbf{W}_f \cdot \mathbf{x}_t + \mathbf{U}_f \cdot \mathbf{h}_{t-1} + b_f) \\
\mathbf{o}_t &= \mathcal{H}(\mathbf{W}_o \cdot \mathbf{x}_t + \mathbf{U}_o \cdot \mathbf{h}_{t-1} + b_o) \\
\mathbf{h}_t &= \begin{cases} 0 & \text{if } t = 0 \\ \mathbf{o}_t \circ \mathcal{H}(\mathbf{c}_t) & \text{if } t \neq 0 \end{cases} \\
\mathbf{c}_t &= \begin{cases} 0 & \text{if } t = 0 \\ \mathbf{f}_t \circ \mathbf{c}_{t-1} + \mathbf{i}_t \circ \mathcal{H}(\mathbf{W}_c \cdot \mathbf{x}_t + \mathbf{U}_c \cdot \mathbf{h}_{t-1} + b_c) & \text{if } t \neq 0 \end{cases}
\end{aligned} \tag{6}$$

where $\mathbf{W}_*$, $\mathbf{U}_*$ and b_* are the weight matrices and biases for the different gates or the cell (depending on $*$).

Finally, the GRU unit is a LSTM variant (see Fig. 6) in which the input and forget gates are changed by update ($\mathbf{z}_t$) and reset ($\mathbf{r}_t$) gates, removing the output gate (which implies less parameters):

$$\begin{aligned}
\mathbf{z}_t &= \mathcal{H}(\mathbf{W}_z \cdot \mathbf{x}_t + \mathbf{U}_z \cdot \mathbf{h}_{t-1} + b_z) \\
\mathbf{r}_t &= \mathcal{H}(\mathbf{W}_r \cdot \mathbf{x}_t + \mathbf{U}_r \cdot \mathbf{h}_{t-1} + b_r) \\
\mathbf{h}'_t &= \tanh(\mathbf{W}_h \cdot \mathbf{x}_t + \mathbf{r}_t \circ \mathbf{U}_h \cdot \mathbf{h}_{t-1} + b_h) \\
\mathbf{h}_t &= \begin{cases} 0 & \text{if } t = 0 \\ \mathbf{z}_t \circ \mathbf{h}_{t-1} + (1 - \mathbf{z}_t) \circ \mathbf{h}'_t & \text{if } t \neq 0 \end{cases}
\end{aligned} \tag{7}$$

As any traditional pixel-based approach, the RNN exploits each HSI pixel in band-to-band fashion, performing a similarity check between temporary data and spectral bands, using a many-to-one scheme such as the LSTM and GRU models for HSI data processing presented by Mou et al. (2017). Also, Guo et al. (2018) propose a LSTM model with a guided filter, taking into account three principal components extracted by PCA, and Zhou et al. (2018a) combining spectral LSTM-classification with PCA extracted spatial LSTM-classification via decision fusion. Lyu et al. (2016) develop the REFEREE change rule for a LSTM-based model in order to enhance the efficiency and performance when dealing with change detection in multispectral and HSI data. Zhang et al. (2018b) introduce the LSS-RNN, a RNN model with a local spatial sequential method (LSS) that includes a low-level FE step, implemented using Gabor filtering and differential morphological profiles (DMPs) (Huang and Zhang, 2013), whose corresponding features are stacked together and passed through the LSS to obtain higher-level features, which finally feed the RNN. Furthermore, Sharma et al. (2018) enhances the pixel-based RNN by implementing a patch-based RNN (PB-RNN) with LSTM units, which is able to process the multi-spectral, multi-temporal and spatial information contained into the dataset.

Other interesting RNN models take advance of the flexibility offered by CONV layers, including some stages of FE and detection with CONV after applying the recurrent unit. For instance, Venkatesan and Prabu (2019) employ a RNN to classify the features obtained by a spectral CNN model (developing a CNN1D), while Luo (2018) proposed a shortened spatial-spectral RNN with Parallel-GRU (St-SS-pGRU) with the aim of improving performance, increasing efficiency and simplifying the training procedure of standard band-by-band GRU models. Zhou et al. (2018b) first perform a spatial FE with CNNs, and then send the obtained features to a fusion network based on GRUs. Mou et al. (2019) implement a multi-spectral-temporal-spatial model for change detection by adopting an end-to-end network with several CONV layers (at the beginning of the architecture) in order to extract spectral-spatial features in a natural and structured way, enhancing the data representation before applying the LSTM unit and a FC layer to perform the final classification. Moreover, Shi and Pun (2018) combine the feature extraction performed by the spectral-spatial CNN model with a multi-scale hierarchical recurrent neural network (MHRNNs) that captures the spatial relations of local spectral-spatial features at different scales. Also, several convolutional RNNs (CRNNs) (Zuo et al., 2015) have been implemented for HSI classification. For instance, Wu and Prasad (2017) present a 1D-CRNN, where several 1D-convolutional layers are used to perform spectral FE, sending the obtained features to the recurrent layers, and finally integrating spatial constraints by adding linear opinion pools (LOP) (Benediktsson and Sveinsson, 2003) at the end of the flowchart in order to improve classification

performance. A similar model is used by Wu and Prasad (2018), where a 1D-CRNN is trained in a semi-supervised way with labeled and unlabeled data using pseudo labels. Yang et al. (2018) introduce the 2D-CRNN and 3D-CRNN for HSI data classification, performing a direct comparison with their CNN counterparts and demonstrating the superiority of the proposed RNN models. Finally, Liu et al. (2017d) introduce a bidirectional-convolutional LSTM (Bi-CLSTM) to learn spectral-spatial features from HSI data, while Seydgar et al. (2019) integrate a CNN model with 3D-kernels and the CLSTM network to extract low-dimensional and shallow spectral-spatial features that are recurrently analyzed, focusing on the spatial information but also considering the spectral one.

4.4. Convolutional neural networks (CNNs)

In contrast with the previous models, in which the FC layer is the basis of their architectures, in the CNN model (Lecun et al., 1998) the CONV layer is the basic structural unit, inspired by the natural vision process to perform FE (LeCun et al., 2015; Goodfellow et al., 2016). In this sense, CNN models elegantly integrate spectral features with spatial-contextual information from HSI data in a more efficient way as compared to previous DNN models. The large flexibility that this model provides regarding the dimensionality of the operational layers, their depth and breadth, and its ability to make strong assumptions about the input images (Krizhevsky et al., 2012), have turned the CNN into one of the most successful and popular DNN models, being the current state-of-the-art in DL (Gu et al., 2018) and an extremely popular tool for HSI data classification.

The architecture of a CNN is composed by two well-differentiated parts that can be interpreted as two networks. These coupled networks are trained together as an end-to-end model to optimize all the weights in the CNN: (i) the FE-net, composed by a hierarchical stack of feature extraction and detection stages that learns high-level representations of the inputs, and (ii) the classifier, composed by a stack of FC layers that performs the final classification task, computing the membership of each input sample to a certain class (Ball et al., 2017).

Focusing on the FE-net, it is composed by several hierarchically stacked extraction and detection stages, where the l-th stage defines the l-th submapping function $f^{(l)}$. Usually, these submapping functions are composed by CONV, activation or ReLU, and POOL layers (Murugan, 2017). In this way, the CNN model is able to reveal the features that are shared across the data domain via localized kernels, extracting the local stationarity properties of $\mathbf{X}$ (Defferrard et al., 2016). In fact, the feature extraction performed by the CNN is very similar to the ones adopted by other DNN models, i.e. the first stages are able to detect *recognizable* features, while the last stages combine all the features detected by the previous layers, detecting more *abstract* features. However, the flexibility when designing kernels allows for a more efficient and natural extraction of spectral, spatial and spectral-spatial features, as Fig. 7 shows, while the locally-connected nature of the convolutional kernels, coupled with the parameter-sharing across layers, permits to alleviate the number of parameters that must be fine-tuned by the model, making the computations more efficient as compared with traditional FC architectures. Focusing on the classifier net, it performs the final classification taking into account the information obtained by the FE-net. Usually, this part is implemented by several stages composed by FC and ReLU layers, placing a softmax on the last FC layer. The resulting output can be interpreted as the probability that each input data pattern belongs to a certain class, where the optimization function can be defined as the difference between all the desired outputs $\mathbf{y}_i$ (for each input data sample $\mathbf{x}_i$) and the obtained ones, $\mathbf{y}'_i$, which can be calculated as the cross-entropy of the data:

$$\phi_c = -\sum_i \mathbf{y}'_i - \log(\mathbf{y}_i) \tag{8}$$

Moreover, the classifier net can be implemented by a standard MLP model, or by other classifiers such as SVM (Paoletti et al., 2017b) or logistic regression (Zheng et al., 2017). Also, the classifier can be disregarded, using the first part, i.e. the FE-net, for other purposes such as unsupervised FE (Romero et al., 2016). In the current literature, three kinds of CNN models can be found for HSI data classification, depending on whether they perform spectral, spatial, or spectral-spatial feature analysis. In the following, we review some available works in each category.

4.4.1. Spectral CNN models for HSI data analysis

Regarding spectral models (top of Fig. 7), they consider the spectral pixels $\mathbf{x}_i \in \mathbb{N}^{n_{channels}}$ as the input data, where $n_{channels}$ can either be the number of original bands n_{bands} or a reasonable number of spectral channels n_{new}, extracted using PCA or other DR methods, to which 1D-kernels are applied on each CONV layer, $K^{(l)} \times q^{(l)}$, obtaining as a result an output $\mathbf{X}^{(l)}$ composed by $K^{(l)}$ feature vectors.

Hu et al. (2015) and Salman and Yüksel (2016) present a deep CNN with five 1-D layers that receive as input data the pixel vectors, classifying HSI data cubes only in the spectral domain, while Charmisha et al. (2018) present a CNN1D architecture called vectorized CNN (VCNN) to perform DR and classification of HSI data based on the topology of Hu et al. (2015). Li et al. (2017a) propose a CNN1D model for exploring spectral information correlated between pixels, extracting pixel pair features (PPFs) from the original data, being the input a combination of the center pixel and each of its surrounding neighbors (exploiting the similarity between pixels). Similarly, Du and Li (2018) develop subtraction PPFs, where the CNN1D model's input is the spectral difference between the central pixel and its adjacent pixels, performing HSI target detection. Mei et al. (2016) train the model by considering the spectrum of the pixel, the spectral mean of neighboring pixels, and the mean and standard deviation per spectral band of the neighboring pixels, introducing several improvements into the CNN1D architecture, such as batch normalization layers (Xu et al., 2015b), a dropout process (Krizhevsky et al., 2012) and a new nonlinear activation function known as Parametric ReLU (PReLU) (He et al., 2015). Acquarelli et al. (2018) develop seven shallow CNN1D models with spectral-locality-aware regularization (R), smoothing-based data augmentation (S) and label-based data augmentation (L), to include some kind of spatial information into the network, creating seven combinations (CNN-R, CNN-S, CNN-L, CNN-RS, CNN-RL, CNN-SL and CNN-RSL), although the spectral pixels are processed independently, i.e. one by one. Finally, Ghamisi et al. (2017a) and Chen et al. (2016) present standard CNN1D models for spectral processing.

In addition to 1-D architectures, the CNN2D architecture can be adapted to work only with spectral information. For instance, Jia et al. (2016) take into account only the pixel spectral array $\mathbf{x}_i$, which is folded into a map matrix and sent to the CNN2D as input.

4.4.2. Spatial CNN for HSI data analysis

Regarding spatial models, they only consider spatial information obtained from the HSI data cube. In this sense, it is usual to employ CNN2D architectures to process the spatial information, where each CONV layer applies $K^{(l)} \times k^{(l)} \times k^{(l)}$ kernels over the input data, obtaining as a result $K^{(l)}$ feature maps.

The spatial information can be extracted from the original HSI data cube by reducing the spectral dimension by employing some DR-method, such as PCA, and cropping spatial patches of $d \times d$ pixel-centered neighbors. For instance Chen et al. (2016) and Haut et al. (2019a) train a CNN2D with one principal component (PC), while Liang and Li (2016) employ three PCs to train the CNN2D and post-process the extracted spatial-features with sparse coding (SC) (Charles et al., 2011; Song et al., 2014) to create a sparse dictionary of more representative spatial features for classification. Xu et al. (2018) propose the random patches network (RPNet) as a CNN2D model where input data is whitened by PCA, taking into account only three PCs. Also, Zheng et al. (2017) perform an end-to-end classification with a CNN2D that receives

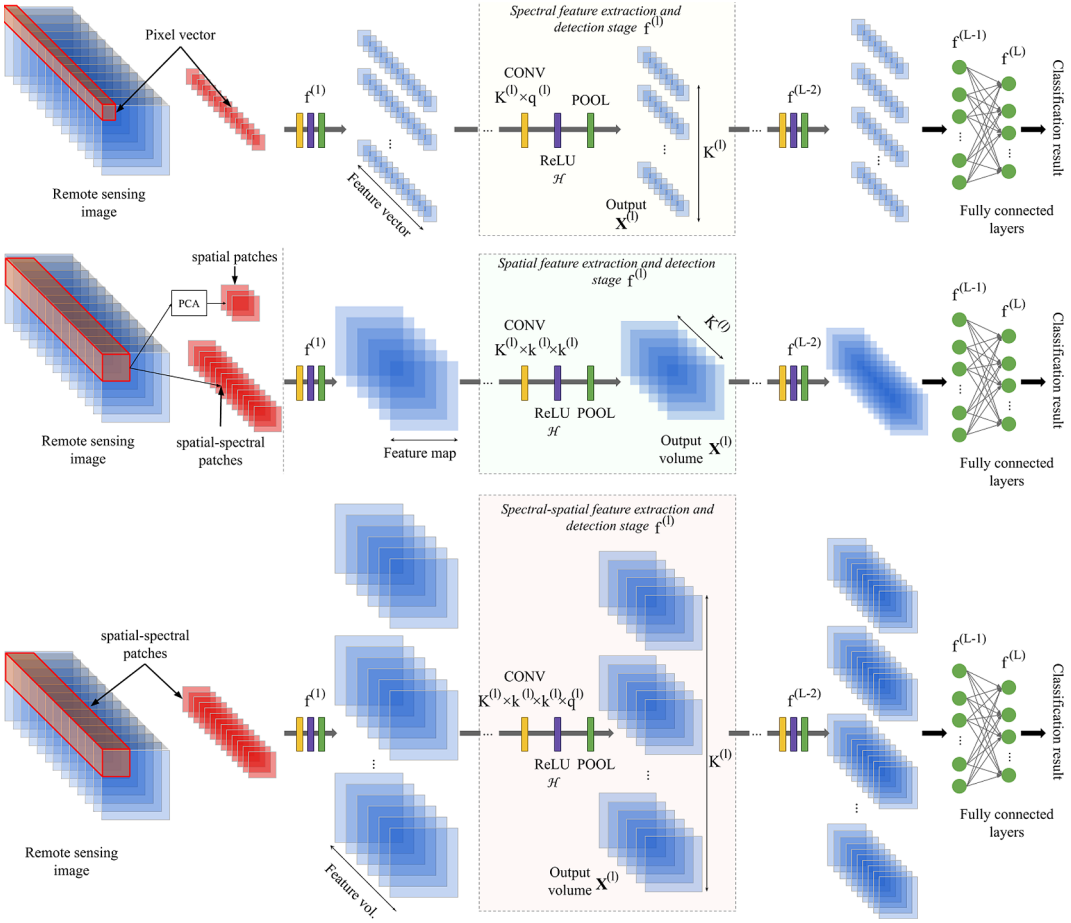


Fig. 7. Traditional architectures of spectral, spatial and spectral-spatial convolutional models employed by CNN1D, CNN2D and CNN3D architectures (top to bottom). The CNN1D architecture is commonly employed for spectral analysis, applying a hierarchical stack of L FE and detection stages, where each CONV layer exhibits kernels of $K^{(l)} \times q^{(l)}$. The CNN2D model can perform both spatial and spectral-spatial analysis by accepting spatial patches with few principal components or spectral-spatial patches with all (or most) available spectral bands, to which each CONV layer applies a kernel of $K^{(l)} \times k^{(l)} \times k^{(l)}$. Finally, the CNN3D model is employed for spectral-spatial analysis, taking full advantage of the spectral signatures contained in the input data by applying CONV layers with $K^{(l)} \times k^{(l)} \times k^{(l)} \times q^{(l)}$ kernels.

as input six PCs. Zhao et al. (2015) propose a a CNN2D architecture for extracting deep spatial features using, on the one hand, a multiscale convolutional AE based on the Laplacian pyramid and, on the other hand, the PCA to extract three PCs. Then, the extracted spatial features are concatenated together with the spectral information, using the logistic regression as a classifier. Furthermore, Ding et al. (2017) consider the HSI cube as a collection of 2-D images (i.e.images from different bands), which are cropped into patches to train a CNN2D model to automatically learn the data-adaptive kernels from the data through clustering.

In addition to introducing PCA-extracted spatial patches, some works propose the use of spatial-handcrafted features. For instance, Chen et al. (2017b) reduce the spectral domain to three PCs and extract spatial features (edges and textures) by applying Gabor Filtering. These features are sent to the CNN2D model, reducing the workload and addressing the overfitting problem. Another example is Romero et al. (2016), which performs a study between shallow and deep CNN2D models trained with the enforcing population and lifetime sparsity (EPLS) algorithm (Romero et al., 2015) for unsupervised learning of sparse multi/hyperspectral features.

Recently, a deformable HSI classification network model (DHCNet) has been proposed by Zhu et al. (2018a), using PCA to extract the three most informative PCs of the original HSI data cube and splitting the image into neighborhood windows to feed a CNN2D model, composed by deformable convolutions and downsampling that fuse the neighboring structural information of each input data sample in an adaptive manner.

4.4.3. Spectral-spatial CNN for HSI data analysis

Regarding spectral-spatial models, they consider both spectral properties and spatial information from the HSI data cube. In this sense, several strategies and architectures can be developed to perform the spectral-spatial processing, mainly due to the great flexibility that CNN models exhibit.

Following traditional pixel-wise methods, the CNN1D can be employed to perform spectral-spatial classification, rearranging the the spatial information and concatenating it to the spectral features (Zhang et al., 2016). For instance, Slavkovikj et al. (2015) integrate spatial and spectral information by reshaping the spectral-spatial neighborhood window to be processed by 1-D kernels, and Ran et al. (2017) improve the contextual information of the CNN1D by developing spatial PPFs (SPPFs), introducing the constraint that only the central pixel and its immediate surrounding pixels are paired.

Focusing on CNN2D architectures, these models can perform spectral-spatial processing in different ways. The most direct one is to feed the model with 3-D neighboring regions of size $d \times d \times n_{channels}$, where $n_{channels}$ can be certain number of PCs or the original n_{bands}. In this regard, some methods perform an initial DR in order to reduce the spectral correlation and redundancy. For instance, Makantasis et al. (2015) compose 3-D inputs with 10–13 PCs, applying the randomized PCA (R-PCA) over the HSI data cube. Yu et al. (2017) develop a spectral-spatial CNN with 1×1 CONV layers, also called cascaded cross-channel parametric pooling or CCCP layers (Lin et al., 2013a), and one global average pooling (GAP) layer instead of the traditional FC layers, to better analyze the HSI data information. Paoletti et al. (2017a)

develop a spectral-spatial model that efficiently takes into account the full spectrum, reaching competitive results, while Dong et al. (2019) propose a spectral-spatial CNN2D with a band-attention mechanism to improve the feature representation of the data.

The spectral-spatial processing can be performed by CNN2D architectures introducing spectral-spatial handcrafted features. For instance He et al. (2018) train the CNN2D model with covariance matrices, which encode the spectral-spatial information of different-sized neighborhoods of 20 PCs, obtaining multiscale covariance maps. Aptoula et al. (2016) use attribute profiles (APs) (Mura et al., 2010) as input to the CNN2D model, taking advantage of the spatial information and spectral properties that APs can capture in an image at various scales. Yue et al. (2015) develop a CNN2D architecture to process spectral and spatial features by composing the spectral information as three different feature maps, and concatenating them to the spatial patches (reduced by PCA to three PCs).

Moreover, several approaches combine the CNN2D with other different models to perform spatial and spectral feature extraction in separated fashion; for instance, Zhao and Du (2016) propose a spectral-spatial feature based classification (SSFC) approach that employs a CNN2D to find spatial-related features, while the spectral feature extraction is performed by a balanced local discriminant embedding algorithm (BLDE). Yue et al. (2016) extract spectral features from a SAE, while a multiscale spatial FE is performed by a CNN2D with spatial pyramid pooling (SPP). Zhang et al. (2017) and Yang et al. (2016) combine the hierarchical spectral and spatial-related features extracted from a CNN1D and CNN2D, respectively, performing the final classification with a softmax regression classifier. Ma et al. (2018b) introduce a two-branch model, where the spatial branch is composed by a CONV-DECONV architecture with skip connections, and the spectral branch is implemented by a contextual DNN.

In addition to the CNN1D and CNN2D models, the CNN3D model is usually adopted for spectral-spatial classification, where the 3-D filters of size $K^{(l)} \times k^{(l)} \times k^{(l)} \times q^{(l)}$ are able to extract high-level spectral-spatial features in a natural way, extracting as output $K^{(l)}$ feature volumes. For instance, Chen et al. (2016) review the three kinds of convolutional models that use the full pixel vectors in the original HSI data cubes to create the input blocks for their CNN3D model, and Li et al. (2017c) perform an interesting comparison between the spectral-spatial CNN3D model, two spectral-based methods (SAE and DBN), and the spatial CNN2D for HSI data classification, demonstrating that the CNN3D-based method is able to outperform these state-of-the-art methods. Furthermore, the CNN3D model can be used as a simple AE in order to obtain spectral-spatial features. For instance, Mei et al. (2019a) and Sellami et al. (2019) perform the classification on the spectral-spatial features obtained by a CNN3D model.

As with CNN1D and CNN2D models, the available literature offers more complex and sophisticated procedures for HSI data processing involving CNN3D architectures. For instance, Luo et al. (2018) develop a hybrid CNN2D-3D architecture able to deal with overfitting problems, using a 3-D kernel as the first layer of the network to extract feature vectors from the original 3D inputs, which are characterized by a small neighborhood window (only 3×3 with the full dimensionality given by n_{bands}). Then, the procedure reshapes the obtained feature vectors into one single matrix that is sent to the second 2-D kernel, and also to the subsequent pooling and FC layers, performing an end-to-end classification. With certain similarities, Leng et al. (2016) propose a cube-CNN-SVM (CCS) architecture which extracts several feature vectors from the original HSI data cube, performing the classification in an easy and efficient way with an SVM classifier. Roy et al. (2019) also combine a CNN3D with a CNN2D, where the CNN3D first extracts spectral-spatial features that are then refined by the CNN2D. Li et al. (2018b) follow a similar architecture, changing the final SVM by an RF. Moreover, Gao et al. (2018b) develop a CNN architecture with as many "branches" as AP features extracted from the HSI data cube, extracting independently the corresponding output volumes, which are concatenated and computed by the rest of the network. Cao et al. (2018) improve the performance of bayesian-inspired CNN3D model by placing spatial smoothness prior on data labels extracted with Markov random fields (MRFs) (Sun et al., 2015). Finally, Wang et al. (2019) introduce the alternately updated spectral-spatial CNN (AUSSC) as an end-to-end CNN3D with a recurrent feedback structure to learn refined spectral and spatial features.

4.4.4. Residual learning

CNN models have revolutionized the image processing field, establishing themselves as the current state-of-the-art. In this sense, the constant improvements in convolutional architectures, and their adoption in HSI data processing problems, have made possible to achieve performances never seen before in HSI classification (Khan et al., 2018). However, like the rest of DNN models, very deep CNN models must face some limitations related to the depth and the data degradation, as pointed out in Section 2.2. To overcome these issues, some works have focused on increasing the network's depth by creating short paths from low-level layers to high-level layers (i.e. residual connections). The development of convolutional architectures with residual learning has been a crucial step in the implementation of VDNN models, allowing the development of models with hundreds of layers.

The internal structure of residual neural networks (ResNets) (He et al., 2016) is based on groups of FE and detection stages which compose the basic building block, known as residual unit (Xie et al., 2017). The inputs of such block are directly connected to the outputs through an aggregation operation, as it can be observed in Fig. 8. Such residual connection performs an identity mapping that helps to propagate previous information to the subsequent units, improving the backward step by promoting the propagation of the gradient. In this regard, the output volume $\mathbf{X}^{(l)}$ of the l-th residual unit is given by Eq. (9), where $\mathcal{G}(\cdot)$ represents all the operations applied over the input data $\mathbf{X}^{(l-1)}$, which depend on all the parameters (weights $\mathcal{W}^{(l)}$ and biases $\mathcal{B}^{(l)}$) of the layers that compose the l-th unit:

$$\mathbf{X}^{(l)} = \mathcal{G}(\mathbf{X}^{(l-1)}, \mathcal{W}^{(l)}, \mathcal{B}^{(l)}) + \mathbf{X}^{(l-1)} \tag{9}$$

Eq. (9) reveals that the previous knowledge, in terms of generated features, is exploited once again in the current unit. The available literature gathers some works concerning the use of the ResNet in HSI processing. For instance, Zhong et al. (2017b) develop an end-to-end spectral-spatial ResNet (SSResNet) for HSI classification, outperforming traditional CNN models even with small training sets (Zhong et al., 2017c). Also, Paoletti et al. (2018c) present a pyramidal ResNet for spectral-spatial HSI data classification, improving the results of Zhong et al. (2017b). Lee et al. (2016) and Lee and Kwon (2017) propose the contextual deep CNN, which employs residual learning to simplify the training of the proposed network. Moreover, Mou et al. (2018) implement an unsupervised classification method based on the AE architecture with CONV-DECONV layers, following the ResNet architecture for spectral-spatial HSI data classification. In addition, Xie et al. (2018) and Yuan et al. (2019) employ the residual-based model as a spectral-spatial denosing AE for HSI data restoration and classification. Song

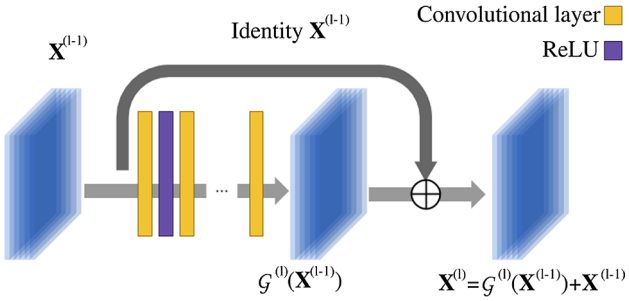


Fig. 8. Graphical visualization of a residual unit. The architecture reinforces the learning process of the model by reusing previous information.

et al. (2018) implement the deep feature fusion network (DFFN), composed by stages or branches of CONV layers connected with internal residual units, whose features are concatenated at the end (before the final classification). Li et al. (2019b) develop the multiscale deep middle-level feature fusion network (MMFN), an architecture that combines CONV layers and residual blocks into two stages to extract optimal multiscale features and to fuse and learn the complementary information from the obtained features. Chen et al. (2019a) present two DL ensemble methods based on CNNs and ResNets, implementing transfer learning to make full use of the learned weights. Moreover, recent works have focused on improving the performance of ResNets through attention techniques, for instance Haut et al. (2019) improve the spectral-spatial classification of the ResNet model including a visual attention mechanism to enhance the analysis of features. Mei et al. (2019b) develop a two-branch model, where the CNN's branch contains the spatial attention mechanism and the ResNet's branch implements the spectral attention mechanism. In addition to classification tasks, the ResNet has been also tested for HSI data super-resolution by Wang et al. (2017a), exhibiting good results.

The introduction of connections between different layers has inspired other models. Particularly, densely connected networks (DenseNets) (Huang et al., 2017) follow and extend the ResNet idea, reusing low-level, middle-level and high-level features by concatenating ($\frown$) all the previous feature maps obtained in a dense block (see Fig. 9). In this sense, the output of each dense block is calculated as the concatenation of the inner blocks that compose it:

$$\mathbf{X}^{(l)} = \mathcal{G}(\mathbf{X}^{(l-1)}, \mathcal{W}^{(l)}, \mathcal{B}^{(l)}) \frown \cdots \frown \mathcal{G}(\mathbf{X}^{(1)}, \mathcal{W}^{(1)}, \mathcal{B}^{(1)}) \tag{10}$$

In both cases, ResNets and DenseNets increase the number of connections, which does not imply a growth of model parameters that must be fine-tuned. Quite opposite, internal connections allow to reduce their number due to the presence of redundant information. At the same time, they reinforce the feature propagation along the network, performing a kind of regularization. Several works have adapted the DenseNet model to HSI processing tasks. For instance, Paoletti et al. (2018a) implement a Deep&Dense CNN model for spectral-spatial classification of HSI data, while Wang et al. (2018a) analyze spectral, spatial and spectral-spatial DenseNets for HSI classification. In a similar way to ResNet, the DenseNet can be combined with attention mechanims. For instance, Ma et al. (2019) propose the double-branch multi-attention mechanism network (DBMA) to separately extract spectral and spatial features, adopting (at each branch) an attention mechanism to extract the most discriminative features and Fang et al. (2019) propose an end-to-end 3-D DenseNet with spectral-wise attention mechanism for enhancing HSI classification. Also, the ResNet and DenseNet can be combined to construct a joint network, known as dual-path network (DPN) (Chen et al., 2017a), composed by bottleneck-blocks whose output is split into two branches: the first branch is element-wisely added to the residual path, and the second branch is concatenated with the densely connected path. This model has been successfully employed for HSI classification purposes; for instance, Kang et al. (2018) reach very good accuracy in comparison with ResNet and DenseNet, taking into account very small training sets (0.3%-0.5%) and using PCA to extract 5–10 PCs.

4.5. Other improved convolutional-based networks

In addition to ResNets and DenseNets, some other convolutional-inspired architectures have been developed for HSI data analysis. For instance, Liu et al. (2018) implement a siamese CNN (S-CNN) (Koch et al., 2015) for HSI data classification, which contains two branches of identical sub-networks that share the same configuration and parameters. This implies less parameters to fine-tune, requiring less training data and reducing the tendency to overfitting, helping to manage datasets with high intraclass variability and interclass similarity, reaching good performance with a small number of training samples.

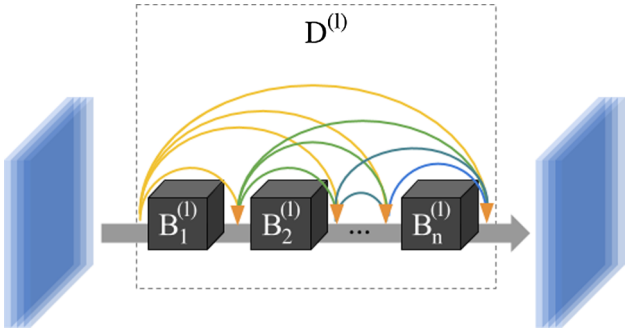


Fig. 9. Graphical visualization of a dense block. Instead of CONV layers, the DCNN is composed by dense blocks $D^{(l)}$, where each one contains several inner blocks $B_i^{(l)}$ composed by several CONV layers. The architecture of each $D^{(l)}$ allows for the reutilization of the low, middle and top feature maps extracted by each inner block.

Inspired by the inception architecture (Szegedy et al., 2015; Szegedy et al., 2016), the work of Lee et al. (2016) introduces an inception module at the beginning of their model, composed by n-parallel streams with several layers and different kernel sizes, whose outputs are merged by concatenation. Moreover, inspired by the network-in-network (NiN) (Lin et al., 2013a) architecture, Shamsolmoali et al. (2018) train a RNN with combined spectral-spatial features extracted by a CNNiN.

Based on the fully convolutional network (FCN) (Long et al., 2015a), whose learnable layers rely only on CONV and DECONV layers, Li et al. (2018a) implement an AE-based FCN for HSI-FE, using an ELM to classify the obtained features. Following the CONV-DECONV architecture and adding skip connections, the hourglass CNN architecture (Newell et al., 2016; Haut et al., 2018b) creates an encoder-decoder structure where each block of CONV layers that compose the encoder is connected to the corresponding DECONV layer at the decoder counterpart. This architecture can be employed for HSI data denoising (Sidorov and Hardeberg, 2019).

Recently, a new kind of network based on capsules and dynamic routing has been implemented, called Capsule Networks (CapNets) (Sabour et al., 2017). This architecture encodes the data internal relationships into an activity vector (instead of a traditional scalar value). Such data representation has demonstrated to be powerful in encoding useful features from the data, solving the limitations exhibited by the pooling layer. In this sense, Paoletti et al. (2018b) present a spectral-spatial CapsNet for HSI classification that outperforms the accuracies reached by traditional CNN models and the ResNet. Also, Deng et al. (2018) present a HSI-CapsNet that provides good results when very few training samples are employed.

5. Overcoming the limitations of DL in HSI Classification

The vast number of works discussing DNN models (in general) and CNNs (in particular) for HSI data analysis reveals the great possibilities that DL-based methods are able to offer in this context, not only in terms of architectural modifications, but also regarding their combination with other methods and algorithms, as we pointed out on Section 3. This also includes a large variety of remote sensing image processing techniques apart from data classification (FE, DR, unmixing, reconstruction, super-resolution, etc.) Convolutional-based networks such as CNNs and ResNets represent the most groundbreaking advance in DL in the last few years, allowing the implementation of VDNN models with hundreds/thousands of layers and compelling performance, following the assumption that deeper models are able to extract more complex and high-level relationships from the data (Srivastava et al., 2015). In the end, this expected to lead to improvements in model accuracy and performance (Krizhevsky, 2012; Yu et al., 2013). This has

also placed CNNs and ResNets as the current mainstream technologies in DL for HSI classification. However, these models must also face the limitations listed on Section 2, related to intrinsic problems of HSI analysis and the efficient management of depth. In order to deal with the aforementioned issues, several techniques and mechanisms have been developed in previous years to enhance the learning process and improve the performance of deep architectures. In the following, we provide a description of the available strategies to mitigate these issues.

5.1. Opening the black box

Regarding the "black box" nature of DNN models (in general) and convolutional-based ones (in particular), several efforts have been made to "open" the box and understand what are the filters actually doing (Rauber et al., 2017). For instance, *mNeuron* (Wei et al., 2017a) is a powerful Matlab plugin that allows the visualization of convolutional neural network parameters, while t-distributed stochastic neighbor embedding (t-SNE) (Maaten and Hinton, 2008) and uniform manifold approximation and projection (UMAP) (McInnes et al., 2018) are non-linear dimensionality reduction techniques which are also employed to visualize the models parameters in a simple way. Liu et al. (2017b) propose to formulate the CNN model as a directed acyclic graph (DAG), developing the CNNVis as a visual analysis system to better understand, diagnose and refine CNN models.

Regarding parameter visualization, several works propose to understand how DL is working in step-by-step fashion. In this context, Lei et al. (2018) present an ambitious dissertation, analyzing the DL-based models as physical systems from a microscopic, macroscopic, and worldview perspectives. Ravanelli and Bengio (2018) present a more concrete proposal, developing the SincNet, a convolutional-based model that exploits parametrized *sinc* functions in the first layer to discover more significant filters. Also, the BagNet (Brendel and Bethge, 2019) employs a visual bag-of-local-features model to perform the classification, extracting features that are easy to identify and interpret.

There is a wide variety of proposals to understand what networks do (Mahendran and Vedaldi, 2015; Nguyen et al., 2015). However, in the current literature about HSI-classification, little attention has been paid to this issue. Qiu and Jensen (2004) propose a method for understanding the performance of a three-layer MLP in HSI classification, but no relevant efforts have been reported with DL-architectures.

5.2. Reducing overfitting

In order to address the overfitting problem in convolutional-based models, several strategies have been reported that can be classified into four main categories: (i) those that affect the data, (ii) those that affect the model, (iii) those that affect the training process, and (iv) new learning paradigms to deal with the limited availability training data.

5.2.1. Data augmenting and noise inclusion

Gathering enough labeled samples to capture the high variability of HSI data is complicated, time consuming and expensive. Several works have focused on addressing this issue through the generation of virtual samples to enhance the robustness of convolutional-based models (Acquarelli et al., 2018). Following traditional methods, Yu et al. (2017) enlarge the training set by rotating and flipping the input spectral-spatial patches, while Lee and Kwon (2017) mirror the spectral-spatial training patches four times, across the horizontal, vertical and diagonal axes. On the other hand, Haut et al. (2019a) implement a mechanism to add spatial-structured noise to the input HSI data by randomly occluding some areas of the input patch in order to enhance the performance and robustness of the CNN model. Chen et al. (2016) present two methods to create additional training samples: the first one changes the spectral radiation of the original training samples $\mathbf{x}_i$ by multiplying them by a random factor and adding random noise, and the second one by mixing the spectral properties of two samples of the same class with proper ratios. Also, Acquarelli et al. (2018) present two methods: smoothing-based data augmentation, which takes advantage of the spectra of neighboring pixels, and label-based data augmentation, which exploits the labels of neighboring pixels to favor those classes with less samples, in addition to creating copies of the original data by inserting random noise. Ghamisi et al. (2016) propose a *dither* algorithm (Simpson, 2015) to suppress non-linear distortions and data aliasing, generating new samples by adding random noise to the original training samples, in addition to using the fractional order Darwinian particle swarm optimization (FODPSO) to select the most informative spectral bands. An interesting work has been recently proposed by He and Chen (2019), who implement a transformation network (STN) to obtain an optimal input of the CNN model for HSI classification, which translates, rotates and scales the network's input until obtaining an optimized one.

5.2.2. Reducing the complexity of the model

The second strategy to reduce overfitting is focused on reducing the computational complexity of deep CNNs (Maji and Mullins, 2018), for instance, by optimizing the internal structures of the CONV layers, pruning them to obtain a more simple and efficient network architecture (Cheng et al., 2017b). The thinning of the network (and the subsequent parameter reduction) make the CNN model lighter, which leads to faster training and execution, although not many efforts have been made in this direction in the HSI arena. Recently, works focused on designing optimal CNN architectures for HSI data processing have been presented. Specifically, Chen et al. (2019b) propose a methodology to automatically design efficient CNN1D and CNN3D architectures for HSI data classification. Given a number of operations (i.e., layers such as CONV, POOL or normalization), a gradient descent-based search algorithm evaluates all possible configurations and selects the best and optimal one.

5.2.3. Enhancing the training process

The methods for this purpose cover a wide range of techniques. For instance, L1/L2 regularization methods insert a penalty into the loss function in order to minimize the absolute value of the weights or the squared magnitude of the weights (weight decay or Tikhonov regularization), respectively (Murugan and Durairaj, 2017). The regularization process forces the model to make compromises on its weights, making it more general. In particular, the L1 regularization enforces the identification of the most relevant features in a dataset, while the L2 pursues a regularization that is less aggressive, but more efficient in computational terms. For instance, Chen et al. (2016) use the L2 regularization.

In addition to these methods, dropout regularization (Hinton et al., 2012; Srivastava et al., 2014) has also been proven to be a good solution to enhance the performance and robustness of the CNN model, preventing complex co-adaptations on training data. The mechanism is quite simple: it randomly deactivates a percentage of the activations in order to improve the network generalization, forcing the neurons to make more compromised assumptions. For instance, Paoletti et al. (2017a) make use of dropout in the layers of the CNN. Based on dropout, multiple regularization techniques have been developed (Ba et al., 2013; Zhang et al., 2016a; Molchanov et al., 2017) such as its generalization to large FC layers: the drop-connect (Wan et al., 2013), which sets randomly selected connection weights to zero. However, traditional dropout injects random single-pixel noise to the feature maps, resulting in spatially unstructured noise, which makes it ineffective in 2-D and 3-D models (Park and Kwak, 2017). In this sense, spatial-dropout (Tompson et al., 2015) and dropblock (Ghiasi et al., 2018) regularization techniques overcome the problem by dropping spatial-regions of the feature maps.

Another interesting regularization technique for preventing overfitting is early stopping (Caruana et al., 2001), which saves at each epoch those models that outperform the previous trained networks, discarding the others and storing at the end the results of the best

model. For instance Ran et al. (2017), Acquarelli et al. (2018), and Wang et al. (2018a) employ this technique to assess the convergence of their convolutional-based models.

5.2.4. Improvements on learning strategies

Overfitting in DNN models is intimately related to the number of samples available for training, representing one of the main limitations of supervised and very deep models. In this context, some improvements on learning paradigms have been developed in the DL field that can effectively improve the performance of DNNs when very few samples are available. We describe four of such paradigms: (i) semi-supervised and (ii) active learning (AL), (iii) transfer learning (TL), and (iv) self-supervised learning.

Semi-supervised learning In-between unsupervised and supervised learning, DNN models allow the implementation of hybrid approaches. In particular, semi-supervised learning (Ratle et al., 2010) provides a wide range of techniques to expand the training set $\mathcal{D}_{train}$ by including unlabeled data during the training stage (Ratle et al., 2010; Sabalel and Jadhav, 2014). For instance Ma et al. (2016c) present a semi-supervised learning strategy based on multi-decision labeling (local, global and self-decision levels), where unlabeled samples with high confidence are selected to extent the training set. Kang et al. (2019) extract pseudo-training samples from PCA and extended morphological attribute profiles (EMAPs) (Dalla Mura et al., 2011), applying extended random walker optimizers to feed a spectral-spatial convolutional-based deep feature fusion network (DFFN). Wu and Prasad (2018) present a convolutional-based recurrent model fed by pseudo training samples obtained by previous clustering. Fang et al. (2018) adopt separated spectral and spatial residual architectures with co-training, where the most confident labeled samples at each iteration are included in $\mathcal{D}_{train}$. A similar approach has been implemented by Zhou et al. (2019b), in which two separated spatial and spectral SAEs are co-trained, enlarging $\mathcal{D}_{train}$ by a region growing method. An interesting trend is the adoption of the ladder network (Rasmus et al., 2015; Pezeshki et al., 2016), a new DNN model based on hierarchichal latent variable models, for semi-supervised classification of HSI data (Liu et al., 2017a; Büchel and Ersoy, 2018).

In addition to the addition of unlabeled data to $\mathcal{D}_{train}$, some semi-supervised techniques are able to replicate new samples. In particular, the DNN structure known as generative adversarial network (GAN) (Goodfellow et al., 2014). For instance, Zhu et al. (2018b) propose convolutional-based GAN1D and GAN3D architectures to learn the intrinsic characteristics of HSI data, enhancing the classification performance achieved by the traditional CNN1D and CNN3D. Also He et al. (2017b), Zhu et al. (2018b), and Zhan et al. (2018) present similar approaches, while Zhang et al. (2018a) introduce a Wasserstein GAN to perform unsupervised FE.

Active learning (AL) AL is a semi-supervised machine learning algorithm (MacKay, 1992) that can easily deal with the availability of a limited amount of labeled data by training the model with a small set of labeled samples that is reinforced by the acquisition of new (representative and intelligently selected) unlabeled samples, reducing the cost of acquiring large labeled training sets and the number of needed training samples. In the literature, several works combining the DNN and AL paradigms can be found. For instance, Haut et al. (2018c) discuss the use of Bayesian CNNs for spectral, spatial and spectral-spatial HSI classification, providing robust classification results in comparison with traditional CNN models. In addition to convolutional models, Liu et al. (2017c) employ AL with a DBN, while Li (2015) develops an AL-based SAE.

Transfer learning (TL) The TL paradigm (Yosinski et al., 2014; Long et al., 2015b) is based on the assumption that learned features in one task can be used for other tasks (Pan and Yang, 2010). Low-level layers in convolutional-based architectures are able to learn generic features that are less dependent on the final task, while the top-level layers learn more specific knowledge, extracting features that are more related with the final task. In this sense, TL-based algorithms usually employ off-the-shelf pre-trained networks (i.e. models that were trained on different datasets) to process the data of interest, tailoring them slightly for the new task by removing some the last few layers of the model and retraining again with some new final layers. As a result, the amount of data used to pre-train the CNN can be leveraged, alleviating the need for new data (this is useful when limited amounts of training sets are available) and producing better results in a shorter amount of time (Windrim et al., 2018). The most widely used off-the-shelf pre-trained networks are trained with the ImageNet dataset, composed by 14 million images belonging to 1000 different classes. The most popular topologies are the following ones:

- *ResNet-50*, composed by 50 residual layers,
- *DenseNet121* (Huang et al., 2017), composed by 4 dense blocks connected by transition layers,
- *VGG-16* and *VGG-19* (Simonyan and Zisserman, 2014), which increase the depth by using many layers: 16 and 19, respectively, using a simple architecture with small kernels (CONV layers of 3×3) and reducing the volume size (through POOL layers of 2×2),
- *MobileNet* (Howard et al., 2017), whose architecture is suitable for onboard processing, maximizing the accuracy while taking into account restricted resources for an integrated application,
- *Xception* (Chollet, 2017), based on inception networks, where the original modules have been replaced with depthwise separable convolutions in order to make a more efficient use of model parameters.

Several works adapt the TL paradigm to process HSI data (Mei et al., 2017; Jiao et al., 2017; Windrim et al., 2018; Yang et al., 2017; Deng et al., 2019; Zhang et al., 2019), visibly improving the training of deep CNN models when limited amounts of labeled data are available for the training stage.

Self-supervised learning This learning strategy emerges as an alternative approach to supervised learning, being able to extract the naturally available contextual information and embedded metadata as supervisory signals, without an explicit need for $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{n_{labelled}}$ pairs. This does not mean learning the inherent structure of data in the form of unsupervised learning (Liu et al., 2019; Jing and Tian, 2019). In HSI data classification, Wang et al. (2018b) propose the HSINet, which contains a three-layer DNN, a multi-feature CNN, and an embedded conditional random field to achieve self-supervised feature learning, extracting spatial, spectral, color, boundary and contextual information. Also, Liang et al. (2018) combine TL with self-supervised learning, developing a pre-trained VGG-16 to extract deep multi-scale spatial information from the HSI data cube, whose spatial information is processed together with the raw spectral information by a SAE.

5.3. Vanishing gradient problem

Apart from improvements based on architectures, such as ResNets and CapsNets, those methods employed to deal with the vanishing gradient problem (Srivastava et al., 2015) can be categorized into three main groups: (i) implementing data normalization between each network layer, (ii) developing better initialization strategies with proper optimizers, and (iii) implementing better non-linear activation functions.

5.3.1. Avoiding vanishing gradient through data normalization

During gradient descent training, the layer's weights $\mathbf{W}^{(l)}$ and the obtained data $\mathbf{X}^{(l)}$ distributions can vary (covariate shift effect), making the learning very unstable and saturating the activations whose first derivative tends to zero. This leads to the vanishing gradient problem. In this sense, it is common to employ normalization methods to control the magnitude and mean of the neurons' activations located into one layer (independently of the other layers of the model). This aims at

performing the parameters' optimization in an easier way (Santurkar et al., 2018) while, at the same time, dealing with the unbounded nature of certain activation functions (for instance, the ReLU), whose outputs are not constrained within a bounded range (such as the tanh function). Table 2 provides a summary of several relevant normalization methods that have been adopted in this context.

5.3.2. Avoiding vanishing gradient through initialization and optimization strategies

Classical DNNs initialize their parameters, setting small random values to the weights and biases that compose the model, under the assumption that this helps the stochastic optimization algorithm used to train the model. In this sense, the selection of the optimization algorithm becomes fundamental in order to obtain a proper performance of the model. This selection must take into account the type of data to be used, the task to be performed, and the features of the problem.

Several optimization methods with different strengths and weaknesses have been developed with the aim of improving the process of minimizing an objective function: the traditional stochastic gradient descent (SGD), which is faster than standard gradient descent (GD) but harder to converge to the minimum due to frequent updates and fluctuations; the minibatch SGD, which reduces the high variance in the parameter updates; the momentum, which speeds SGD by descending along the relevant direction, reducing oscillations; the preconditioned SGD (PSGD) (Li, 2018), which adaptively estimates a preconditioner for handling efficiently the gradient noise and non-convexity of a target function at the same time, giving good results in deep neural models optimization (Li et al., 2016); Adagrad (Duchi et al., 2011), a variant of PSGD which adapts the learning rate based on the parameters, being well-suited for dealing with sparse data although its learning rate can suffer from constant decaying, producing the *decaying learning rate* problem and hampering the optimization process; AdaDelta (Zeiler, 2012), which tries to avoid the decaying learning rate problem by calculating different learning rates for each parameter and is often combined with the momentum technique; and finally the adaptive moment estimation (Adam) (Kingma and Ba, 2014), which is a combination of Adagrad and AdaDelta, outperforming the previous optimization techniques. It is based on processing adaptive learning rates for each parameter and storing several past gradients to keep a decaying average of those past gradients, which makes it efficient, with fast convergence and effective when dealing the vanishing learning rate. The excellent results of Adam algorithm have positioned it as the method that is most widely used for optimizing deep networks, being employed in some HSI-related works as Paoletti et al. (2017a, 2018c).

In addition to the improvements implemented on the optimizers (Martens et al., 2012; Sutskever et al., 2013; Dauphin et al., 2014), recently new investigations have been made in order to improve the initialization of model parameters (Bengio et al., 2007a; Glorot and Bengio, 2010; Erhan et al., 2010; He et al., 2015; Koturwar and Merchant, 2017; Guo and Zhu, 2018), for instance by performing unsupervised pre-training (Romero et al., 2016; Li et al., 2015a), which initializes the parameters near to a local minimum, allowing for a better generalization via unsupervised FE.

5.3.3. Avoiding vanishing gradient through new non-linear activation techniques

Currently, several efforts for preventing the vanishing-gradient problem have been made based on developing effective non-linear activation functions $\mathcal{H}(\cdot)$ (Xu et al., 2015a; Pedamonti, 2018). In particular, some rectified-based activation functions have been adapted to overcome the problem by preventing the gradient from being zero. For instance, in order to face the dying ReLU problem the *leaky* ReLU (LReLU) (Maas et al., 2013) and *parametric* ReLU (PReLU) (He et al., 2015) functions have been implemented with Eq. (11) (see Fig. 3).

$$\mathcal{H}(x) = \begin{cases} a \cdot x & \text{if } x \leq 0 \\ x & \text{if } x > 0 \end{cases} \tag{11}$$

In particular, the LReLU sets the gradient signal as a linear component of the input layer data $\mathbf{X}^{(l)}$, employing a small and constant negative slope (usually $a = 0.001$) when the data is equal or smaller than 0. This avoids the *dying* ReLU problem, as the function will not have zero-slope parts, making the LReLU more balanced and allowing a faster learning. PReLU works similarly, being a learnable in this case. In this context, the vanishing gradient depends on the slope a. Instead of that, the *scaled exponential linear unit* (SeLU) (Klambauer et al., 2017; Paoletti et al., 2018) derives two parameters: α and λ from the inputs, as we can observe in Eq. (12), allowing $-\lambda\alpha$ as the smallest gradient value and mapping the means and variances from one layer to the next one in order to minimize the covariate shift effect.

$$\mathcal{H}(x) = \lambda \begin{cases} \alpha e^{x} - \alpha & \text{if } x \leq 0 \\ x & \text{if } x > 0 \end{cases} \tag{12}$$

6. Popular deep learning frameworks

The current trend in the literature is to implement deeper and more complex networks, with new topologies, more branches and connections, and better optimizers and functions. In this sense, certain programming frameworks have been deployed in order to provide technical and coding support to developers of DL methods. In particular, these DL frameworks offer a black-box environment for training and validating DNN models through a high level programming interface. Furthermore, instead of *ad hoc* software, the framework provides quality and maintainability of applications at low cost, allowing for the model to better adapt to market standards. In terms of performance, available frameworks are able to easily exploit computing tools, developed and supported by large communities, relying on well-known high performance computing (HPC) libraries such as CUDA, CUDNN, MKL, BLAS, AVX operations and Protobuf, among others.

Table 3 provides a summary of the main DL frameworks currently available in public repositories, including a brief description, the programming language that was used for coding purposes, and the available application interfaces (APIs). It is interesting to highlight the use of Python as one of the main programming languages in the community to implement DL frameworks, due to its versatility and flexibility. In addition, the number of stars and forks have been provided as indirect evaluation metrics of those repositories (see Fig. 10), where stars measure the degree of popularity of the repository, while forks measure the number of copies that have been made of the original repository. These data has been obtained on two different dates: July 16th, 2018 and September 8th, 2019, in order to compare the evolution of these indicators. It can be observed that the most popular DL framework is TensorFlow (Abadi et al., 2016a; Abadi et al., 2016b), tracked by more than 120.000 followers and with more than 70.000 branches and forks, being the framework that has grown the most from 2018 to 2019. Concerning TensorFlow, the high-level library Keras (Ketkar, 2017) has also experimented an increase in the number of followers, allowing for the development of ANN models in an easy and simple way. Also, it is interesting to note that the frameworks based on Torch (Collobert et al., 2002), Pytorch (Fey and Lenssen, 2019) and Fast.AI have also significantly grown, providing an easy-to-debug tool for the implementation of neural models.

7. Experimental results

7.1. Hyperspectral datasets

After reviewing the main models and frameworks, we perform a comparison between the most popular DL-based architectures and traditional ML-based algorithms in order to quantify the improvements

Table 2
Some examples of normalization methods for neural networks.

Method	Description
Local response normalization (LRN) (Krizhevsky et al., 2012)	It was introduced by the first time in the AlexNet model to enhance the *lateral inhibition* property of the neurons, i.e. the ability of the neural nodes to reduce the activity of its neighbors by competition, modulating the feedback signals and enhancing the visual contrast (i.e. performing an attention mechanism) (Wyatte et al., 2012). In this sense, the LRN allows to diminish responses that are uniformly large for the neighborhood, making large activation more pronounced within a neighborhood and creating higher contrast in activation maps in order to increase the sensory perception. It has been successfully applied by (Lee et al., 2016; Lee and Kwon, 2017).
Batch normalization (BN) (Ioffe and Szegedy, 2015)	Considering the output volume $\mathbf{X}^{(l)}$ of the l-th layer as the data to normalize, with n_{batch} data representations, where each one comprises $K^{(l)}$ feature maps of size $d^{(l)} \times d^{(l)} \times n^{(l)}_{channels}$, being n_{batch} the batch size, $d^{(l)} \times d^{(l)}$ the spatial dimensions and $n^{(l)}_{channels}$ the number of spectral bands, the BN method normalizes the obtained features by computing the mean μ and variance σ^2 respect to the feature maps (channel) dimension. It has been widely used by the HSI community (Liu et al., 2017a; Zhong et al., 2017c; Gao et al., 2018a; Deng et al., 2018), being usually applied before the activation function, to maintain the data distribution to zero-mean and unit variance, scaling and shifting the data through the learnable parameters γ and β, respectively. This allows to reach a more independent and high-speed learning (with larger learning rates and high accuracy), although it is very sensitive to the batch size n_{batch} (Bjorck et al., 2018).
Weight normalization (WN) (Salimans and Kingma, 2016)	It normalizes the weights of the l-th layer, reparameterizing $W^{(l)}$ in terms of a parameter vector $\mathbf{v}$ (weights' direction $\mathbf{v}/\|\mathbf{v}\|$) and a scalar parameter g (weights' norm, which is also obtained by a learnable log-scale parameter s as $g = e^s$) and directly performing the backpropagation with respect to those parameters instead, in order to fix the Euclidean norm of $\mathbf{W}^{(l)}$. This, coupled with the mean-only batch normalization, allows the scale of neural activations to be approximately independent of the parameter $\mathbf{v}$, as well as their mean (Gitman and Ginsburg, 2017).
Layer normalization (LN) (Ba et al., 2016)	Similar to BN, LN normalizes the data computing the mean $\mu_{\mathcal{B}}$ and variance $\sigma^2_{\mathcal{B}}$ with respect to the batch dimension, in order to avoid the limitations of the BN method. In this sense, LN does not employ batch statistics, being the normalization of each sample independent of other samples. This enables a beneficial behaviour in networks such as RNNs.
Instance normalization (IN) (Ulyanov et al., 2016b)	Inspired by Huang and Belongie (2017) in neural style transfer tasks Ulyanov et al. (2016a), the IN normalizes across each feature map dimension in the batch independently avoiding the dependency of the batch and normalizing the constrast of the content image. It is commonly used to remove variance of images on low-level vision tasks (Pan et al., 2018). Also, coupled with BN, IN has inspired the development of other methods, such as batch-instance normalization (BIN) (Nam et al., 2018) that extends the handling of the variability introduced by visual styles (textures, lighting, filters) to general recognition problems.
Group normalization (GN) (Wu and He, 2018)	GN divides the feature map (channel) dimension into several groups, normalizing each group in the current batch, exhibiting a behavior that straddles the layer and instance normalization methods depending on the number of groups that it creates.
Batch re-normalization (BRN) (Ioffe, 2017)	It extends the BN method in order to deal with small or non-independent and identically distributed (non-i.i.d) batches, normalizing the activations through the combination of the batch's mean and variance ($\mu_{\mathcal{B}}$ and $\sigma^2_{\mathcal{B}}$, respectively) and the moving averages (μ and σ^2) in an affine transformation.
Decorrelated batch normalization (DeBN) (Huangi et al., 2018)	BN is able to scale and shift the obtained activations through parameters γ and β. In this sense, the DeBN extends the BN method to perform data whitening, taking into account the zero-phase component analysis (ZCA) method (Kessy et al., 2018) to decorrelate the neural activations.

Table 3
Some of the most widely used DL-based frameworks (data obtained on September 8th, 2019).

NAME	DESCRIPTION	APIs	STARS	FORKS
Tensorflow	An Open Source Machine Learning Framework for Everyone	C++, Go, Java, JavaScript, Python, Swift	133322	77067
Keras	Deep Learning for Humans	Python	43786	16673
OpenCV	Open Source Computer Vision Library	C++, Java, Python	37798	27983
PyTorch	Tensors and Dynamic Neural Networks in Python with Strong GPU Acceleration	C++, Python	31416	7712
Caffe	A fast open framework for deep learning	CLI, Matlab, Python	29000	17530
MXNet	Lightweight, Portable, Flexible Distributed/Mobile Deep Learning with Dynamic, Mutation-aware Dataflow Deep Scheduler	C++, Clojure, Java, Julia, Perl, Python, R, Scala	17647	6276
CNTK	Microsoft cognitive toolkit (CNTK), an Open Source Deep-learning Toolkit	C++, C#, Python	16394	4365
Fast.AI	The Fast.ai Deep Learning Library, plus Lessons and Tutorials	Python	15384	5517
Deeplearning4j	Eclipse Deeplearning4j, ND4J, DataVec and more - deep learning & linear algebra for Java/Scala with GPUs + Spark	Java/Scala	11130	4735
Paddle	PArallel Distributed Deep LEarning	Python	9851	2628
ConvNetJS	Deep Learning in Javascript. Train CNNs (or ordinary ones) in your browser.	Javascript	9787	1951
Theano	Python library that allows to define, optimize, and evaluate efficiently mathematical expressions involving multi-dimensional arrays	Python	8900	2504
Horovod	Distributed training framework for TensorFlow, Keras, PyTorch, and Apache MXNet	Python	7380	1130
Chainer	A flexible Framework of Neural Networks for Deep Learning	Python	5028	1327
BigDL	BigDL: Distributed Deep Learning Library for Apache Spark	Python/Scala	3152	797
MatConvNet	CNNs for MATLAB	MATLAB	1205	713

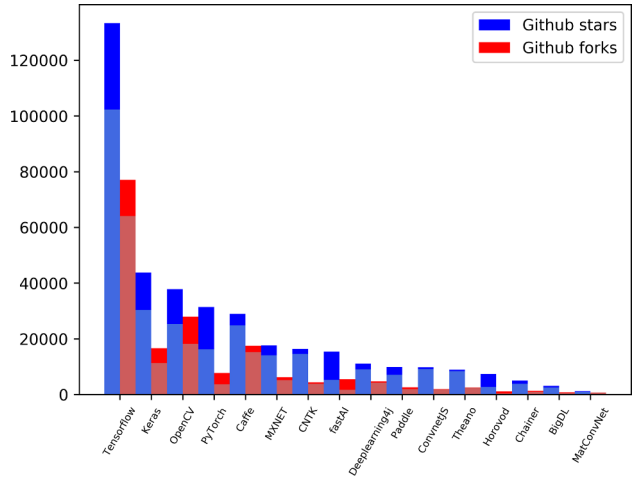


Fig. 10. Stars and forks of the most representative DL framework repositories. The light blue and red bars refer to the number of stars and forks measured on July 16th, 2018, while the dark blue and red bars correspond to the number of stars and forks measured on September 8th, 2019 (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.).

and advantages that can be gained by DL models in terms of performance and classification accuracy. To this end, four images widely used in the field of hyperspectral image processing have been selected to complete the experimental part of the work: the Indian Pines (IP) and Salinas Valley (SV) scenes, collected by AVIRIS, the University of Pavia (UP) scene, gathered by ROSIS and the University of Houston (UH) scene, collected by CASI. Table 4 shows a brief summary of these HSI datasets, including the number of labeled samples per class, as well as the available ground-truth information:

- The IP dataset (Table 4) was captured in 1992 by the AVIRIS sensor (Green et al., 1998) over the Indian Pines test site in NW Indiana, an agricultural area characterized by its crops of regular geometry and also irregular forest regions. The scene consists of 145×145 pixels with a spatial resolution of 20 mpp and with 224 spectral bands, which have been collected in the wavelength range from 0.4 to 2.5 μm. From these bands, 24 were removed for being null or water absorption bands (in particular [104–108], [150–163] and 220), considering the remaining 200 bands for the experiments. The ground truth available is divided into sixteen classes and about half of the data (10249 pixels from a total of 21025) contains labeled samples.
- The UP scene (Table 4) was acquired by the ROSIS sensor (Kunkel et al., 1988) over the campus of the University of Pavia, in the north of Italy. The dataset contains nine different classes that belong to an urban environment with multiple solid structures, natural objects and shadows. After discarding the noisy bands, the considered scene contains 103 spectral bands, with a size of 610×340 pixels with spatial resolution of 1.3 mpp and covering the spectral range from 0.43 to 0.86 μm. Finally, about 20% of the pixels (42776 of 207400) contain ground-truth information.
- The SV image (Table 4) was gathered by the 224-band AVIRIS sensor over several agricultural fields of Salinas Valley, California, and it is characterized by a spatial resolution of 3.7 mpp. The area covered comprises 512×217 spectral samples. As in the case of the IP dataset, we discard 20 bands due to water absorption and noise.
- The UH scene (Xu et al., 2016a) was collected by CASI in June 2012 over the University of Houston campus and the neighboring urban area. This scene forms a cube of dimension $349 \times 1905 \times 144$, with spatial resolution of 2.5 m and spectral information captured in the range from 0.38 to 1.05 μm, containing 15 ground-truth classes divided in two categories: training (top UH map in Table 4) and and testing (bottom UH map in Table 4). In this sense, the UH scene provides an interesting benchmark dataset, which was first presented at the IEEE Geoscience and Remote Sensing Society (GRSS) Image Analysis and Data Fusion Technical Committee during the 2013 Data Fusion Contest (DFC) (Debes et al., 2014).

These datasets, along with the training and test data, are all available online from the GRSS Data and Algorithm Standard Evaluation (DASE) website (http://dase.grss-ieee.org).

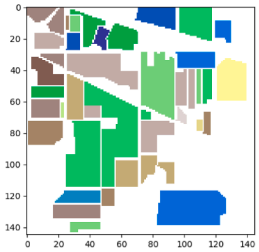
7.2. Experimental settings

In order to make an exhaustive analysis of the main DL-based architectures employed for HSI classification purposes, an extensive set of experiments have been carried out.

1. The first experiment compares the performance of supervised standard ML and DL classification methods with different amounts of training samples over the four considered HSI datasets, studying how they are affected by the lack of information and the type of samples. In particular 1%, 5%, 10%, 15%, 20% and 25% of the labeled samples per class have been randomly selected to compose the training set on IP, UP and SV, while the full available training set for UH has been considered. Also, some of the most popular classification algorithms available in the literature have been considered: (1) random forest (RF), (2) multinomial logistic regression (MLR), (3) support vector machine (SVM) with radial basis function kernel (Waske et al., 2010), (4) multilayer perceptron (MLP), (5) vanilla recurrent neural network (RNN), (6) RNN with gated recurrent unit (GRU), (7) RNN with long short term memory (LSTM), (8) spectral CNN (CNN1D), (9) spatial CNN with 2-D kernels and one PC (CNN2D), (10) spectral-spatial CNN with 2-D kernels and forty PCs (CNN2D40), and (11) spectral-spatial CNN with 3-D kernels and also forty PCs (CNN3D). Regarding the configuration of the experiment, the available training set has been divided into batches of 100 samples, using Adam optimizer with learning rate of 0.0008 for SV and UP and 0.001 for IP and UH. Regarding the number of epochs, MLP, CNN1D, CNN2D and CNN2D40 have been trained using 300 epochs. The parameters of RNN, GRU and LSTM models have been adjusted using 200 epochs. Finally, the parameters of CNN3D have been trained using 100 epochs. Furthermore, the topology details of each model are reported on Table 5. It must be noted that we follow the convention that deep architectures have at least two or more hidden layers, while shallow models are composed by single-hidden layer architectures (Bengio et al., 2007b; Schmidhuber, 2015). In addition, the inclusion of batch normalization (BN) in some layers of the convolutional models intends to, on the one hand, avoid vanishing/exploding gradients and, on the other hand, maintain the distribution of the layer's inputs (internal covariate shift) (Ioffe and Szegedy, 2015). We have empirically observed that, on some (but not all) CNN models, BN stabilizes and accelerates the training stage. In this sense, we noted that these configurations helped these particular convolutional models. Furthermore, we also empirically observed that a filter size of 5×5 provided better results than traditional kernels of 3×3 (widely used in VGG-16 and similar architectures).
2. The second experiment performs a specific comparison between CNN models with and without handcrafted features. In this sense, the IP, UP and UH datasets have been considered, employing as spectral-spatial model the CNN baseline, the CNN with extended morphological profiles (EMP-CNN) and the CNN with Gabor filtering (Gabor-CNN) proposed by Ghamisi et al. (2018), whose architectures are composed by two feature extraction and detection stages, where each one contains a stack of CONV-ReLU-POOL layers. These have been fed with input patches of 27×27 pixels, preserving

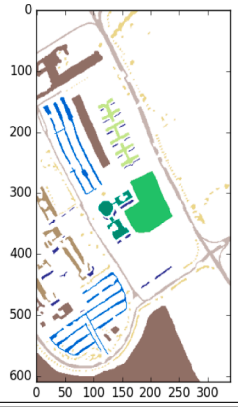
Table 4
Number of available samples in the Indian Pines (IP), University of Pavia (UP), Salinas Valley (SV), and the University of Houston (UH) datasets. The samples for the latter scene are divided in two categories: training (top) and testing (bottom).

INDIAN PINES (IP)



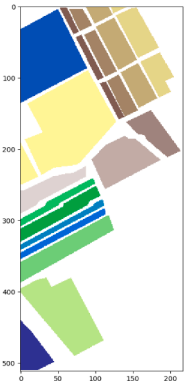
Color	Land-cover type	Samples
	Background	10776
	Alfalfa	46
	Corn-notill	1428
	Corn-min	830
	Corn	237
	Grass/Pasture	483
	Grass/Trees	730
	Grass/pasture-mowed	28
	Hay-windrowed	478
	Oats	20
	Soybeans-notill	972
	Soybeans-min	2455
	Soybean-clean	593
	Wheat	205
	Woods	1265
	Bldg-Grass-Tree-Drives	386
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

UNIVERSITY OF PAVIA (UP)



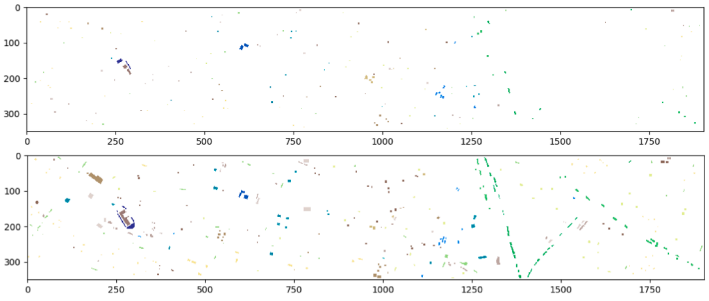
Color	Land-cover type	Samples
	Background	164624
	Asphalt	6631
	Meadows	18649
	Gravel	2099
	Trees	3064
	Painted metal sheets	1345
	Bare Soil	5029
	Bitumen	1330
	Self-Blocking Bricks	3682
	Shadows	947
	Total samples	207400

SALINAS (SV)



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

UNIVERSITY OF HOUSTON (UH)



Color	Land cover type	Samples train	Samples test
	Background	649816	
	Grass-healthy	198	1053
	Grass-stressed	190	1064
	Grass-synthetic	192	505
	Tree	188	1056
	Soil	186	1056
	Water	182	143
	Residential	196	1072
	Commercial	191	1053
	Road	193	1059
	Highway	191	1036
	Railway	181	1054
	Parking-lot1	192	1041
	Parking-lot2	184	285
	Tennis-court	181	247
	Running-track	187	473
	Total samples	2832	12197

three PCs for EMP-CNN and Gabor-CNN models and the full spectrum for the CNN baseline. Also, 50 samples per class have been considered (when using the IP scene) to train the models, and 548, 540, 392, 524, 256, 532, 375, 514, and 231 labels of each class (see Table 4) have been employed for testing the UP scene, while for the UH scene all available training samples have been considered.

3. The third experiment compares the performance of several improved convolutional-based architectures, in particular residual and capsule-based models, over two HSI datasets, using 20% and 10% of the available labeled samples for IP and UP datasets, respectively. We have considered five deep architectures: (1) the spectral-spatial residual network (SSRN) (Zhong et al., 2017b), (2) the spectral-spatial pyramidal residual network (P-RN) (Paoletti et al., 2018c), (3) the densely connected CNN (DenseNet) (Paoletti et al., 2018a), (4) the spectral-spatial dual-path network (DPN) (Kang et al., 2018), and (5) the capsule network (CapsNet) (Paoletti et al., 2018b). Moreover, with the aim of exploring the performance of these methods with different levels of spatial information, four different spatial neighborhoods have been tested: 5×5, 7×7, 9×9 and 11×11.

Table 5
Neural network base model topologies considered in our experiments, emphasizing the input, hidden and output layers in order to demonstrate the depth of each architecture. In this sense, the term "*linear input*" refers to the input layer of each model, while the last densely-connected layer with sofmax function is the output layer. Regarding the input layer, spectral models receive pixel-vectors of n_{bands} elements, while spatial and spectral-spatial methods employ an input patch size of $19 \times 19 \times n_{channels}$, being $n_{channels} = 1$ for CNN2D and $n_{channels} = 40$ for CNN2D40 and CNN3D. Finally, the term "*recurrentLayer*"(†) indicates that this layer has been implemented by a RNN/GRU/LSTM layer, depending on the kind of neural network. The number in the parentheses indicates the number of units (i.e. the dimensionality of the layer).

Model	Main layer	Norm.	Ac. Function	Downsampling
MLP	Linear input(n_{bands})	–	–	–
	FC($n_{bands} \cdot \frac{2}{3} + 10$)	–	ReLU	–
	FC(n_{class})	–	Softmax	–
RNN	Linear input(n_{bands})	–	–	–
GRU	recurrentLayer†(64)	–	Tanh	–
LSTM	recurrentLayer†(64)	–	Tanh	–
	FC(n_{class})	–	Softmax	–
CNN1D	Linear input(n_{bands})	–	–	–
	CONV(20×24)	–	ReLU	POOL(5)
	FC(100)	BN	ReLU	–
	FC(n_{class})	–	Softmax	–
CNN2D	Linear input($19 \times 19 \times n_{channels}$)	–	–	–
	CONV($50 \times 5 \times 5$)	–	ReLU	–
	CONV($100 \times 5 \times 5$)	–	ReLU	POOL(2×2)
CNN2D40	FC(100)	BN	ReLU	–
	FC(n_{class})	–	Softmax	–
CNN3D	Linear input($19 \times 19 \times n_{channels}$)	–	–	–
	CONV($32 \times 5 \times 5 \times 24$)	BN	ReLU	–
	CONV($64 \times 5 \times 5 \times 16$)	BN	ReLU	POOL($2 \times 2 \times 1$)
	FC(300)	BN	ReLU	–
	FC(n_{class})	–	Softmax	–

4. The fourth experiment studies how semi-supervised techniques (in particular, the AL paradigm) are affected by the amount of training data available when combined with DL-models, in particular spectral, spatial and spectral-spatial convolutional-based models, taking into account four classifiers with Bayesian perspective (Haut et al., 2018c): (1) AL-MLR, (2) spectral CNN (CNN1D), (3) spatial CNN with 2-D kernels and input patches keeping one PC with PCA (CNN2D) and (4) spectral-spatial CNN with 3-D kernels and input patches keeping all the spectral bands of the original datasets (CNN3D). The IP and SV datasets have been considered for this experiment.
5. The fith experiment performs two comparisons to analyze the performance of different TLapproaches. Specifically, the first one performs a comparison between five off-the-shelf deep models, studying their classification accuracies over three HSI datasets: IP, UP and SV, and employing the TL paradigm. In this sense (1) VGG16 (Simonyan and Zisserman, 2014), (2) VGG19 (Simonyan and Zisserman, 2014), (3) ResNet50 (He et al., 2016), (4) MobileNet (Howard et al., 2017), and (5) DenseNet121 (Huang et al., 2017) have been considered. These models have been pre-trained with the ImageNet, followed by a general training using IP, UP and SV datasets in order to fit their BN layers, using Adam optimizer, a learning rate of 0.0001 and 5 epochs. Then, a hidden FC layer with 256 neurons and an output FC layer with $n_{classes}$ neurons have been added at the end of these models, which were fine-tuned employing several training percentages (1%, 5%, 10%, 15%, 20% and 25%). This fine-tuning is carried out with Adam optimizer, a learning rate of 0.001 and 50 epochs. In addition, a second comparison is carried out, comparing the performance of the CNN1D, CNN2D, CNN2D40 and CNN3D models implemented in our first experiment (see Table 5) employing the TL paradigm. In this sense, IP and SV have been considered because of their spectral similarities, as the two scenes were collected by the same spectrometer (AVIRIS). First, these models have been pre-tained using the IP scene, because of its spectral complexity, and then tested over the SV scene. The model parameters are adjusted with 2, 4, 8, 16, 32, 64, 128 and 256 samples per SV class.
6. All previous experiments have been developed by randomly selecting the training data from the available set of labeled samples (with the exception of UH scene, which employs its own set of fixed training samples). In this context, new trends suggest that the high correlation between neighboring pixels can affect the performance of the network, in the sense that the test set will be very close to the train set, allowing the model to obtain too optimistic results, which are not adjusted to the real generalization power of the model. Regarding this, our sixth experiment compares the performance of the models considered on the first experiment (i.e. RF, MLR, SVM, MLP, RNN, GRU, LSTM, CNN1D, CNN2D, CNN2D40 and CNN3D) trained with spatially disjoint samples of IP and UP datasets (these training and test sets are available from the GRSS DASE website at http://dase.grss-ieee.org).

In order to assess the results of these experiments, three widely used quantitative metrics are used to evaluate the classification performance: (i) the *overall accuracy* (OA), that computes the number of correctly classified HSI pixels divided by the number of samples, (ii) the *average accuracy* (AA), that computes the mean of the classification accuracies of all classes, and (iii) the *Kappa coefficient*, that measures the agreement between the obtained classification map and the original ground-truth map.

All our experiments have been conducted on a hardware environment composed by a 6th-generation Intel R Core TM i7-6700 K processor, with 8 MB of Cache and a processing speed of 4.20 GHz with 4 cores/8 way multi-task processing. It includes 40 GB of DDR4 RAM with 2400 MHz serial speed and a Toshiba DT01ACA hard disk with 7200RPM and 2 TB capacity. The environment is completed with a NVIDIA GeForce GTX 1080 graphics processing unit (GPU) with 8 GB GDDR5X video memory and 10 Gbps memory rate, and an ASUS Z170

pro-gaming motherboard. The software environment consists of the Ubuntu 18.04.1 x64 operating system with CUDA 9.0 and cuDNN 7.1.1 and Python 2.7 as the programming language.

7.3. Experimental discussion

7.3.1. Comparison between standard supervised HSI classifiers and DL-based networks

Our first experiment intends to compare different supervised classifiers, analyzing how the training percentage affects their performance. In this sense, the considered methods can be separated into two broad categories: traditional ML-based methods (RF, MLR, SVM, MLP) and DL-based networks (RNN, GRU, LSTM, CNN1D, CNN2D, CNN2D40 and CNN3D). Also, a second categorization can be made by dividing the proposed methods into spectral classifiers (RF, MLR, SVM, MLP, RNN, GRU, LSTM, CNN1D), spatial classifiers (CNN2D), and spectral-spatial classifiers (CNN2D40 and CNN3D).

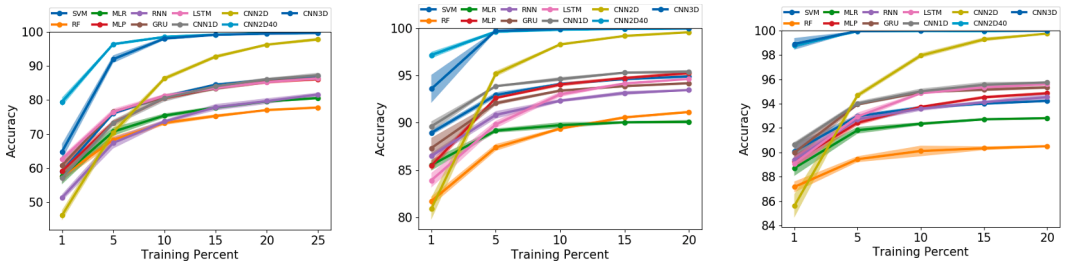
Fig. 11 gives the obtained results for IP, UP and SV datasets after the execution of five Monte-Carlo runs. It is interesting to analyze the behavior of traditional ML methods when few labeled samples are available: they are highly affected by the lack of training data, being the MLP the one exhibiting the best performance and RF the worst, in general. Also, the pixel-based DL classifiers: vanilla RNN, GRU, LSTM and CNN1D are highly affected by the limited availability of training samples, exhibiting a similar behavior with regards to SVM and MLP, with slightly higher accuracy when enough training data are employed. In this case, we highlight the more stable performance of the CNN1D. Regarding the spatial classifier (CNN2D), it presents the worst accuracy when few samples are used in the training stage, even below traditional ML methods, although the spatial information when using patches of size 19×19 seems to be sufficient to reach a remarkably good accuracy with 15%-25% of training. Although the use of spatial information is highly effective (with a suitable training percent), the conjunction of spatial and spectral information achieves the best classification results. In this sense, the CNN2D40 and the CNN3D are able to achieve an OA near 100% with only 5% of training data in the considered datasets, being the IP the hardest scene to classify in our opinion. If we compare CNN2D and CNN2D40, we can observe how the spectral information is able to reduce the uncertainty of the classifier when few training data is available. In addition, the 2-D kernels of CNN2D40 classifier allows to reduce the overfitting in comparison with the 3-D kernels of the CNN3D, reaching similar results when enough training data are available.

Figs. 12–14 and Tables 6–8 present detailed classification maps and accuracy measures for IP (15% of training), UP (10%) and SV (10%) scenes. As we can observe, the spectral classifiers exhibit the familiar *salt and pepper* noise (significantly less in the DL-based methods), because they ignore spatial-contextual information when providing a pixel prediction. On the contrary, spatial and spectral-spatial classifiers exhibit more regular results, with less noise at the edges. However, the spatial CNN2D results often degrade some object and material shapes, a problem that is considerably reduced with the spectral-spatial CNN3D, providing classification results that are more similar with regards to the corresponding ground-truth maps for IP, UP and SV datasets. In addition, Tables 6–8 indicate the runtime of each considered method, being the standard ML-based classifiers the fastest ones (in particular, the SVM) although the consumed time during their parameter search has not been reflected, and the CNN-based algorithms the slowest ones due to the computational complexity of the CONV layers. Moreover, we can observe the number of parameters that each neural model needs to adjust during the training phase, being the MLP the model with fewest parameters and the CNN3D the one with the most parameters to fit.

Also, in order to provide a detailed comparison with a HSI benchmark, Table 9 and Fig. 15 show the classification results of considered methods over the UH dataset, employing the available training data to adjust the parameters of each supervised model. As we can observe in Table 9, spectral classifiers (RF, MLR, SVM, and MLP, RNN, GRU, LST and CNN1D) are able to reach good accuracies: between 73–87% of OA, with the CNN1D being the best pixel-wise classifier, because its kernel is able to process the spectral signatures in a more robust way than tradidional ML models and FC architectures of neural-inspired models. However, if we focus on the spatial classifier (CNN2D), we can see that it exhibits the worst OA, AA and Kappa values. This behaviour may be due to the fact that the reduction of the spectrum to a single band can generate samples that are very mixed and difficult to discriminate. In this sense, the available training samples are less descriptive for setting the parameter values, and they become insufficient for the 378015 parameters of the spatial model. In this sense, the spectral information is the key to discriminate correctly the samples of the UH dataset, as it can be observed in the spectral-spatial CNN2D40. Although this model has 48750 parameters more than its spatial counter-part, the CNN2D40 is able to take into account the original spectral information in its spatial features, obtaining feature maps that are more representative of the input data and being 1.86 times better than those provided by the spatial CNN2D. Furthermore, the 3-D kernels of the spectral-spatial CNN3D model are able to process these spectral features, combining them with the spatial information in order to obtain the output volumes. The classification maps in Fig. 15 demonstrate that spectral classifiers are very noisy, being in general unable to classify the area hidden by the cloud in the UH scene, while the CNN3D reaches a better result in general (see the parking areas, for instance) and showing some spatial structures of the hidden area under the cloud, such as buildings and parking lots.

7.3.2. Comparison between convolutional models, with and without handcrafted features

Our second experiment compares the performance of: (i) a classic spectral-spatial CNN for HSI classification (Ghamisi et al., 2018), which receives as input data patches of size $27 \times 27 \times n_{bands}$ extracted from the original cube, (ii) a spatial CNN that processes extended morphological profiles (EMP-CNN) obtained from the HSI data (Ghamisi et al., 2018) (using input patches of $27 \times 27 \times 3$), and (iii) a spatial CNN that



(a) Classification on IP (b) Classification on UP (c) Classification on SV

Fig. 11. OA evolution (y-axis) of each considered classifier with different training percentages (x-axis) over IP, UP, SV datasets. The standard deviation is also shown around each plot.

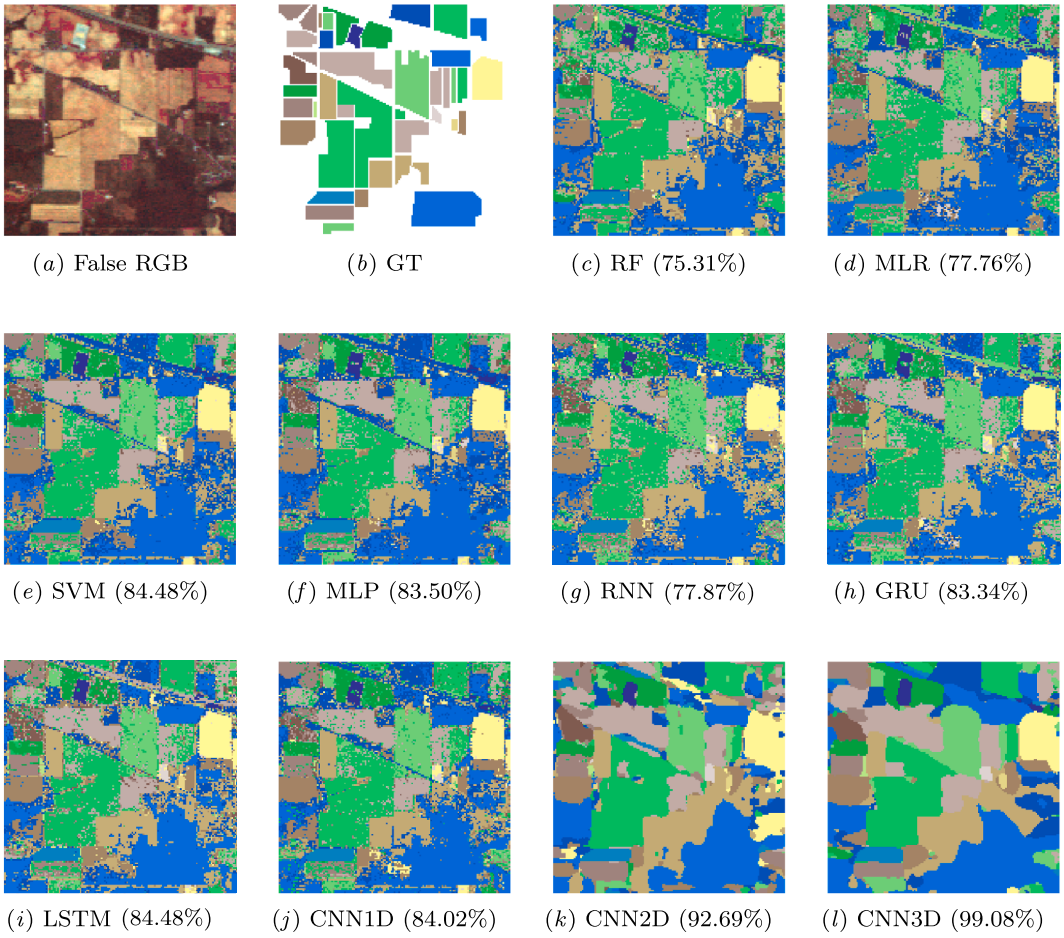


Fig. 12. Classification maps for the IP dataset with 15% of training data. Images from (a) to (j) provide the classification maps corresponding to Table 6. The corresponding overall classification accuracies (OAs) are shown in brackets.

processes the Gabor filtered data (Gabor-CNN) (Ghamisi et al., 2018), also employing input patches of size $27 \times 27 \times 3$, in order to observe the effects of extracting deep features directly from the data or from handcrafted features.

The results obtained over three HSI datasets: IP, UP and UH, are reported on Table 10. If we focus on the CNN baseline, the obtained results are in line with those shown in Tables 6, 7 and 9. Comparing the baseline with EMP-CNN and Gabor-CNN models, it is easy to confirm that the Gabor-CNN model exhibits the best performance for all the considered datasets, with the EMP-CNN being slightly worse. In this context, the CNN-baseline appears to provide the worst results in this particular case, with two to four percentage points below the Gabor-CNN. With these results in mind, we highlight that spatial-based processing of the data by powerful pre-processing methods, such as EMPs and Gabor filters, can significantly improve the performance of convolutional models. Particularly, Gabor filters exhibit optimal localization properties in both the spatial and frequency domains, allowing for the successful combination of spatial and spectral information for the extraction of edges and textures, while EMPs are also quite effective in the task of modelling the spatial-contextual information contained in the HSI data cube. This confirms and extends the obtained results of previous works, such as the one by Anwer et al. (2018), where explicit texture descriptors (local binary patterns) are used to improve classification results on several pre-trained models and aerial remote sensing benchmarks with RGB images.

7.3.3. Comparison between improved convolutional-based architectures

Our third experiment performs a study about the performance of improved convolutional-based models, considering different levels of spatial information. In this sense, it must be noted that these architectures have been particularly developed to efficiently exploit their depth, to obtain deeper and more abstract features, while avoiding the problems associated with the depth through communication mechanisms that reuse the data of the model, such as residual connections or dynamic routing. Table 11 shows the obtained results. As we can observe in the table, these methods are able to reach good accuracy, with small-sized patches being able to reach the 99% of OA using patches of size 7×7. In addition, although the SSRN and the P-RN use the same residual learning approach, the selection of the topology and the residual block architecture can substantially improve the performance of the network. In particular, the SSRN implements two networks (both with two residual units): one spectral network with all its kernels of size $1 \times 1 \times 7$, and one spatial network with all its kernels of size $3 \times 3 \times 128$. This reduces the number of parameters but prevents the efficient extraction of spectral-spatial information. However, the P-RN introduces only one network with three pyramidal residual modules, each one composed by three pyramidal bottleneck residual units, implementing its CONV layers of kernels 1×1, 7×7 and 8×8. Although the P-RN is more complex and deep than the SSRN, its performance is significantly better. At the end, the topology allows the P-RN to achieve significant precision gains, especially with smaller input spatial sizes. Also, it is interesting to highlight the standard deviation of both classifiers, which is lower in the P-RN model. The residual block architecture of P-RN is able to extract additional feature maps (as the residual units become deeper) in comparison with the SSRN, exploiting better the information contained within HSI input patches. In the end, this improves the OA results and reduces the standard deviation, i.e. the uncertainty.

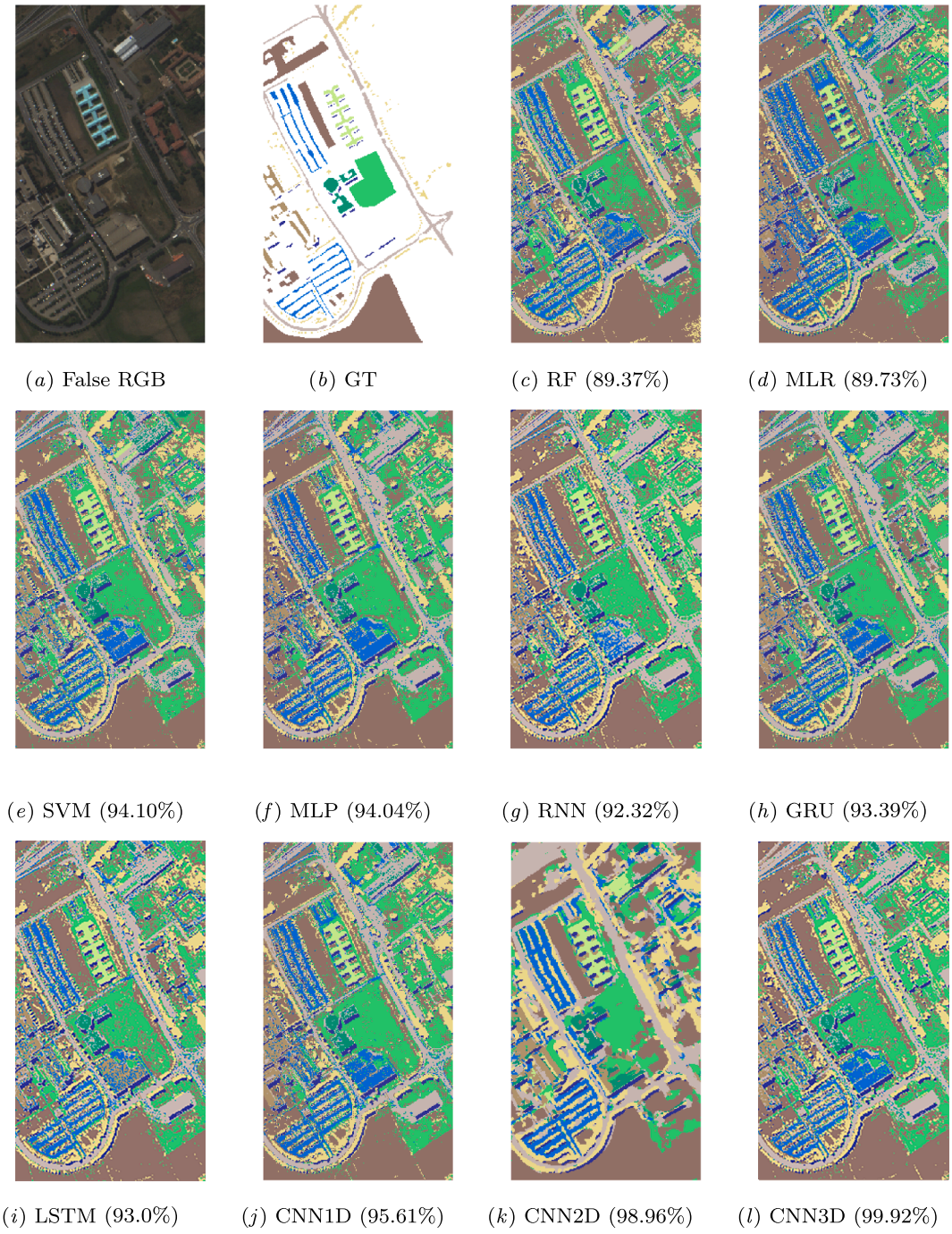


Fig. 13. Classification maps for the UP dataset with 10% of training data. Images from (a) to (j) provide the classification maps corresponding to Table 7. The corresponding overall classification accuracies (OAs) are shown in brackets.

Looking at the results obtained by DenseNet and CapsNet, these classifiers exhibit very similar behavior, reaching accuracy values between those obtained by the SSRN and P-RN when small spatial patches are used as input data (maintaining significant quantitative improvements with respect to the other HSI classifiers in the previous experiment), and even outperforming the results obtained with a high amount of spatial-contextual information. Finally, if we compare the residual models (SSRN and P-RN) and the DenseNet with the DPN model, we can observe that the DPN is able to outperform the results obtained by the SSRN with few training samples, while its accuracy is normally between that achieved by the P-RN and the DenseNet.

7.3.4. Comparison between semi-supervised and AL models

Our fourth experiment explores the use of labeled data during the training stage in AL models, with the aim of analyzing how the amount of training data affects their performance. In this context, the considered models follow a Bayesian perspective (Haut et al., 2018c), where each one extracts probabilistic information about the samples, in order to select those samples that provide more information to the model while, at the same time, reducing the number of training samples.

The obtained results over IP and SV datasets are shown in Table 12. As we can observe, the AL-CNN3D is able to reach 99% OA with approximately 3.92% of labeled data from IP and 0.53% of labeled data

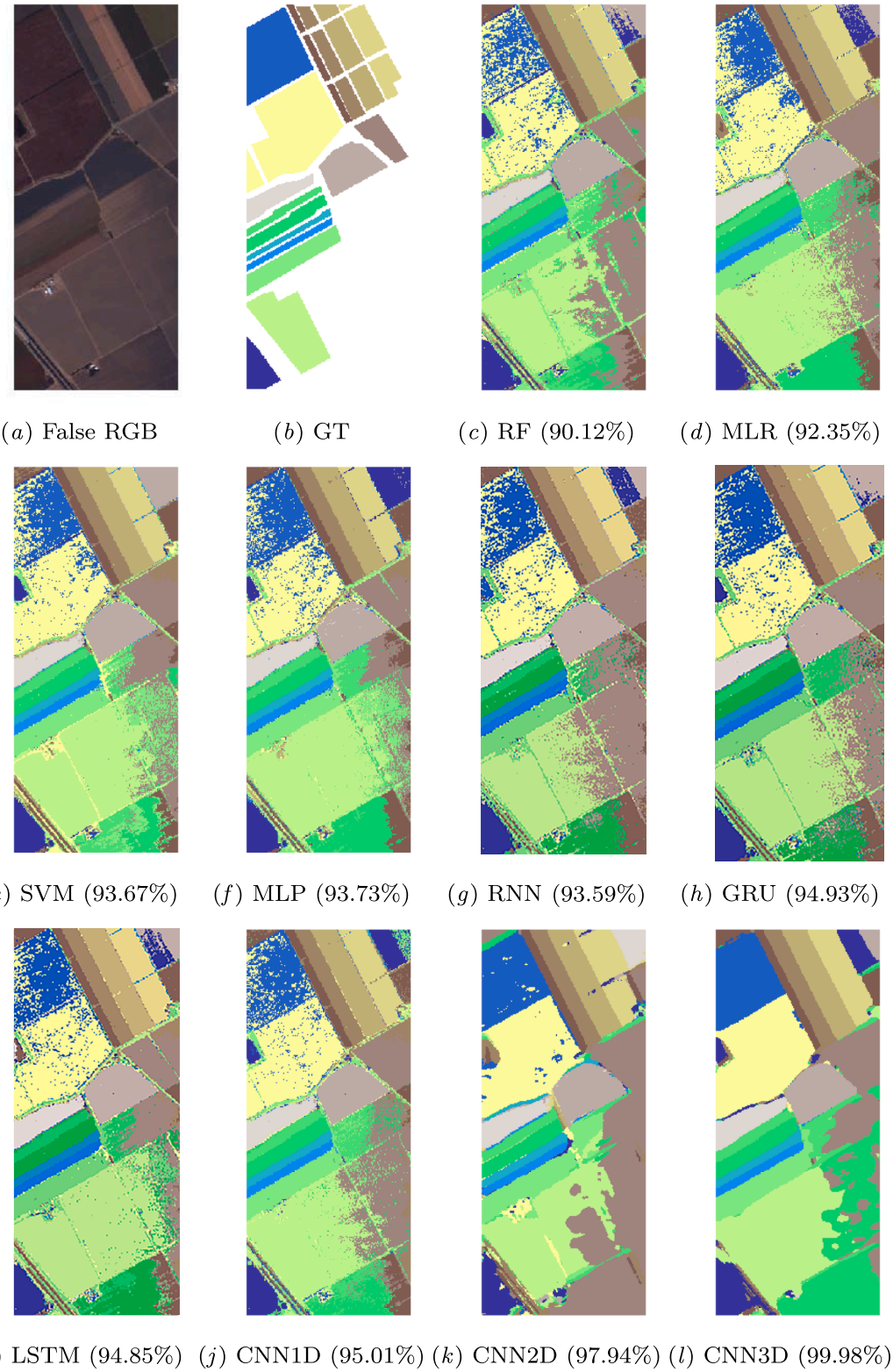


Fig. 14. Classification maps for the SV dataset with 10% of training data. Images from (a) to (j) provide the classification maps corresponding to Table 8. The corresponding overall classification accuracies (OAs) are shown in brackets.

from SV, which is in line with the results obtained in previous experiments: IP exhibits higher complexity compared with SV, whose pixels are spectrally less mixed and the spatial distribution is more geometric, with bigger areas made up of crops. On the contrary, the AL-MLR and AL-CNN1D are unable to reach such high OAs. For instance the AL-CNN1D is not able to improve 90% OA with the IP dataset, and it also cannot reach 99% OA with the SV scene. Furthermore, although the AL-CNN2D classifier is able to reach 99% OA in all the considered HSI scenes, it generally needs more labeled data than its spectral-spatial counterpart. These results strongly support the fact that joint spectral-spatial features are more useful than separate spatial and spectral features, making the AL-CNN3D model ideal for the extraction of highly

Table 6
Classification results for the IP dataset using 15% of the available labeled data.

Class	RF	MLR	SVM	MLP	RNN	GRU	LSTM	CNN1D	CNN2D	CNN2D40	CNN3D
Alfalfa	20.00	32.82	62.05	50.77	36.92	57.43	80.51	44.61	75.38	95.39	**96.92**
Corn-notill	61.53	75.07	81.45	78.90	73.49	80.17	82.19	81.04	91.54	98.73	**98.91**
Corn-min	53.62	57.96	70.55	66.27	58.04	70.33	70.16	70.69	86.95	**98.95**	98.84
Corn	35.12	45.67	72.93	61.19	44.28	66.47	53.83	60.10	88.56	**99.50**	97.71
Grass/Pasture	84.39	86.98	93.17	89.61	86.98	88.83	89.76	92.34	86.05	98.58	**99.32**
Grass/Trees	96.10	96.36	97.32	96.55	96.97	95.84	96.77	97.29	96.13	99.06	**99.74**
Grass/pasture-mowed	29.57	47.83	84.35	75.65	53.91	75.65	81.74	69.57	82.61	**93.91**	93.04
Hay-windrowed	96.11	99.16	98.32	97.54	98.67	98.67	98.52	98.18	97.88	100.00	100.00
Oats	1.18	18.82	51.76	61.18	27.06	68.23	62.35	44.70	65.88	98.82	**100.00**
Soybeans-notill	65.96	66.54	77.87	78.18	67.41	78.86	76.78	78.67	89.85	99.15	**99.15**
Soybeans-min	89.13	79.53	85.10	86.10	80.09	81.87	83.13	83.42	95.28	**99.62**	99.23
Soybean-clean	46.59	58.25	79.09	78.85	65.56	81.11	80.75	83.97	88.65	97.14	**97.86**
Wheat	92.18	98.51	98.39	98.74	97.93	98.51	98.62	98.62	97.82	99.77	**99.89**
Woods	94.53	95.31	95.59	94.55	92.11	95.35	93.58	94.51	98.40	**99.87**	99.59
Bldg-Grass-Tree-Drives	40.55	63.90	61.28	65.55	65.18	64.21	67.44	67.44	89.21	**99.45**	98.48
Stone-steel towers	83.54	85.06	87.60	89.37	86.08	86.58	82.78	87.59	82.53	**96.20**	95.70
OA	75.31	77.76	84.48	83.50	77.87	83.34	83.48	84.02	92.69	**99.14**	99.08
AA	61.88	69.24	81.05	79.31	70.67	80.51	81.18	78.30	88.29	98.38	**98.40**
K(x100)	71.41	74.46	82.26	81.13	74.65	80.98	81.13	81.75	91.65	**99.02**	98.95
Parameters				31047	217296	242640	255248	72616	378116	426866	1805196
Time (s.)	1.29	6.05	**0.25**	26.46	63.59	47.22	53.36	53.91	59.28	103.76	185.07

Table 7
Classification results for the UP dataset using 10% of the available labeled data.

Class	RF	MLR	SVM	MLP	RNN	GRU	LSTM	CNN1D	CNN2D	CNN2D40	CNN3D
Asphalt	91.63	92.39	94.29	93.81	92.33	94.33	93.02	95.85	98.01	99.97	**100.00**
Meadows	97.71	96.09	97.49	97.58	97.08	96.98	97.01	98.13	99.41	99.98	**100.00**
Gravel	66.88	73.27	80.84	78.11	75.43	77.63	78.18	81.48	93.90	**99.43**	99.35
Trees	89.10	86.90	94.21	93.59	91.89	94.04	94.14	94.15	98.14	99.32	**99.74**
Painted metal sheets	98.60	99.59	99.22	99.52	99.49	99.44	99.54	99.82	99.57	100.00	100.00
Bare Soil	64.35	77.83	90.91	91.64	87.2	88.14	86.4	91.71	98.08	99.99	**100.00**
Bitumen	77.66	56.34	87.35	85.53	82.07	84.88	86.77	87.52	89.72	99.80	**99.98**
Self-Blocking Bricks	88.52	86.68	87.47	88.92	84.38	88.37	87.27	85.68	98.28	99.61	**99.74**
Shadows	99.74	99.67	99.86	99.53	99.7	99.67	99.79	**99.88**	98.87	98.33	99.60
OA	89.37	89.73	94.10	94.04	92.32	93.39	93.0	94.61	98.27	99.83	**99.92**
AA	86.02	85.41	92.40	92.02	89.95	91.5	91.35	92.69	97.11	99.60	**99.82**
K(x100)	85.67	86.27	92.17	92.09	89.79	91.22	90.7	92.84	97.71	99.78	**99.89**
Parameters				8823	71817	97161	109769	33909	377409	426159	1803089
Time (s.)	4.29	8.63	**0.44**	68.22	150.06	113.07	128.09	139.58	139.82	226.22	448.32

discriminative features for classification purposes.

7.3.5. Comparison between transfer learning approaches

In the first test of our fifth experiment, we compare the performance of five off-the-shelf DL-based models, which have been pre-trained over the ImageNet and fine-tuned with different training percentages for the IP, UP and SV datasets.

The obtained results are given in Fig. 16. Focusing on the IP dataset, it can be observed that, with only 1% of labeled samples, the best OA is reached by the DenseNet121, which achieves 67.80% OA, being closely followed by the MobileNet. In this regard, it must be noted that IP dataset exhibits higher complexity than the UP and SV scenes, where the best results with 1% of the available labeled samples used for training are achieved by DenseNet121 with 94.37% and 98.02%, respectively. However we must highlight that, although these deep and very deep models have not been specifically developed for HSI analysis, they are able to reach interesting results in comparison with those obtained by the specially-designed CNN models in the first experiment (see Section 7.3.1). For instance, if we focus on the IP scene, VGG16, Resnet50, MobileNet and DenseNet121 models are able to outperform the CNN2D model with 1% of training data, while MobileNet and DenseNet121 outperform the results obtained by CNN1D and CNN2D40. With the UP dataset, VGG16, Resnet50, MobileNet and DenseNet121 models outperform the OA of CNN1D and CNN2D models, while with the SV dataset all pre-trained models outperform the CNN2D's results, and VGG16, Resnet50, MobileNet and DenseNet121 improve the classification accuracy of the CNN1D.

When more labeled samples are used for training, the OA increases quite fast. For instance, with 5% of training data, the vast majority of classifiers are able to reach at least 90% OA in the IP scene, and 99% in the UP and SV scenes, with few exceptions (for instance the VGG19 with the IP scene). Compared with the previous results reported on Section 7.3.1 we can observe that, in general, pre-trained models are slightly worse than the CNN2D40 and the CNN3D. In addition to the architectural design of the models, it must be highlighted that the spatial size and the spectral resolution of the input patch is decisive in improving the behavior of these deep networks. In this case, all the TL-based models have been fed with patches of size $32 \times 32 \times 3$, which are then scaled to the original inputs of the networks (for instance, the VGG16 employs patches of $224 \times 224 \times 3$). This limitation forces us to reduce the spectral dimensionality with PCA, which leads to a reduced capacity for spectral discrimination (while employing an excessively

Table 8
Classification results for the SV dataset using 10% of the available labeled data.

Class	RF	MLR	SVM	MLP	RNN	GRU	LSTM	CNN1D	CNN2D	CNN2D40	CNN3D
Brocoli green weeds 1	99.46	99.47	99.63	99.57	99.48	99.67	99.42	99.88	99.45	99.90	**100.00**
Brocoli green weeds 2	99.83	99.94	99.91	99.87	99.91	99.94	99.9	99.96	99.51	99.96	**100.00**
Fallow	99.15	98.60	99.68	99.44	99.1	99.58	99.65	99.85	99.62	100.00	100.00
Fallow rough plow	99.42	99.28	99.31	99.25	98.36	99.52	99.39	99.57	**99.89**	99.84	99.86
Fallow smooth	97.87	99.12	99.35	99.09	98.56	99.33	99.37	99.05	99.88	99.88	**99.95**
Stubble	99.68	99.92	99.80	99.85	99.8	99.86	99.89	99.85	99.78	100.00	100.00
Celery	99.39	99.89	99.54	99.57	99.71	99.75	99.7	99.84	99.64	100.00	100.00
Grapes untrained	84.42	87.98	90.51	86.88	87.22	89.83	90.79	90.98	95.60	99.96	**99.97**
Soil vinyard develop	99.07	99.73	99.92	99.73	99.77	99.79	99.76	99.83	99.54	100.00	100.00
Corn senesced green weeds	91.56	95.79	97.71	96.56	96.49	97.56	96.63	98.03	98.45	99.94	**99.99**
Lettuce romaine 4wk	94.13	95.90	98.88	97.81	97.59	98.52	98.86	98.33	98.73	99.94	**100.00**
Lettuce romaine 5wk	98.79	99.63	99.79	99.65	99.46	99.87	99.63	99.96	99.58	99.99	99.99
Lettuce romaine 6wk	97.86	99.03	98.88	99.03	98.45	98.66	99.05	99.17	99.13	**100.00**	99.98
Lettuce romaine 7wk	91.34	96.03	97.65	96.80	96.82	98.09	97.61	97.34	97.53	99.88	**99.98**
Vinyard untrained	60.46	66.63	70.54	77.81	76.79	80.98	79.59	79.52	95.01	**99.96**	99.95
Vinyard vertical trellis	97.06	98.89	99.18	99.08	98.95	99.07	98.7	99.00	97.00	99.94	**99.94**
OA	90.12	92.35	93.67	93.73	93.59	94.93	94.85	95.01	97.94	99.96	**99.98**
AA	94.34	95.99	96.89	96.87	96.65	97.5	97.37	97.51	98.65	99.95	**99.98**
K(x100)	88.98	91.47	92.94	93.02	92.86	94.35	94.27	94.44	97.71	99.96	**99.98**
Parameters				32282	221392	246736	259344	74616	378116	426866	1805196
Time (s.)	2.85	65.21	**0.94**	86.63	191.62	152.44	163.35	177.78	177.29	282.69	551.72

large spatial size).

At this point, it is important to note that the use of TL-based approaches in image processing tasks has two main benefits: the ability to achieve good results with few training samples extracted from the target scene and the reduction of runtime in the training procedure. However, although TL-based approaches are fairly reliable when few labeled samples are available, the models employed for HSI classification are based on those trained by the DL community over RGB datasets, such as ImageNet. In this sense, the effectiveness of TL methods depends mostly on the source application with which the models were pre-trained, and on the relationship with the final target application in which they will be used (Patricia and Caputo, 2014). In such case, the Imagenet dataset is not related with the employed HSI data and, hence, it was expected that these models would not be able to exhibit their full potential in HSI classification.

To overcome this limitation, in our second test we study the performance of the proposed CNN1D, CNN2D, CNN2D40 and CNN3D models implemented on the first experiment (see Table 5) employing the TL paradigm over two HSI datasets: IP and SV. In this context, the pursued goal is to take advantage of TL's ability to learn general knowledge from other datasets and then apply such knowledge to a specific task, polishing it on the target scene. Regarding this goal, we take advantage of the most spectrally mixed and difficult samples from IP to later recognize more precise characteristics in SV, employing 100% of the labeled data from IP scene (i.e. 10249 samples) to perform the pre-training stage, while extracting 2, 4, 8, 16, 32, 64, 128 and 256 samples from SV scene to adjust the considered models. The obtained results are given in Fig. 17. As we can observe, the behaviour is very similar for each model. In other words, pre-training with IP labels allows the considered models to achieve better accuracy when they are inferring the SV samples with very few training samples. However, the improvement is less significant when additional labeled samples from SV are added to adjust the model parameters, which demonstrates that (broadly speaking) TL is only recommended when there are very few

Table 9
Classification results for UH dataset using the fixed training set available.

Class	RF	MLR	SVM	MLP	RNN	GRU	LSTM	CNN1D	CNN2D	CNN2D40	CNN3D
Grass healthy	82.49	**82.62**	82.34	81.58	82.19	82.24	82.05	81.75	61.12	80.48	81.79
Grass stressed	83.36	83.93	83.36	81.67	83.44	81.35	81.56	**95.04**	50.08	85.49	87.20
Grass synthetic	97.82	99.80	99.80	99.64	99.84	99.88	99.76	99.88	29.35	88.99	94.73
Tree	91.74	98.01	**98.96**	88.69	94.64	96.14	91.89	89.45	46.61	83.66	84.74
Soil	96.80	97.16	98.77	97.08	97.99	97.12	97.56	98.63	41.36	**100.00**	99.81
Water	99.16	94.41	97.90	94.41	95.24	**99.3**	96.5	95.94	44.06	92.59	97.76
Residential	75.28	74.25	77.43	76.79	**81.05**	77.76	78.1	80.88	61.14	74.65	76.56
Commercial	33.01	65.15	60.30	55.82	42.72	48.4	39.79	80.32	32.95	80.85	**81.06**
Road	69.40	69.12	76.77	69.91	79.28	74.96	77.94	77.09	59.43	81.34	**88.46**
Highway	43.86	54.44	61.29	49.71	48.86	61.64	48.17	72.57	32.45	63.69	**78.30**
Railway	70.36	76.09	80.55	75.67	74.84	80.91	77.53	86.36	44.42	93.74	**96.28**
Parking lot1	54.77	73.39	79.92	77.16	74.99	81.73	81.4	91.91	33.68	96.96	**98.91**
Parking lot2	60.14	68.42	70.88	72.21	69.61	69.4	71.02	74.74	**84.00**	82.88	72.56
Tennis court	98.87	98.79	100.00	99.03	100.0	99.92	99.43	99.36	68.67	98.79	97.90
Running track	97.50	95.98	96.41	**98.31**	97.29	97.76	97.25	98.14	15.69	97.34	96.36
OA	73.09	79.53	81.86	77.98	78.44	80.39	78.11	86.66	45.80	85.18	**87.95**
AA	76.97	82.10	84.31	81.18	81.46	83.23	81.33	88.14	47.00	86.76	**88.83**
K(x100)	71.09	77.89	80.43	76.29	76.75	78.79	76.46	85.53	41.53	83.90	**86.91**
Parameters				16975	150735	176079	188687	50515	378015	426765	1804895
Time (s.)	2.68	21.25	**0.37**	46.09	105.81	78.45	88.90	94.41	81.69	165.33	311.56

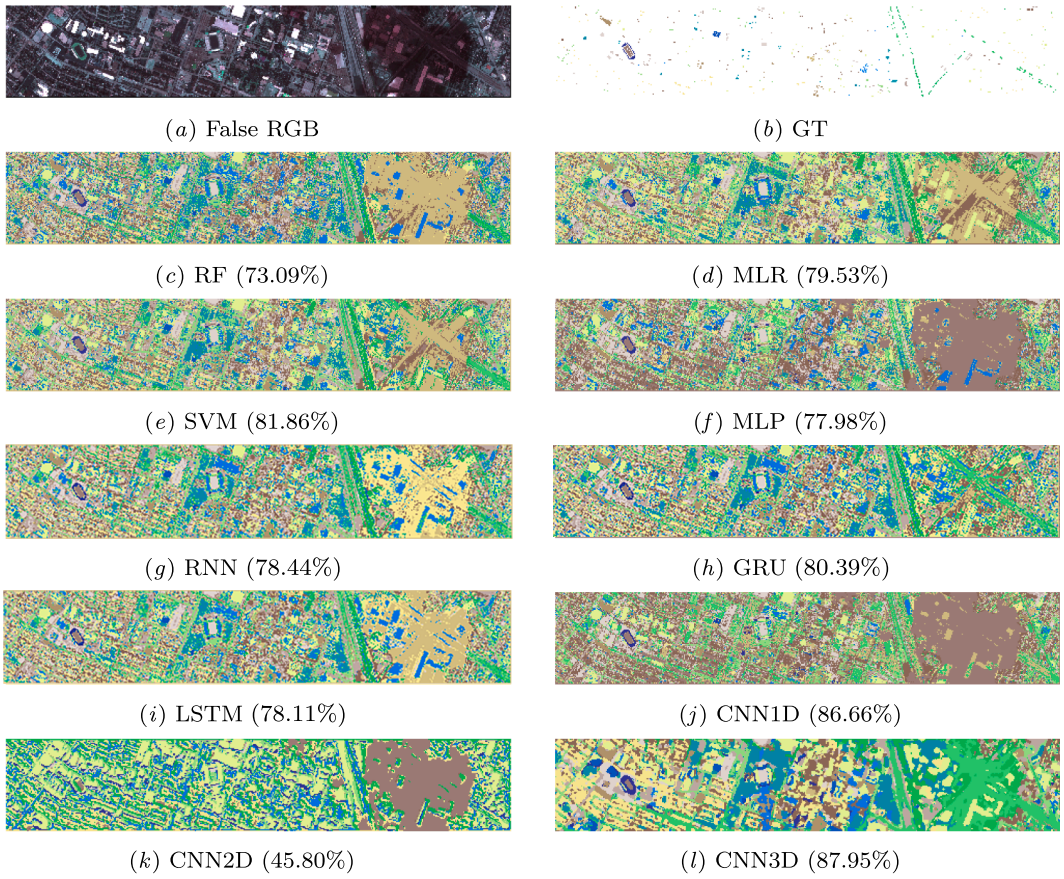


Fig. 15. Classification maps for the UH dataset. Images from (a) to (j) provide the classification maps corresponding to Table 9. The corresponding overall classification accuracies (OAs) are shown in brackets.

samples in the target scene and there is another (larger) dataset with similar characteristics that can help to model the parameters of a classification network in a reasonable way.

7.3.6. Training and testing with spatially disjoint samples

As it can be observed on Figs. 18 and 19, given a particular HSI scene, spectral-spatial DL-based classifiers have been traditionally trained by extracting randomly selected samples (from the available ground-truth) over the whole image, and cropping spectral-spatial patches of $d \times d \times n_{channels}$ pixel-centered neighbors. In this sense, it is likely that the test set is very close to the train set, or even that part of the test is used in the train set as part of the neighboring region $d \times d$

Table 10
Classification results for the IP, UP and UH datasets considering the CNN models with EMP and Gabor handcrafted features (Ghamisi et al., 2018).

IP dataset				UP dataset				UH dataset			
Class	**CNN**	**EMP CNN**	**GABOR CNN**	**Class**	**CNN**	**EMP CNN**	**GABOR CNN**	**Class**	**CNN**	**EMP CNN**	**GABOR CNN**
Alfalfa	79.25	**85.02**	84.44	Asphalt	88.43	**95.87**	87.75	Grass healthy	82.33	**87.49**	87.47
Corn-notill	90.14	73.45	**91.53**	Meadows	91.64	**99.50**	97.25	Grass stressed	84.30	80.99	**86.01**
Corn-min	98.77	**100.00**	98.77	Gravel	**75.95**	61.12	70.92	Grass synthetic	**95.84**	87.72	78.22
Corn	90.94	92.8	**94.70**	Trees	96.53	94.81	**97.09**	Tree	**92.60**	90.43	85.02
Grass/Pasture	98.85	98.70	**99.28**	Painted metal sheets	98.56	95.15	**98.83**	Soil	99.90	**100.00**	99.89
Grass/Trees	100.00	100.00	100.00	Bare Soil	57.87	**64.84**	64.62	Water	93.00	**97.90**	89.44
Grass/pasture-mowed	95.10	93.13	**95.84**	Bitumen	80.43	**80.63**	76.66	Residential	80.39	**90.48**	90.19
Hay-windrowed	91.20	**92.25**	90.94	Self-Blocking Bricks	98.10	97.26	**99.05**	Commercial	70.42	58.51	**74.44**
Oats	94.34	**94.85**	88.59	Shadows	96.84	96.08	**98.36**	Road	77.77	79.77	**84.42**
Soybeans-notill	100.00	100.00	100.00					Highway	56.08	**64.28**	63.61
Soybeans-min	95.54	99.34	99.34					Railway	75.59	78.37	**80.06**
Soybean-clean	89.66	89.53	89.66					Parking lot1	86.55	78.29	**87.30**
Wheat	100.00	100.00	100.00					Parking lot2	84.21	76.84	**85.06**
Woods	100.00	100.00	97.37					Tennis court	93.11	99.19	**100.00**
Bldg-Grass-Tree-Drives	100.00	100.00	100.00					Running track	**88.37**	77.04	56.95
Stone-steel towers	100.00	100.00	100.00								
OA	91.53	92.40	**92.84**	OA	87.01	91.37	**91.62**	OA	82.75	84.04	**84.12**
AA	95.24	94.94	**95.65**	AA	87.15	87.25	**87.83**	AA	**84.04**	83.33	82.94
K	90.08	91.05	**91.61**	K	83.08	88.67	**89.14**	K	80.61	**82.54**	82.51

Table 11
Overall accuracy (%) achieved by different DL-based approaches when considering different sizes of the input spatial patches. Also, for each model a parameter estimation has been conducted in order to provide an overview of the different architectures.

Spatial Size	SSRN	P-RN	DenseNet	DPN	CapsNet
			IP dataset		
5 × 5	92.83 ± 0.66	**98.80** ± 0.10	97.85 ± 0.28	97.53 ± 0.15	97.79 ± 0.40
7 × 7	97.81 ± 0.34	99.26 ± 0.06	99.24 ± 0.14	99.29 ± 0.06	**99.30** ± 0.11
9 × 9	98.68 ± 0.29	99.64 ± 0.08	99.58 ± 0.09	99.64 ± 0.10	**99.67** ± 0.06
11 × 11	98.70 ± 0.21	**99.82** ± 0.07	99.74 ± 0.08	99.67 ± 0.06	99.74 ± 0.09
			UP dataset		
5 × 5	98.72 ± 0.17	**99.52** ± 0.05	99.13 ± 0.08	99.21 ± 0.11	99.13 ± 0.08
7 × 7	99.54 ± 0.11	**99.81** ± 0.09	99.71 ± 0.10	99.70 ± 0.07	99.75 ± 0.03
9 × 9	99.57 ± 0.54	99.79 ± 0.11	99.73 ± 0.15	**99.88** ± 0.04	99.73 ± 0.10
11 × 11	99.79 ± 0.08	99.92 ± 0.02	99.93 ± 0.03	**99.94** ± 0.03	99.93 ± 0.02
Parameters	360 K.	2.4 M.	1.7 M.	370 K.	9.0 M.

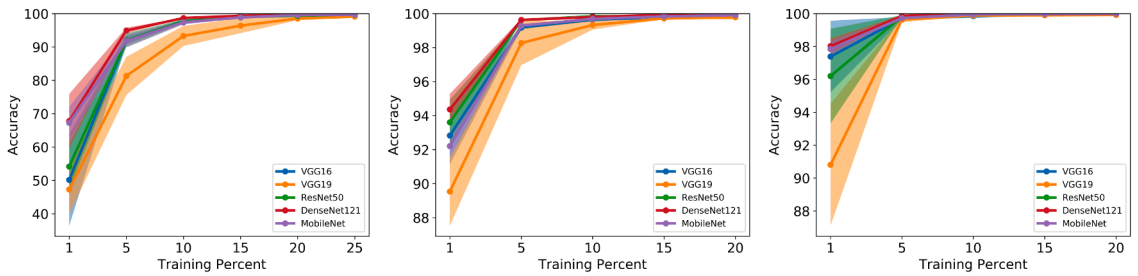
Table 12
Number of samples that the AL-based MLR, CNN1D, CNN2D and CNN3D need to reach a given % of OA for the IP and SV datasets.

Algorithm	Overall Accuracy						
	70%	75%	80%	85%	90%	95%	99%
				IP dataset			
AL-MLR	342	522	–	–	–	–	–
AL-CNN1D	252	352	502	662	–	–	–
AL-CNN2D	222	252	292	352	402	512	662
AL-CNN3D	72	82	112	152	172	232	402
				SV dataset			
AL-MLR	32	32	52	132	412	–	–
AL-CNN1D	32	32	42	62	232	–	–
AL-CNN2D	72	92	122	162	272	412	622
AL-CNN3D	32	32	32	52	72	112	292

selected around the training pixels. Some works (Hänsch et al., 2017) point out that the random sampling strategy has a great influence on the reliability and quality of the obtained solution, because this may significantly facilitate the subsequent classification of the test samples during the inference stage (as they have been previously processed in some way by the network during the training step). As a result, the performance obtained by the model may not be realistic, as artificially optimistic results can be obtained. In order to avoid this important issue, several works (Zhou et al., 2015; Hänsch et al., 2017; Liang et al., 2017; Lange et al., 2018) support the strict spatial-separation between train and test sets, allowing for the acquisition of more realistic accuracy results and a more accurate measurement of the real generalization-power of the model.

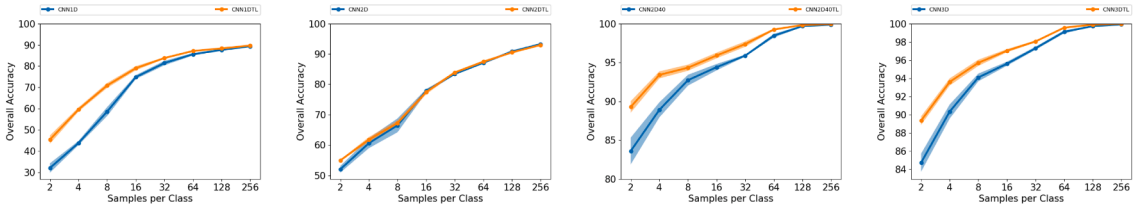
In this context, the aim of this experiment is to compare the results obtained by the spectral (RF, MLR, SVM, MLP, RNN, GRU, LSTM and CNN1D), spatial (CNN2D), and spectral-spatial (CNN2D40 and CNN3D) methods using, on the one hand, the traditional random sampling technique adopted by the methods discussed before in this paper and, on the other hand, a sampling strategy based on selecting spatially separated samples. To pursue this, spatially disjoint training and test sets for the IP and UP datasets (available from the GRSS DASE website (http://dase.grss-ieee.org) have been considered, as depicted on Figs. 18 and 19. For spatial and spectral-spatial methods, neighboring regions of $19 \times 19 \times n_{channels}$ have been cropped from the scenes, setting $n_{channels}$ to 1 and 40 spectral bands for spatial and spectral-spatial methods, respectively, while the spectral-based methods only process the original spectral pixels. The obtained results (in terms of OA) are reported on Table 13. As we can observe, there is a significant performance gap between the obtained results considering randomly selected training samples and spatially disjoint training samples, not only in the spatial and spectral-spatial methods [which is relatively expected due to the aforementioned aspects, as demonstrated by Ham et al. (2005)), but also on purely spectral-based models (which are not really affected by spatial correlations).

Focusing on the IP dataset, it can be noticed that the CNN2D, CNN3D, CNN2D40 and RF are the methods that suffer the most from this phenomenon. While the spectral-spatial methods' performance is significantly affected (reaching OA results in line with those obtained for the UH dataset in the first experiment, whose training and testing samples are spread over a larger area, preventing the possible overlapping effects between the test and train sets), the spectral-based RF is also suffering from a drastic performance drop due to another factor: the suitability of the selected samples. As it was pointed out on Section 2.2, HSI scenes generally suffer from high intraclass variability and interclass similarity, resulting from uncontrolled phenomena such as variations in illumination, presence of areas shaded and/or covered by clouds, and noise distortions, among others. In this sense, the selection of training samples must be carried out very carefully, to avoid situations in which the training and testing samples that belong to the same



(a) IP dataset (b) UP dataset (c) SV dataset

Fig. 16. OA evolution (y-axis) of each considered TL-based classifier with different training percentages (x-axis) over IP, UP and SV datasets. Standard deviation is showed as shaded areas.



(a) TL on CNN1D (b) TL on CNN2D (b) TL on CNN2D40 (c) TL on CNN3D

Fig. 17. Transfer learning experiment from IP to SV datasets, employing (from left to right) the CNN1D, CNN2D, CNND240 and CNN3D models of Table 5.

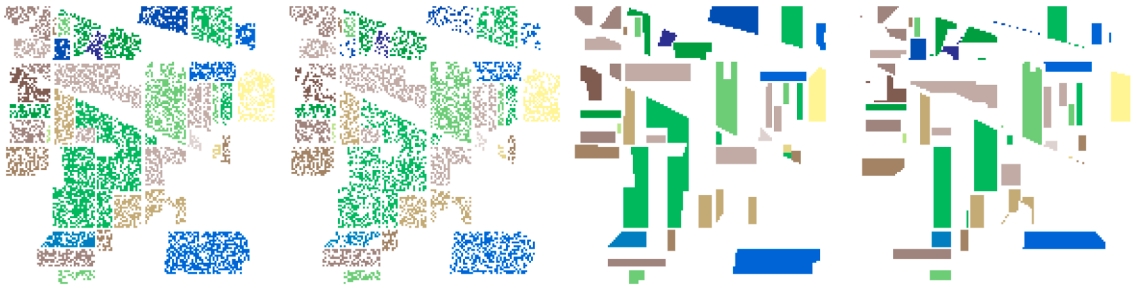
class could be spectrally quite different due to the presence of shadows or noise, for instance. This particularly affects spatial and spectral-spatial convolutional networks (Su et al., 2019).

Focusing on the UP dataset, the same gap between models trained and tested with randomly selected and spatially disjoint samples can be observed. Here, all methods, including the spectral ones (except CNN1D), reduce their OA values in more than 10 percentage points when spatially disjoint training samples are used. In particular, the CNN2D is the most significantly affected method, followed by the RF method. In this sense, it can be concluded that spatial and spectral-spatial methods are significantly affected by the spatial correlation between the training and test sets, which calls for the development of advanced sampling strategies to properly address the high variability of HSI data.

8. Conclusions and future lines

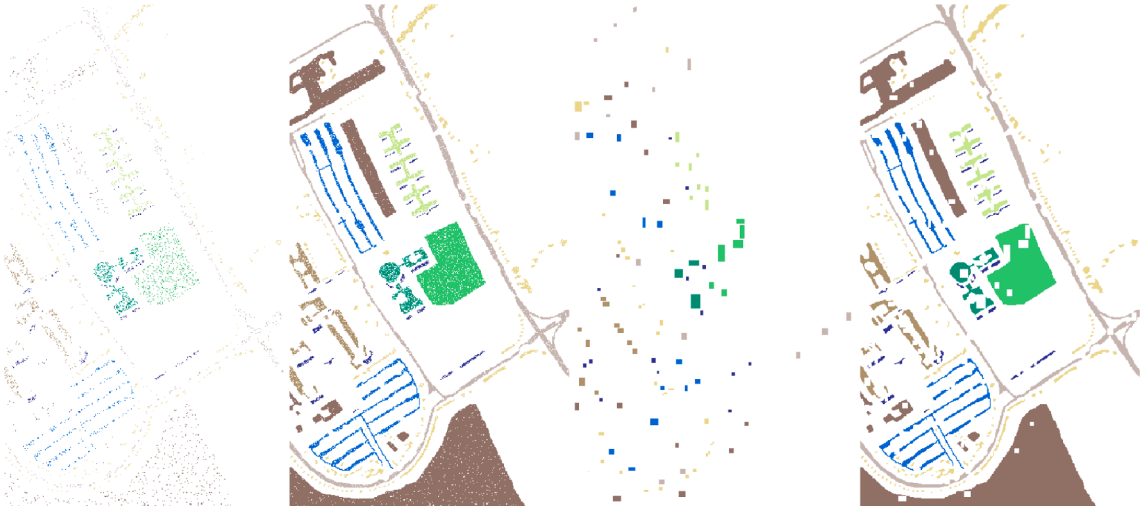
DL methods have revolutionized image analysis and proved to be a powerful tool for processing high-dimensional remotely sensed data, adapting their behavior to the special characteristics of HSI data. In this paper, we have provided an exhaustive review of DL models in the HSI arena. Models based on the CNN architecture have been found to be particularly effective, due to their capacity to extract highly discriminatory features and effectively leverage the spatial-contextual and spectral information contained in HSI data cubes. Traditional and hierarchical structures, composed by chains of blocks concatenated one after another, demonstrate a great generalization power that can be improved through new connections and paths. In fact, the use of standardization techniques, together with the reusability of the information contained in HSI data via residual connections (such as ResNet and DenseNet) and the concatenation of different paths, such as inception modules, have allowed to overcome important problems such as overfitting and the vanishing gradient when few training samples are available, or when very deep structures are implemented. Also, techniques such as AL and TL can help to improve the final performance of very deep neural models in training scenarios dominated by limited training samples, by employing semi-supervised strategies and pre-trained models. In the latter case, additional efforts need to be made in order to perform a more adequate training and adapt the available networks to the special characteristics of HSI data.

One of the main aspects preventing the full adaptation of the discussed paradigms to practical problems is that most of the considered models are highly demanding in computational terms, particularly when applied to complex HSI scenes. However, advances in computer technology and hardware platforms are rapidly allowing to increment the complexity and depth of the networks, making the required fine-tuning processes feasible in a reasonable amount of time. In this sense, there have been several efforts in the field of hardware accelerators that have made possible to implement deep models into embedded processors, GPUs and field programmable gate arrays (FPGAs), which can effectively parallelize the workload of DL-based networks (Randhe et al., 2016; Dong et al., 2017; Zhao et al., 2017b; Haut et al., 2018a). In addition, some research efforts are being carried out to distribute such high computational workloads among various cores using big data strategies, in particular, cloud computing techniques offer great flexibility and scalability, leading to a natural solution for the management of large and complex data HSI datasets. In this regard, we note that more efforts are needed in the remote sensing community in order to deploy cloud computing models, although there are already some works dealing with the exploitation of processing algorithms on cloud architectures (Wu et al., 2016; Haut et al., 2017a; Haut et al., 2017b; Quirita et al., 2017; Haut et al., 2019b). In summary, HPC is an attractive future research direction which can provide efficient mechanisms to address the enormous computational requirements introduced by DL-based HSI data processing, since the acquisition ratios of imaging spectrometers and the volume of future available repositories are expected to be extremely large (Bioucas-Dias et al., 2013), calling for the implementation of complex but faster and more efficient DL-based architectures. Last but not least, another important aspect worth being investigated in future developments is the design of new sample selection methods able to avoid any overlapping between the training and the testing set due to the patch size used by spatial-based methods in the training stage.



(a) Random Train (b) Random Test (c) Disjoint Train (d) Disjoint Test

Fig. 18. Comparison between the random selection method and the selection of spatially disjoint samples on the IP dataset, considering the same number of samples per class.



(a) Random Train (b) Random Test (c) Disjoint Train (d) Disjoint Test

Fig. 19. Comparison between the random selection method and the selection of spatially disjoint samples on the UP scene, considering the same number of samples per class.

Table 13
Comparison (in terms of OA) between different HSI classification models trained using randomly selected samples and spatial-disjoint samples during the training and inference stages.

	INDIAN PINES			PAVIA UNIVERSITY		
	Disjoint	Random	Diff	Disjoint	Random	Diff
RF	65.79	80.31	14.52	69.64	85.81	16.17
MLR	78.22	83.15	4.93	72.23	86.75	14.52
SVM	85.08	90.56	5.48	77.80	92.36	14.56
MLP	83.81	90.61	6.8	81.96	93.06	11.1
RNN	79.40	86.98	7.58	76.77	91.19	14,42
GRU	83.21	90.28	7.07	81.47	92.53	11.06
LSTM	82.94	90.76	7.82	79.54	91.09	11.55
CNN1D	84.94	91.96	7.02	87.59	94.15	6.56
CNN2D	56.66	99.77	43.11	76.47	98.22	21.75
CNN2D40	82.97	99.99	17.02	84.98	99.94	14.96
CNN3D	79.58	99.96	20.38	86.82	99.95	13.13

Acknowledgements

This work has been suported by:

- Spanish Ministerio de Educación (Resolución de 26 de diciembre de 2014 y de 19 de noviembre de 2015, de la Secretaría de Estado de Educación, Formación Profesional y Universidades, por la que se convocan ayudas para la formación de profesorado universitario, de los subprogramas de Formación y de Movilidad incluidos en el Programa Estatal de Promoción del Talento y su Empleabilidad, en el marco del Plan Estatal de Investigación Científica y Técnica y de Innovación 2013–2016.
- Junta de Extremadura (Decreto 14/2018, de 6 de febrero, por el que se establecen las bases reguladoras de las ayudas para la realización de actividades de investigación y desarrollo tecnológico, de divulgación y de transferencia de conocimiento por los Grupos de Investigación de Extremadura, Ref. GR18060).
- European Union's Horizon 2020 research and innovation programme under grant agreement No. 734541 (EOXPOSURE).

The authors would like to gratefully thank the Associate Editor and the two Anonymous Reviewers for their outstanding comments and suggestions, which greatly helped us to improve the technical quality and presentation of the manuscript.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., et al., 2016a. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467.

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al., 2016b. Tensorflow: A system for large-scale machine learning. In: 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16). pp. 265–283.

Abbate, G., Fiumi, L., Lorenzo, C.D., Vintila, R., May 2003. Evaluation of remote sensing data for urban planning. applicative examples by means of multispectral and hyperspectral data. In: 2003 2nd GRSS/ISPRS Joint Workshop on Remote Sensing and Data Fusion over Urban Areas. pp. 201–205.

Ablin, R., Sulochana, C.H., 2013. A survey of hyperspectral image classification in remote sensing. Int. J. Adv. Res. Comput. Commun. Eng. 2 (8), 2986–3000.

Acosta, I.C.C., Khodadadzadeh, M., Tusa, L., Ghamisi, P., Gloaguen, R., 2019. A machine learning framework for drill-core mineral mapping using hyperspectral and high-resolution mineralogical data fusion. IEEE J. Sel. Top. Appl. Earth Obser. Remote Sens.

Acquarelli, J., Marchiori, E., Buydens, L., Tran, T., Laarhoven, T., 2018. Spectral-spatial classification of hyperspectral images: Three tricks and a new learning setting. Remote Sensing 10 (7), 1156.

Agostinelli, F., Hoffman, M.D., Sadowski, P.J., Baldi, P., 2014. Learning activation functions to improve deep neural networks. arXiv preprint arXiv:1412.6830. http://arxiv.org/abs/1412.6830.

Ahmad, M., Protasov, S., Khan, A.M., 2017. Hyperspectral band selection using unsupervised non-linear deep auto encoder to train external classifiers. CoRR abs/1705.06920. URL http://arxiv.org/abs/1705.06920.

Al-khafaji, S.L., Zhou, J., Zia, A., Liew, A.W., 2018. Spectral-spatial scale invariant feature transform for hyperspectral images. IEEE Trans. Image Process. 27 (2), 837–850.

Anand, R., Veni, S., Aravinth, J., 2017. Big data challenges in airborne hyperspectral image for urban landuse classification. In: 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 1808–1814.

Anwer, R.M., Khan, F.S., van de Weijer, J., Molinier, M., Laaksonen, J., 2018. Binary patterns encoded convolutional neural networks for texture recognition and remote sensing scene classification. ISPRS J. Photogramm. Remote Sensing 138, 74–85.

Aptoula, E., Ozdemir, M.C., Yanikoglu, B., 2016. Deep learning with attribute profiles for hyperspectral image classification. IEEE Geosci. Remote Sens. Lett. 13 (12), 1970–1974.

Ardouin, J.P., Levesque, J., Rea, T.A., 2007. A demonstration of hyperspectral image exploitation for military applications. In: 2007 10th International Conference on Information Fusion, pp. 1–8.

Aslett, Z., Taranik, J.V., Riley, D.N., 2018. Mapping rock forming minerals at boundary canyon, death valey national park, california, using aerial sebass thermal infrared hyperspectral image data. Int. J. Appl. Earth Obs. Geoinf. 64, 326–339.

Audebert, N., Le Saux, B., Lefèvre, S., 2019. Deep learning for classification of hyperspectral data: A comparative review. IEEE Geosci. Remote Sens. Mag. 7 (2), 159–173.

Ba, J., Frey, B., 2013. Adaptive dropout for training deep neural networks. In: Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q. (Eds.), Advances in Neural Information Processing Systems 26. Curran Associates, Inc., pp. 3084–3092.

Ba, J.L., Kiros, J.R., Hinton, G.E., 2016. Layer normalization. arXiv preprint arXiv:1607.06450.

Babey, S., Anger, C., 1989. A compact airborne spectrographic imager (casi). In: Quantitative Remote Sensing: An Economic Tool for the Nineties, vol. 1. pp. 1028–1031.

Bach, F., 2017. Breaking the curse of dimensionality with convex neural networks. J. Machine Learn. Res. 18 (19), 1–53.
Baldi, P., Hornik, K., 1989. Neural networks and principal component analysis: Learning from examples without local minima. Neural Networks 2 (1), 53–58.
Ball, J.E., Anderson, D.T., Chan, C.S., 2017. Comprehensive survey of deep learning in remote sensing: theories, tools, and challenges for the community. J. Appl. Remote Sens. 11, 11–54.
Bannari, A., Pacheco, A., Staenz, K., McNairn, H., Omari, K., 2006. Estimating and mapping crop residues cover on agricultural lands using hyperspectral and ikonos data. Remote Sens. Environ. 104 (4), 447–459.
Bellman, R., 2015. Adaptive Control Processes: A Guided Tour. Princeton Legacy Library. Princeton University Press.
Benediktsson, J.A., Sveinsson, J.R., 2003. Multisource remote sensing data classification based on consensus and pruning. IEEE Trans. Geosci. Remote Sens. 41 (4), 932–936.
Benediktsson, J.A., Swain, P.H., Ersoy, O.K., 1993. Conjugate-gradient neural networks in classification of multisource and very-high-dimensional remote sensing data. Int. J. Remote Sens. 14 (15), 2883–2903.
Bengio, Y., 2009. Learning deep architectures for ai. Found. Trends Machine Learn. 2 (1), 1–127.
Bengio, Y., Courville, A., Vincent, P., 2013. Representation learning: A review and new perspectives. IEEE Trans. Pattern Anal. Machine Intell. 35 (8), 1798–1828.
Bengio, Y., Courville, A.C., Vincent, P., 2012. Unsupervised feature learning and deep learning: A review and new perspectives. CoRR, abs/1206.5538 1, 2012.
Bengio, Y., Lamblin, P., Popovici, D., Larochelle, H., 2007a. Greedy layer-wise training of deep networks. In: Schölkopf, B., Platt, J.C., Hoffman, T. (Eds.), Advances in Neural Information Processing Systems 19. MIT Press, pp. 153–160.
Bengio, Y., LeCun, Y., et al., 2007b. Scaling learning algorithms towards ai. Large-scale Kernel Machines 34 (5), 1–41.
Benítez, J.M., Castro, J.L., Requena, I., 1997. Are artificial neural networks black boxes? IEEE Trans. Neural Networks 8 (5), 1156–1164.
Bhardwaj, K., Patra, S., 2018. An unsupervised technique for optimal feature selection in attribute profiles for spectral-spatial classification of hyperspectral images. ISPRS J. Photogramm. Remote Sens. 138, 139–150.
Bioucas-Dias, J.M., Plaza, A., Camps-Valls, G., Scheunders, P., Nasrabadi, N., Chanussot, J., 2013. Hyperspectral remote sensing data analysis and future challenges. IEEE Geosci. Remote Sens. Mag. 1 (2), 6–36.
Bioucas-Dias, J.M., Plaza, A., Dobigeon, N., Parente, M., Du, Q., Gader, P., Chanussot, J., 2012. Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 5 (2), 354–379.
Bishop, C., 1995. Neural Networks for Pattern Recognition. Advanced Texts in Econometrics. Clarendon Press.
Bjorck, N., Gomes, C.P., Selman, B., Weinberger, K.Q., 2018. Understanding batch normalization. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (Eds.), Advances in Neural Information Processing Systems 31. Curran Associates, Inc., pp. 7694–7705.
Blum, A., Rivest, R.L., 1989. Training a 3-node neural network is np-complete. In: Advances in Neural Information Processing Systems. pp. 494–501.
Boureau, Y.-L., Ponce, J., LeCun, Y., 2010. A theoretical analysis of feature pooling in visual recognition. In: Proceedings of the 27th International Conference on Machine Learning (ICML-10). pp. 111–118.
Brendel, W., Bethge, M., 2019. Approximating CNNs with bag-of-local-features models works surprisingly well on imagenet. In: International Conference on Learning Representations, pp. 15.
Briottet, X., Boucher, Y., Dimmeler, A., Malaplate, A., Cini, A., Diani, M., Bekman, H., Schwering, P., Skauli, T., Kasen, I., et al., 2006. Military applications of hyperspectral imagery. In: Targets and backgrounds XII: Characterization and representation. Vol. 6239. International Society for Optics and Photonics, p. 62390B.
Bruce, L.M., Koger, C.H., Li, J., 2002. Dimensionality reduction of hyperspectral data using discrete wavelet transform feature extraction. IEEE Trans. Geosci. Remote Sensing 40 (10), 2331–2338.
Büchel, J., Ersoy, O., 2018. Ladder networks for semi-supervised hyperspectral image classification. arXiv preprint arXiv:1812.01222.
Bue, B.D., Thompson, D.R., Eastwood, M., Green, R.O., Gao, B.C., Keymeulen, D., Sarture, C.M., Mazer, A.S., Luong, H.H., 2015. Real-time atmospheric correction of aviris-ng imagery. IEEE Trans. Geosci. Remote Sens. 53 (12), 6419–6428.
Calin, M.A., Parasca, S.V., Manea, D., 2018. Comparison of spectral angle mapper and support vector machine classification methods for mapping skin burn using hyperspectral imaging. In: Unconventional Optical Imaging. Vol. 10677. International Society for Optics and Photonics, p. 106773P.
Camps-Valls, G., 2016. Kernel spectral angle mapper. Electron. Lett. 52 (14), 1218–1220.
Camps-Valls, G., Gomez-Chova, L., Muñoz-Marí, J., Vila-Francés, J., Calpe-Maravilla, J., 2006. Composite kernels for hyperspectral image classification. IEEE Geosci. Remote Sens. Letters 3 (1), 93–97.
Camps-Valls, G., Tuia, D., Bruzzone, L., Benediktsson, J.A., 2014. Advances in hyperspectral image classification: Earth monitoring with statistical learning methods. IEEE Signal Process. Mag. 31 (1), 45–54.
Cao, X., Zhou, F., Xu, L., Meng, D., Xu, Z., Paisley, J., 2018. Hyperspectral image classification with markov random fields and a convolutional neural network. IEEE Trans. Image Process. 27 (5), 2354–2367.
Cariou, C., Chehdi, K., 2015. Unsupervised nearest neighbors clustering with application to hyperspectral images. IEEE J. Sel. Top. Signal Process. 9 (6), 1105–1116.
Caruana, R., Lawrence, S., Giles, C.L., 2001. Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. In: Advances in Neural Information Processing Systems. pp. 402–408.
Chabrillat, S., Milewski, R., Schmid, T., Rodriguez, M., Escribano, P., Pelayo, M., Palacios-Orueta, A., July 2014. Potential of hyperspectral imagery for the spatial assessment of soil erosion stages in agricultural semi-arid spain at different scales. In: 2014 IEEE Geoscience and Remote Sensing Symposium. pp. 2918–2921.
Chang, C.-I., 2007. Hyperspectral Data Exploitation: Theory and Applications. John Wiley & Sons.
Charles, A.S., Olshausen, B.A., Rozell, C.J., 2011. Learning sparse codes for hyperspectral imagery. IEEE J. Sel. Top. Signal Process. 5 (5), 963–978.
Charmisha, K., Sowmya, V., Soman, K., 2018. Dimensionally reduced features for hyperspectral image classification using deep learning. In: International Conference on Communications and Cyber Physical Engineering 2018. Springer, pp. 171–179.
Chen, G., Qian, S.-E., 2011. Denoising of hyperspectral imagery using principal component analysis and wavelet shrinkage. IEEE Trans. Geosci. Remote Sens. 49 (3), 973–980.
Chen, S., Wang, Y., 2014. Convolutional neural network and convex optimization. Dept. of Elect. and Comput. Eng., Univ. of California at San Diego, San Diego, CA, USA, Tech. Rep.
Chen, X., Xiang, S., Liu, C.-L., Pan, C.-H., 2014a. Vehicle Detection in Satellite Images by Hybrid Deep Convolutional Neural Networks. IEEE Geosci. Remote Sens. Lett. 11 (10), 1797–1801.
Chen, Y., Jiang, H., Li, C., Jia, X., Ghamisi, P., 2016. Deep feature extraction and classification of hyperspectral images based on convolutional neural networks. IEEE Trans. Geosci. Remote Sens. 54 (10), 6232–6251.
Chen, Y., Li, J., Xiao, H., Jin, X., Yan, S., Feng, J., 2017a. Dual path networks. In: Advances in Neural Information Processing Systems. pp. 4467–4475.
Chen, Y., Lin, Z., Zhao, X., Wang, G., Gu, Y., 2014b. Deep Learning-Based Classification of Hyperspectral Data. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 7 (6), 2094–2107.
Chen, Y., Wang, Y., Gu, Y., He, X., Ghamisi, P., Jia, X., 2019a. Deep learning ensemble for hyperspectral image classification. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.
Chen, Y., Zhao, X., Jia, X., 2015. Spectral-Spatial Classification of Hyperspectral Data Based on Deep Belief Network. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 8 (6), 2381–2392.
Chen, Y., Zhu, K., Zhu, L., He, X., Ghamisi, P., Benediktsson, J.A., 2019b. Automatic design of convolutional neural network for hyperspectral image classification. IEEE Trans. Geosci. Remote Sens.
Chen, Y., Zhu, L., Ghamisi, P., Jia, X., Li, G., Tang, L., 2017b. Hyperspectral images classification with gabor filtering and convolutional neural network. IEEE Geosci. Remote Sens. Lett. 14 (12), 2355–2359.
Cheng, G., Han, J., Lu, X., 2017a. Remote sensing image scene classification: Benchmark and state of the art. Proc. IEEE 105 (10), 1865–1883.
Cheng, Y., Wang, D., Zhou, P., Zhang, T., 2017b. A survey of model compression and acceleration for deep neural networks. arXiv preprint arXiv:1710.09282.
Cho, K., Van Merriënboer, B., Bahdanau, D., Bengio, Y., 2014. On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259.
Chollet, F., 2017. Xception: Deep learning with depthwise separable convolutions. arXiv preprint, 1610–02357.
Chutia, D., Bhattacharyya, D., Sarma, K.K., Kalita, R., Sudhakar, S., 2016. Hyperspectral remote sensing classifications: a perspective survey. Trans. GIS 20 (4), 463–490.
Cocks, T., Jenssen, R., Stewart, A., Wilson, I., Shields, T., 1998. The hymaptm airborne hyperspectral sensor: the system, calibration and performance. In: Proceedings of the 1st EARSeL workshop on Imaging Spectroscopy. EARSeL, pp. 37–42.
Collobert, R., Bengio, S., 2004. Links between perceptrons, mlps and svms. In: Proceedings of the Twenty-first International Conference on Machine Learning. ACM, pp. 23.
Collobert, R., Bengio, S., Mariéthoz, J., 2002. Torch: a modular machine learning software library. Tech. Rep., Idiap.
Coops, N.C., Smith, M.L., Martin, M.E., Ollinger, S.V., 2003. Prediction of eucalypt foliage nitrogen content from satellite-derived hyperspectral data. IEEE Trans. Geosci. Remote Sens. 41 (6), 1338–1346.
Cybenko, G., 1989. Approximation by superpositions of a sigmoidal function. Math. Control, Signals Syst. 2 (4), 303–314.
Dalla Mura, M., Villa, A., Benediktsson, J.A., Chanussot, J., Bruzzone, L., 2011. Classification of hyperspectral images by using extended morphological attribute profiles and independent component analysis. IEEE Geosci. Remote Sens. Lett. 8 (3), 542–546.
Dauphin, Y.N., Pascanu, R., Gulcehre, C., Cho, K., Ganguli, S., Bengio, Y., 2014. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In: Advances in Neural Information Processing Systems. pp. 2933–2941.
Debes, C., Merentitis, A., Heremans, R., Hahn, J., Frangiadakis, N., van Kasteren, T., Liao, W., Bellens, R., Pižurica, A., Gautama, S., Philips, W., Prasad, S., Du, Q., Pacifici, F., 2014. Hyperspectral and lidar data fusion: Outcome of the 2013 grss data fusion contest. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 7 (6), 2405–2418.
Defferrard, M., Bresson, X., Vandergheynst, P., 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In: Lee, D.D., Sugiyama, M., Luxburg, U.V., Guyon, I., Garnett, R. (Eds.), Advances in Neural Information Processing Systems 29. Curran Associates, Inc., pp. 3844–3852.
Deng, C., Xue, Y., Liu, X., Li, C., Tao, D., 2019. Active transfer learning network: A unified deep joint spectral–spatial feature learning model for hyperspectral image classification. IEEE Trans. Geosci. Remote Sens. 57 (3), 1741–1754.
Deng, F., Pu, S., Chen, X., Shi, Y., Yuan, T., Pu, S., 2018. Hyperspectral image classification with capsule network using limited training samples. Sensors 18 (9).
Ding, C., Li, Y., Xia, Y., Wei, W., Zhang, L., Zhang, Y., 2017. Convolutional neural networks based hyperspectral image classification method with adaptive kernels. Remote Sens. 9 (6), 618.
Dong, H., Li, T., Leng, J., Kong, L., Bai, G., 2017. Gcn: Gpu-based cube cnn framework for

hyperspectral image classification. In: 2017 46th International Conference on Parallel Processing (ICPP), pp. 41–49.
Dong, H., Zhang, L., Zou, B., 2019. Band attention convolutional networks for hyperspectral image classification. arXiv preprint arXiv:1906.04379.
Du, J., Li, Z., 2018. A hyperspectral target detection framework with subtraction pixel pair features. IEEE Access 6, 45562–45577.
Du, Q., Chang, C.-I., 2001. A linear constrained distance-based discriminant analysis for hyperspectral image classification. Pattern Recogn. 34 (2), 361–373.
Duchi, J., Hazan, E., Singer, Y., 2011. Adaptive subgradient methods for online learning and stochastic optimization. J. Machine Learn. Res. 12 (Jul), 2121–2159.
Dugas, C., Bengio, Y., Bélisle, F., Nadeau, C., Garcia, R., 2001. Incorporating second-order functional knowledge for better option pricing. In: Advances in Neural Information Processing Systems. pp. 472–478.
Dumke, I., Nornes, S.M., Purser, A., Marcon, Y., Ludvigsen, M., Ellefmo, S.L., Johnsen, G., Søreide, F., 2018. First hyperspectral imaging survey of the deep seafloor: High-resolution mapping of manganese nodules. Remote Sens. Environ. 209, 19–30.
Eckardt, A., Horack, J., Lehmann, F., Krutz, D., Drescher, J., Whorton, M., Soutullo, M., 2015. Desis (dlr earth sensing imaging spectrometer for the iss-muses platform). In: 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS). IEEE, pp. 1457–1459.
Eismann, M.T., Hardie, R.C., 2005. Hyperspectral resolution enhancement using high-resolution multispectral imagery with arbitrary response functions. IEEE Trans. Geosci. Remote Sens. 43 (3), 455–465.
El-Magd, I.A., El-Zeiny, A., 2014. Quantitative hyperspectral analysis for characterization of the coastal water from damietta to port said, egypt. Egyp. J. Remote Sens. Space Sci. 17 (1), 61–76.
El-Sharkawy, Y.H., Elbasuney, S., 2019. Hyperspectral imaging: Anew prospective for remote recognition of explosive materials. Remote Sensing Appl.: Soc. Environ. 13, 31–38.
Erhan, D., Bengio, Y., Courville, A., Manzagol, P.-A., Vincent, P., Bengio, S., 2010. Why does unsupervised pre-training help deep learning? J. Machine Learn. Res. 11 (Feb), 625–660.
Fang, B., Li, Y., Zhang, H., Chan, J., 2018. Semi-supervised deep learning classification for hyperspectral image based on dual-strategy sample selection. Remote Sens. 10 (4), 574.
Fang, B., Li, Y., Zhang, H., Chan, J.C.-W., 2019. Hyperspectral images classification based on dense convolutional networks with spectral-wise attention mechanism. Remote Sens. 11 (2), 159.
Fauvel, M., Benediktsson, J.A., Chanussot, J., Sveinsson, J.R., 2008. Spectral and spatial classification of hyperspectral data using svms and morphological profiles. IEEE Trans. Geosci. Remote Sens. 46 (11), 3804–3814.
Fauvel, M., Tarabalka, Y., Benediktsson, J.A., Chanussot, J., Tilton, J.C., 2013. Advances in spectral-spatial classification of hyperspectral images. Proc. IEEE 101 (3), 652–675.
Feingersh, T., Dor, E.B., 2015. Shalom–a commercial hyperspectral space mission. Opt. Payloads Space Missions 247–263.
Feng, W., Qi, S., Heng, Y., Zhou, Y., Wu, Y., Liu, W., He, L., Li, X., 2017. Canopy vegetation indices from in situ hyperspectral data to assess plant water status of winter wheat under powdery mildew stress. Front. Plant Sci. 8 (1219).
Fernandez, D., Gonzalez, C., Mozos, D., Lopez, S., 2016. Fpga implementation of the principal component analysis algorithm for dimensionality reduction of hyperspectral images. J. Real-Time Image Proc. 1–12.
Fey, M., Lenssen, J.E., 2019. Fast graph representation learning with pytorch geometric. arXiv preprint arXiv:1903.02428.
Field, D.J., 1999. Wavelets, vision and the statistics of natural scenes. Philosoph. Trans. Roy. Soc. London A: Math., Phys. Eng. Sci. 357 (1760), 2527–2542.
Fisher, P., 1997. The pixel: a snare and a delusion. Int. J. Remote Sens. 18 (3), 679–685.
Galeazzi, C., Sacchetti, A., Cisbani, A., Babini, G., 2008. The prisma program. In: Geoscience and Remote Sensing Symposium, 2008. IGARSS 2008. IEEE International, vol. 4. IEEE, pp. IV–105.
Gao, H., Lin, S., Yang, Y., Li, C., Yang, M., 2018a. Convolution neural network based on two-dimensional spectrum for hyperspectral image classification. J. Sensors 2018, 13.
Gao, Q., Lim, S., Jia, X., 2018b. Hyperspectral image classification using convolutional neural networks and multiple feature learning. Remote Sens. 10 (2), 299.
Ghamisi, P., Chen, Y., Zhu, X.X., 2016. A self-improving convolution neural network for the classification of hyperspectral data. IEEE Geosci. Remote Sens. Lett. 13 (10), 1537–1541.
Ghamisi, P., Maggiori, E., Li, S., Souza, R., Tarablaka, Y., Moser, G., De Giorgi, A., Fang, L., Chen, Y., Chi, M., et al., 2018. New frontiers in spectral-spatial hyperspectral image classification: The latest advances based on mathematical morphology, markov random fields, segmentation, sparse representation, and deep learning. IEEE Geosci. Remote Sens. Mag. 6 (3), 10–43.
Ghamisi, P., Plaza, J., Chen, Y., Li, J., Plaza, A.J., 2017a. Advanced spectral classifiers for hyperspectral images: A review. IEEE Geosci. Remote Sens. Mag. 5 (1), 8–32.
Ghamisi, P., Yokoya, N., Li, J., Liao, W., Liu, S., Plaza, J., Rasti, B., Plaza, A., 2017b. Advances in hyperspectral image and signal processing: A comprehensive overview of the state of the art. IEEE Geosci. Remote Sens. Mag. 5 (4), 37–78.
Ghiasi, G., Lin, T.-Y., Le, Q.V., 2018. Dropblock: A regularization method for convolutional networks. In: Advances in Neural Information Processing Systems. pp. 10750–10760.
Gitman, I., Ginsburg, B., 2017. Comparison of batch normalization and weight normalization algorithms for the large-scale image classification. arXiv preprint arXiv:1709.08145.
Glorot, X., Bengio, Y., 13–15 May 2010. Understanding the difficulty of training deep feedforward neural networks. In: Teh, Y.W., Titterington, M. (Eds.), Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. Vol. 9 of Proceedings of Machine Learning Research. PMLR, Chia Laguna Resort, Sardinia, Italy, pp. 249–256.
Goetz, A.F.H., Vane, G., Solomon, J.E., Rock, B.N., 1985. Imaging Spectrometry for Earth Remote Sensing. Science 228 (4704), 1147–1153.
Gomez, C., Drost, A., Roger, J.-M., 2015. Analysis of the uncertainties affecting predictions of clay contents from vnir/swir hyperspectral data. Remote Sens. Environ. 156, 58–70.
Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press.
Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative adversarial nets. In: Advances in Neural Information Processing Systems. pp. 2672–2680.
Green, A.A., Berman, M., Switzer, P., Craig, M.D., 1988. A transformation for ordering multispectral data in terms of image quality with implications for noise removal. IEEE Trans. Geosci. Remote Sens. 26 (1), 65–74.
Green, R.O., Eastwood, M.L., Sarture, C.M., Chrien, T.G., Aronsson, M., Chippendale, B.J., Faust, J.A., Pavri, B.E., Chovit, C.J., Solis, M., Olah, M.R., Williams, O., 1998. Imaging spectroscopy and the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS). Remote Sens. Environ. 65 (3), 227–248.
Große-Stoltenberg, A., Hellmann, C., Werner, C., Oldeland, J., Thiele, J., 2016. Evaluation of continuous vnir-swir spectra versus narrowband hyperspectral indices to discriminate the invasive acacia longifolia within a mediterranean dune ecosystem. Remote Sens. 8 (4).
Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J., et al., 2018. Recent advances in convolutional neural networks. Pattern Recogn. 77, 354–377.
Guanter, L., Kaufmann, H., Segl, K., Foerster, S., Rogass, C., Chabrillat, S., Kuester, T., Hollstein, A., Rossner, G., Chlebek, C., et al., 2015. The enmap spaceborne imaging spectroscopy mission for earth observation. Remote Sens. 7 (7), 8830–8857.
Guo, A.J., Zhu, F., 2018. A cnn-based spatial feature fusion algorithm for hyperspectral imagery classification. arXiv preprint arXiv:1801.10355.
Guo, Y., Han, S., Cao, H., Zhang, Y., Wang, Q., 2018. Guided filter based deep recurrent neural networks for hyperspectral image classification. Procedia Computer Science 129, 219–223, 2017 International Conference on Identification, Information and Knowledge in the Internet of Things.
Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., Lew, M.S., 2016. Deep learning for visual understanding: A review. Neurocomputing 187, 27–48 recent Developments on Deep Big Vision.
Guofeng, T., Yong, L., Lihao, C., Chen, J., June 2017. A dbn for hyperspectral remote sensing image classification. In: 2017 12th IEEE Conference on Industrial Electronics and Applications (ICIEA). pp. 1757–1762.
Haboudane, D., Miller, J.R., Pattey, E., Zarco-Tejada, P.J., Strachan, I.B., 2004. Hyperspectral vegetation indices and novel algorithms for predicting green lai of crop canopies: Modeling and validation in the context of precision agriculture. Remote Sens. Environ. 90 (3), 337–352.
Ham, J., Chen, Y., Crawford, M.M., Ghosh, J., 2005. Investigation of the random forest framework for classification of hyperspectral data. IEEE Trans. Geosci. Remote Sens. 43 (3), 492–501.
Han, Y., Li, J., Zhang, Y., Hong, Z., Wang, J., 2017. Sea ice detection based on an improved similarity measurement method using hyperspectral data. Sensors 17 (5), 1124.
Hänsch, R., Ley, A., Hellwich, O., 2017. Correct and still wrong: The relationship between sampling strategies and the estimation of the generalization error. In: 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS). IEEE, pp. 3672–3675.
Hao, S., Wang, W., Ye, Y., Nie, T., Bruzzone, L., 2018. Two-stream deep architecture for hyperspectral image classification. IEEE Trans. Geosci. Remote Sens. 56 (4), 2349–2361.
Hassanzadeh, A., Kaarna, A., Kauranne, T., 2017. Unsupervised multi-manifold classification of hyperspectral remote sensing images with contractive autoencoder. In: Sharma, P., Bianchi, F.M. (Eds.), Image Analysis. Springer International Publishing, Cham, pp. 169–180.
Hassanzadeh, A., Kaarna, A., Kauranne, T., 2018. Sequential spectral clustering of hyperspectral remote sensing image over bipartite graph. Appl. Soft Comput. 73, 727–734.
Haut, J., Paoletti, M., Paz-Gallardo, A., Plaza, J., Plaza, A., 2017a. Cloud implementation of logistic regression for hyperspectral image classification. In: Vigo-Aguiar, J. (Ed.), Proceedings of the 17th International Conference on Computational and Mathematical Methods in Science and Engineering, CMMSE 2017. Costa Ballena (Rota), Cádiz, Spain, pp. 1063–2321.
Haut, J., Paoletti, M., Plaza, J., Plaza, A., 2017b. Cloud implementation of the K-means algorithm for hyperspectral image analysis. J. Supercomput. 73 (1).
Haut, J., Paoletti, M., Plaza, J., Plaza, A., 2019a. Hyperspectral image classification using random occlusion data augmentation. IEEE Geosci. Remote Sens. Lett.
Haut, J.M., Bernabé, S., Paoletti, M.E., Fernandez-Beltran, R., Plaza, A., Plaza, J., 2018a. Low–high-power consumption architectures for deep-learning models applied to hyperspectral image classification. IEEE Geosci. Remote Sens. Lett. 16 (5), 776–780.
Haut, J.M., Fernandez-Beltran, R., Paoletti, M.E., Plaza, J., Plaza, A., Pla, F., 2018b. A new deep generative network for unsupervised remote sensing single-image super-resolution. IEEE Trans. Geosci. Remote Sens. 1–19.
Haut, J.M., Gallardo, J.A., Paoletti, M.E., Cavallaro, G., Plaza, J., Plaza, A., Riedel, M., 2019b. Cloud deep networks for hyperspectral image analysis. IEEE Trans. Geosci. Remote Sens.
Haut, J.M., Paoletti, M.E., Plaza, J., Li, J., Plaza, A., 2018c. Active learning with convolutional neural networks for hyperspectral image classification using a new bayesian approach. IEEE Trans. Geosci. Remote Sens. 1–22.
Haut, J.M., Paoletti, M.E., Plaza, J., Plaza, A., 2018d. Fast dimensionality reduction and

classification of hyperspectral images with extreme learning machines. J. Real-Time Image Proc. 1–24.
Haut, J.M., Paoletti, M.E., Plaza, J., Plaza, A., Li, J., 2019. Visual attention-driven hyperspectral image classification. IEEE Trans. Geosci. Remote Sens. 1–16.
He, K., Sun, J., Tang, X., 2013. Guided image filtering. IEEE Trans. Pattern Anal. Machine Intell. 35 (6), 1397–1409.
He, K., Zhang, X., Ren, S., Sun, J., 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 1026–1034.
He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778.
He, L., Li, J., Plaza, A., Li, Y., 2017a. Discriminative low-rank gabor filtering for spectral-spatial hyperspectral image classification. IEEE Trans. Geosci. Remote Sens. 55 (3), 1381–1395.
He, N., Paoletti, M.E., Haut, J.n.M., Fang, L., Li, S., Plaza, A., Plaza, J., 2018. Feature extraction with multiscale covariance maps for hyperspectral image classification. IEEE Trans. Geosci. Remote Sens. 1–15.
He, X., Chen, Y., 2019. Optimized input for cnn-based hyperspectral image classification using spatial transformer network. IEEE Geosci. Remote Sens. Lett.
He, Z., Liu, H., Wang, Y., Hu, J., 2017b. Generative adversarial networks-based semi-supervised learning for hyperspectral image classification. Remote Sens. 9 (10), 1042.
Heldens, W., Heiden, U., Esch, T., Stein, E., Müller, A., 2011. Can the future enmap mission contribute to urban applications? a literature survey. Remote Sens. 3 (9), 1817–1846.
Heylen, R., Parente, M., Gader, P., 2014. A review of nonlinear hyperspectral unmixing methods. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 7 (6), 1844–1868.
Hinton, G., Salakhutdinov, R., 2006. Reducing the dimensionality of data with neural networks. Science 313, 504–507.
Hinton, G.E., Osindero, S., Teh, Y.-W., 2006. A fast learning algorithm for deep belief nets. Neural Comput. 18 (7), 1527–1554.
Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R., 2012. Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580.
Hinton, G.E., Zemel, R.S., 1993. Autoencoders, minimum description length and helmholtz free energy. In: Proceedings of the 6th International Conference on Neural Information Processing Systems. NIPS'93. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 3–10.
Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9 (8), 1735–1780.
Hornik, K., 1991. Approximation capabilities of multilayer feedforward networks. Neural Networks 4 (2), 251–257.
Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H., 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.
Hu, W., Huang, Y., Wei, L., Zhang, F., Li, H., 2015. Deep convolutional neural networks for hyperspectral image classification. J. Sensors.
Huadong, G., Jianmin, X., Guoqiang, N., Jialing, M., 2001. A new airborne earth observing system and its applications. In: IGARSS 2001. Scanning the Present and Resolving the Future. Proceedings. IEEE 2001 International Geoscience and Remote Sensing Symposium (Cat. No. 01CH37217). Vol. 1. pp. 549–551 vol 1.
Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q., 2017. Densely connected convolutional networks. In: CVPR. Vol. 1. p. 3.
Huang, X., Belongie, S., 2017. Arbitrary style transfer in real-time with adaptive instance normalization. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1501–1510.
Huang, X., Zhang, L., 2013. An svm ensemble approach combining spectral, structural, and semantic features for the classification of high-resolution remotely sensed imagery. IEEE Trans. Geosci. Remote Sens. 51 (1), 257–272.
Huangi, L., Huangi, L., Yang, D., Lang, B., Deng, J., 2018. Decorrelated batch normalization. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 791–800.
Hughes, G., 1968. On the mean accuracy of statistical pattern recognizers. IEEE Trans. Inf. Theory 14 (1), 55–63.
Ioffe, S., 2017. Batch renormalization: Towards reducing minibatch dependence in batch-normalized models. In: Advances in Neural Information Processing Systems. pp. 1945–1953.
Ioffe, S., Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167.
Iyer, R.P., Raveendran, A., Bhuvana, S.K.T., Kavitha, R., 2017. Hyperspectral image analysis techniques on remote sensing. In: 2017 Third International Conference on Sensing, Signal Processing and Security (ICSSS), pp. 392–396.
Jia, P., Zhang, M., Yu, W., Shen, F., Shen, Y., 2016. Convolutional neural network based classification for hyperspectral data. In: 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), pp. 5075–5078.
Jiao, L., Liang, M., Chen, H., Yang, S., Liu, H., Cao, X., 2017. Deep fully convolutional network-based spatial distribution prediction for hyperspectral image classification. IEEE Trans. Geosci. Remote Sens. 55 (10), 5585–5599.
Jiménez, L.O., Rivera-Medina, J.L., Rodríguez-Díaz, E., Arzuaga-Cruz, E., Ramírez-Vélez, M., 2005. Integration of spatial and spectral information by means of unsupervised extraction and classification for homogenous objects applied to multispectral and hyperspectral data. IEEE Trans. Geosci. Remote Sens. 43 (4), 844–851.
Jing, L., Tian, Y., 2019. Self-supervised visual feature learning with deep neural networks: A survey. arXiv preprint arXiv:1902.06162.
Jolliffe, I., 2002. Principal Component Analysis. Springer Series in Statistics. Springer.
Kallepalli, A., Kumar, A., Khoshelham, K., Nov. 2014. Entropy based determination of optimal principal components of Airborne Prism Experiment (APEX) imaging spectrometer data for improved land cover classification. ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 781–786.
Kang, X., Li, C., Li, S., Lin, H., 2018. Classification of hyperspectral images by gabor filtering based deep network. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 11 (4), 1166–1178.
Kang, X., Zhang, X., Li, S., Li, K., Li, J., Benediktsson, J.A., 2017. Hyperspectral anomaly detection with attribute and edge-preserving filters. IEEE Trans. Geosci. Remote Sens. 55 (10), 5600–5611.
Kang, X., Zhuo, B., Duan, P., 2018. Dual-path network-based hyperspectral image classification. IEEE Geosci. Remote Sens. Lett. 1–5.
Kang, X., Zhuo, B., Duan, P., 2019. Semi-supervised deep learning for hyperspectral image classification. Remote Sens. Lett. 10 (4), 353–362.
Karhunen, J., Raiko, T., Cho, K., 2015. Unsupervised Deep Learning: A Short Review.
Kaufmann, H., Segl, K., Guanter, L., Hofer, S., Foerster, K.-P., Stuffler, T., Mueller, A., Richter, R., Bach, H., Hostert, P., et al., 2008. Environmental mapping and analysis program (enmap)-recent advances and status. In: Geoscience and Remote Sensing Symposium, 2008. IGARSS 2008. IEEE International, vol. 4. IEEE, pp. IV–109.
Keshava, N., 2004. Distance metrics and band selection in hyperspectral processing with applications to material identification and spectral libraries. IEEE Tran. Geosci. Remote Sens. 42 (7), 1552–1565.
Kessy, A., Lewin, A., Strimmer, K., 2018. Optimal whitening and decorrelation. Am. Stat. 72 (4), 309–314.
Ketkar, N., 2017. Introduction to keras. In: Deep Learning with Python. Springer, pp. 97–111.
Khan, M.J., Khan, H.S., Yousaf, A., Khurshid, K., Abbas, A., 2018. Modern trends in hyperspectral image analysis: A review. IEEE Access 6, 14118–14129.
Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.
Klambauer, G., Unterthiner, T., Mayr, A., Hochreiter, S., 2017. Self-normalizing neural networks. In: Advances in Neural Information Processing Systems. pp. 971–980.
Koch, G., Zemel, R., Salakhutdinov, R., 2015. Siamese neural networks for one-shot image recognition. In: ICML Deep Learning Workshop. Vol. 2. p.
Kokaly, R.F., Hoefen, T.M., Graham, G.E., Kelley, K.D., Johnson, M.R., Hubbard, B.E., Goldfarb, R.J., Buchhorn, M., Prakash, A., 2016. Mineral information at micron to kilometer scales: Laboratory, field, and remote sensing imaging spectrometer data from the orange hill porphyry copper deposit, alaska, usa. In: 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), pp. 5418–5421.
Kokaly, R.F., King, T.V., Hoefen, T.M., 2013. Surface mineral maps of afghanistan derived from hymap imaging spectrometer data, version 2. Tech. Rep., U.S. Geological Survey Data Series 787.
Koponen, S., Pulliainen, J., Kallio, K., Hallikainen, M., 2002. Lake water quality classification with airborne hyperspectral spectrometer and simulated meris data. Remote Sens. Environ. 79 (1), 51–59.
Kotsiantis, S.B., Zaharakis, I., Pintelas, P., 2007. Supervised machine learning: A review of classification techniques. Emerg. Artif. Intell. Appl. Comput. Eng. 160, 3–24.
Kotsiantis, S.B., Zaharakis, I.D., Pintelas, P.E., 2006. Machine learning: a review of classification and combining techniques. Artif. Intell. Rev. 26 (3), 159–190.
Koturwar, S., Merchant, S., 2017. Weight initialization of deep neural networks (dnns) using data statistics. arXiv preprint arXiv:1710.10570.
Krizhevsky, A., 5 2012. Learning multiple layers of features from tiny images. Tech. Rep., University of Toronto.
Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems. pp. 1097–1105.
Kruse, F., Boardman, J., Lefkoff, A., Young, J., Kierein-Young, K., Cocks, T., Jensen, R., Cocks, P., 2000. Hymap: an australian hyperspectral sensor solving global problems-results from usa hymap data acquisitions. In: Proc. of the 10th Australasian Remote Sensing and Photogrammetry Conference. pp. 18–23.
Kuching, S., 2007. The performance of maximum likelihood, spectral angle mapper, neural network and decision tree classifiers in hyperspectral image analysis. J. Comput. Sci. 3 (6), 419–423.
Kunkel, B., Blechinger, F., Lutz, R., Doerffer, R., van der Piepen, H., Schroder, M., 1988. ROSIS (Reflective Optics System Imaging Spectrometer) - A candidate instrument for polar platform missions. In: Seeley, J., Bowyer, S. (Eds.), Proc. SPIE 0868 Optoelectronic technologies for remote sensing from space, pp. 8.
Landgrebe, D., 2002. Hyperspectral image data analysis. IEEE Signal Process. Mag. 19 (1), 17–28.
Landgrebe, D.A., 2005. Signal Theory Methods in Multispectral Remote Sensing. Wiley-Blackwell.
Lange, J., Cavallaro, G., Götz, M., Erlingsson, E., Riedel, M., 2018. The influence of sampling methods on pixel-wise hyperspectral image classification with 3d convolutional neural networks. In: IGARSS 2018–2018 IEEE International Geoscience and Remote Sensing Symposium, pp. 2087–2090.
Larochelle, H., Bengio, Y., 2008. Classification using Discriminative Restricted Boltzmann Machines. In: Proceedings of the 25th international conference on Machine learning - ICML '08. p. 536.
Le, J.H., Yazdanpanah, A.P., Regentova, E.E., Muthukumar, V., 2015. A deep belief network for classifying remotely-sensed hyperspectral data. In: Bebis, G., Boyle, R., Parvin, B., Koracin, D., Pavlidis, I., Feris, R., McGraw, T., Elendt, M., Kopper, R., Ragan, E., Ye, Z., Weber, G. (Eds.), Advances in Visual Computing. Springer International Publishing, Cham, pp. 682–692.
LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep Learning. Nature 521, 436–444.
Lecun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. Proc. IEEE 86 (11), 2278–2324.
Lee, C.A., Gasster, S.D., Plaza, A., Chang, C.-I., Huang, B., 2011. Recent developments in

high performance computing for remote sensing: A review. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 4 (3), 508–527.
Lee, H., Kwon, H., 2016. Contextual deep cnn based hyperspectral classification. In: 2016 IEEE International Geoscience and Remote Sensing Symposium IGARSS, pp. 3322–3325.
Lee, H., Kwon, H., 2017. Going deeper with contextual cnn for hyperspectral image classification. IEEE Trans. Image Process. 26 (10), 4843–4855.
Lei, D., Chen, X., Zhao, J., 2018. Opening the black box of deep learning. arXiv preprint arXiv:1805.08355.
Leng, J., Li, T., Bai, G., Dong, Q., Dong, H., 2016. Cube-cnn-svm: A novel hyperspectral image classification method. In: 2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI), pp. 1027–1034.
Li, C., Chen, C., Carlson, D., Carin, L., 2016. Preconditioned stochastic gradient langevin dynamics for deep neural networks. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence. AAAI'16. AAAI Press, pp. 1788–1794.
Li, J., June 2015. Active learning for hyperspectral image classification with a stacked autoencoders based neural network. In: 2015 7th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS). pp. 1–4.
Li, J., Bruzzone, L., Liu, S., 2015a. Deep feature representation for hyperspectral image classification. In: 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS). pp. 4951–4954.
Li, J., Zhao, X., Li, Y., Du, Q., Xi, B., Hu, J., 2018a. Classification of hyperspectral imagery using a new fully convolutional neural network. IEEE Geosci. Remote Sens. Lett. 15 (2), 292–296.
Li, S., Song, W., Fang, L., Chen, Y., Ghamisi, P., Benediktsson, J.A., 2019a. Deep learning for hyperspectral image classification: An overview. IEEE Trans. Geosci. Remote Sens.
Li, T., Leng, J., Kong, L., Guo, S., Bai, G., Wang, K., 2018b. Dcnr: deep cube cnn with random forest for hyperspectral image classification. Multimedia Tools Appl.
Li, T., Zhang, J., Zhang, Y., 2014. Classification of hyperspectral image based on deep belief networks. In: Proc. IEEE Int. Conf. Image Proces. pp. 5132–5136.
Li, W., Chen, C., Su, H., Du, Q., 2015b. Local binary patterns and extreme learning machine for hyperspectral imagery classification. IEEE Trans. Geosci. Remote Sens. 53 (7), 3681–3693.
Li, W., Du, Q., 2016. A survey on representation-based classification and detection in hyperspectral remote sensing imagery. Pattern Recogn. Lett. 83, 115–123.
Li, W., Wu, G., Zhang, F., Du, Q., 2017a. Hyperspectral image classification using deep pixel-pair features. IEEE Trans. Geosci. Remote Sens. 55 (2), 844–853.
Li, X., 2018. Preconditioned stochastic gradient descent. IEEE Trans. Neural Networks Learn. Syst. 29 (5), 1454–1466.
Li, Y., Xie, W., Li, H., 2017b. Hyperspectral image reconstruction by deep convolutional neural network for classification. Pattern Recogn. 63, 371–383.
Li, Y., Zhang, H., Shen, Q., 2017c. Spectral-spatial classification of hyperspectral imagery with 3d convolutional neural network. Remote Sens. 9 (1), 67.
Li, Z., Huang, L., He, J., 2019b. A multiscale deep middle-level feature fusion network for hyperspectral classification. Remote Sens. 11 (6), 695.
Liang, H., Li, Q., 2016. Hyperspectral imagery classification using sparse representations of convolutional neural network features. Remote Sens. 8 (2), 99.
Liang, J., Zhou, J., Qian, Y., Wen, L., Bai, X., Gao, Y., 2017. On the sampling strategy for evaluation of spectral-spatial methods in hyperspectral image classification. IEEE Trans. Geosci. Remote Sens. 55 (2), 862–880.
Liang, M., Jiao, L., Yang, S., Liu, F., Hou, B., Chen, H., 2018. Deep multiscale spectral-spatial feature fusion for hyperspectral images classification. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 11 (8), 2911–2924.
Lin, M., Chen, Q., Yan, S., 2013a. Network in network. arXiv preprint arXiv:1312.4400.
Lin, Z., Chen, Y., Zhao, X., Wang, G., Dec 2013b. Spectral-spatial classification of hyperspectral image using autoencoders. In: 2013 9th International Conference on Information, Communications Signal Processing. pp. 1–5.
Lipton, Z.C., 2016. The mythos of model interpretability. arXiv preprint arXiv:1606.03490.
Liu, B., Yu, X., Zhang, P., Tan, X., Yu, A., Xue, Z., 2017a. A semi-supervised convolutional neural network for hyperspectral image classification. Remote Sens. Lett. 8 (9), 839–848.
Liu, B., Yu, X., Zhang, P., Yu, A., Fu, Q., Wei, X., 2018. Supervised deep feature extraction for hyperspectral image classification. IEEE Trans. Geosci. Remote Sens. 56 (4), 1909–1921.
Liu, M., Shi, J., Li, Z., Li, C., Zhu, J., Liu, S., 2017b. Towards better analysis of deep convolutional neural networks. IEEE Trans. Visual. Comput. Graphics 23 (1), 91–100.
Liu, P., Zhang, H., Eom, K.B., 2017c. Active deep learning for classification of hyperspectral images. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 10 (2), 712–724.
Liu, Q., Zhou, F., Hang, R., Yuan, X., 2017d. Bidirectional-convolutional lstm based spectral-spatial feature learning for hyperspectral image classification. Remote Sens. 9 (12), 1330.
Liu, X., Van De Weijer, J., Bagdanov, A.D., 2019. Exploiting unlabeled data in cnns by self-supervised learning to rank. IEEE Trans. Pattern Anal. Machine Intell.
Liu, Y., Cao, G., Sun, Q., Siegel, M., 2015. Hyperspectral classification via deep networks and superpixel segmentation. Int. J. Remote Sens. 36 (13), 3459–3482.
Long, J., Shelhamer, E., Darrell, T., June 2015a. Fully convolutional networks for semantic segmentation. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3431–3440.
Long, M., Cao, Y., Wang, J., Jordan, M.I., 2015b. Learning transferable features with deep adaptation networks. arXiv preprint arXiv:1502.02791.
Lu, D., Weng, Q., 2007. A survey of image classification methods and techniques for improving classification performance. Int. J. Remote Sens. 28 (5), 823–870.
Lucas, R., Rowlands, A., Niemann, O., Merton, R., 2004. Advanced Image Processing Techniques for Remotely Sensed Hyperspectral Data. Springer Berlin Heidelberg, Berlin, Heidelberg.
Lulla, V., 2009. Hyperspectral applications in urban geography. In: Gatrell, J.D., Jensen, R.R. (Eds.), Planning and Socioeconomic Applications. Springer, Netherlands, Dordrecht, pp. 79–86.
Luo, H., 2018. Shorten spatial-spectral rnn with parallel-gru for hyperspectral image classification. arXiv preprint arXiv:1810.12563.
Luo, Y., Zou, J., Yao, C., Zhao, X., Li, T., Bai, G., 2018. Hsi-cnn: A novel convolution neural network for hyperspectral image. In: 2018 International Conference on Audio, Language and Image Processing (ICALIP), pp. 464–469.
Lyu, H., Lu, H., Mou, L., 2016. Learning a transferable change rule from a recurrent neural network for land cover change detection. Remote Sens. 8 (6), 506.
Ma, N., Peng, Y., Wang, S., Leong, P., 2018a. An unsupervised deep hyperspectral anomaly detector. Sensors 18 (3), 693.
Ma, W., Yang, Q., Wu, Y., Zhao, W., Zhang, X., 2019. Double-branch multi-attention mechanism network for hyperspectral image classification. Remote Sens. 11 (11), 1307.
Ma, X., Fu, A., Wang, J., Wang, H., Yin, B., 2018b. Hyperspectral image classification based on deep deconvolution network with skip architecture. IEEE Trans. Geosci. Remote Sens. 56 (8), 4781–4791.
Ma, X., Wang, H., Geng, J., 2016a. Spectral-spatial classification of hyperspectral image based on deep auto-encoder. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 9 (9), 4073–4085.
Ma, X., Wang, H., Geng, J., Wang, J., July 2016b. Hyperspectral image classification with small training set by deep network and relative distance prior. In: 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS). pp. 3282–3285.
Ma, X., Wang, H., Wang, J., 2016c. Semisupervised classification for hyperspectral image based on multi-decision labeling and deep feature learning. ISPRS J. Photogramm. Remote Sens. 120, 99–107.
Maas, A.L., Hannun, A.Y., Ng, A.Y., 2013. Rectifier nonlinearities improve neural network acoustic models. In: Proc. icml. vol. 30. p. 3.
Maaten, L.v.d., Hinton, G., 2008. Visualizing data using t-sne. J. Machine Learn. Res. 9 (Nov), 2579–2605.
MacKay, D.J.C., 1992. Information-based objective functions for active data selection. Neural Comput. 4 (4), 590–604.
Mahendran, A., Vedaldi, A., 2015. Understanding deep image representations by inverting them. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5188–5196.
Mahesh, S., Jayas, D., Paliwal, J., White, N., 2015. Hyperspectral imaging to classify and monitor quality of agricultural materials. J. Stored Prod. Res. 61, 17–26.
Maji, P., Mullins, R., 2018. On the reduction of computational complexity of deep convolutional neural networks. Entropy 20 (4), 305.
Makantasis, K., Karantzalos, K., Doulamis, A., Doulamis, N., 2015. Deep supervised learning for hyperspectral data classification through convolutional neural networks. In: 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), pp. 4959–4962.
Man, Q., Dong, P., Guo, H., 2015. Pixel- and feature-level fusion of hyperspectral and lidar data for urban land-use classification. Int. J. Remote Sens. 36 (6), 1618–1644.
Martens, J., Sutskever, I., 2012. Training deep and recurrent networks with hessian-free optimization. In: Montavon, G., Orr, G.B., Müller, K.-R. (Eds.), Neural Networks: Tricks of the Trade, Second Edition. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 479–535.
Mazhari, N., Malekzadeh Shafaroudi, A., Ghaderi, M., 2017. Detecting and mapping different types of iron mineralization in sangan mining region, ne iran, using satellite image and airborne geophysical data. Geosci. J. 21 (1), 137–148.
McInnes, L., Healy, J., Melville, J., 2018. Umap: Uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:1802.03426.
Mei, S., Ji, J., Bi, Q., Hou, J., Du, Q., Li, W., 2016. Integrating spectral and spatial information into deep convolutional neural networks for hyperspectral classification. In: 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), pp. 5067–5070.
Mei, S., Ji, J., Geng, Y., Zhang, Z., Li, X., Du, Q., 2019a. Unsupervised spatial-spectral feature learning by 3d convolutional autoencoder for hyperspectral classification. IEEE Trans. Geosci. Remote Sens.
Mei, S., Ji, J., Hou, J., Li, X., Du, Q., 2017. Learning sensor-specific spatial-spectral features of hyperspectral images via convolutional neural networks. IEEE Trans. Geosci. Remote Sens. 55 (8), 4520–4533.
Mei, X., Pan, E., Ma, Y., Dai, X., Huang, J., Fan, F., Du, Q., Zheng, H., Ma, J., 2019b. Spectral-spatial attention networks for hyperspectral image classification. Remote Sens. 11 (8), 963.
Melgani, F., Bruzzone, L., 2004. Classification of hyperspectral remote sensing images with support vector machines. IEEE Trans. Geosci. Remote Sens. 42 (8), 1778–1790.
Min, E., Guo, X., Liu, Q., Zhang, G., Cui, J., Long, J., 2018. A survey of clustering with deep learning: From the perspective of network architecture. IEEE Access 6, 39501–39514.
Molchanov, D., Ashukha, A., Vetrov, D., 2017. Variational dropout sparsifies deep neural networks. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70. JMLR. org, pp. 2498–2507.
Mookambiga, A., Gomathi, V., 2016. Comprehensive review on fusion techniques for spatial information enhancement in hyperspectral imagery. Multidimension. Syst. Signal Process. 27 (4), 863–889.
Mou, L., Bruzzone, L., Zhu, X.X., 2019. Learning spectral-spatial-temporal features via a recurrent convolutional neural network for change detection in multispectral imagery. IEEE Trans. Geosci. Remote Sens. 57 (2), 924–935.
Mou, L., Ghamisi, P., Zhu, X.X., 2017. Deep recurrent neural networks for hyperspectral image classification. IEEE Trans. Geosci. Remote Sens. 55 (7), 3639–3655.
Mou, L., Ghamisi, P., Zhu, X.X., 2018. Unsupervised spectral-spatial feature learning via deep residual conv-deconv network for hyperspectral image classification. IEEE

Trans. Geosci. Remote Sens. 56 (1), 391–406.
Mouroulis, P., Van Gorp, B., Green, R.O., Dierssen, H., Wilson, D.W., Eastwood, M., Boardman, J., Gao, B.-C., Cohen, D., Franklin, B., et al., 2014. Portable remote imaging spectrometer coastal ocean sensor: design, characteristics, and first flight results. Appl. Opt. 53 (7), 1363–1380.
Mughees, A., Tao, L., 2016. Efficient deep auto-encoder learning for the classification of hyperspectral images. In: 2016 International Conference on Virtual Reality and Visualization (ICVRV), pp. 44–51.
Mura, M.D., Benediktsson, J.A., Waske, B., Bruzzone, L., 2010. Morphological attribute profiles for the analysis of very high resolution images. IEEE Trans. Geosci. Remote Sens. 48 (10), 3747–3762.
Murugan, P., 2017. Feed forward and backward run in deep convolution neural network. arXiv preprint arXiv:1711.03278.
Murugan, P., Durairaj, S., 2017. Regularization and optimization strategies in deep convolutional neural network. arXiv preprint arXiv:1712.04711.
Nair, V., Hinton, G.E., 2010. Rectified Linear Units Improve Restricted Boltzmann Machines. In: Johannes Fürnkranz and Thorsten Joachims (Ed.), Proceedings of the 27th International Conference on Machine Learning (ICML-10). Omnipress, pp. 807–814.
Nam, H., Kim, H.-E., 2018. Batch-instance normalization for adaptively style-invariant neural networks. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (Eds.), Advances in Neural Information Processing Systems 31. Curran Associates, Inc., pp. 2558–2567.
Narumalani, S., Mishra, D.R., Wilson, R., Reece, P., Kohler, A., 2009. Detecting and mapping four invasive species along the floodplain of north Platte river, Nebraska. Weed Technol. 23 (1), 99–107.
Newell, A., Yang, K., Deng, J., 2016. Stacked hourglass networks for human pose estimation. In: European Conference on Computer Vision. Springer, pp. 483–499.
Nguyen, A., Yosinski, J., Clune, J., 2015. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 427–436.
Nguyen, Q., Hein, M., 2018. Optimization landscape and expressivity of deep cnns. In: International Conference on Machine Learning, pp. 3727–3736.
Nogueira, K., Penatti, O.A., dos Santos, J.A., 2017. Towards better exploiting convolutional neural networks for remote sensing scene classification. Pattern Recogn. 61, 539–556.
Okan, A., Özdemir, B., Gedik, B.E., Yasemin, C., Çetin, Y., 2014. Hyperspectral classification using stacked autoencoders with deep learning. In: 2014 6th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS). pp. 1–4.
Olmanson, L.G., Brezonik, P.L., Bauer, M.E., 2013. 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. 130, 254–265.
Pan, S.J., Yang, Q., 2010. A survey on transfer learning. IEEE Trans. Knowledge Data Eng. 22 (10), 1345–1359.
Pan, X., Luo, P., Shi, J., Tang, X., 2018. Two at once: Enhancing learning and generalization capacities via ibn-net. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 464–479.
Paoletti, M., Haut, J., Plaza, J., Plaza, A., 2018a. Deep&dense convolutional neural network for hyperspectral image classification. Remote Sens. 10 (9), 1454.
Paoletti, M.E., Haut, J.M., Fernandez-Beltran, R., Plaza, J., Plaza, A.J., Pla, F., 2018b. Capsule networks for hyperspectral image classification. IEEE Trans. Geosci. Remote Sens. 1–16.
Paoletti, M.E., Haut, J.M., Fernandez-Beltran, R., Plaza, J., Plaza, A.J., Pla, F., 2018c. Deep pyramidal residual networks for spectral-spatial hyperspectral image classification. IEEE Trans. Geosci. Remote Sens. 1–15.
Paoletti, M.E., Haut, J.M., Plaza, J., Plaza, A., 2017a. A new deep convolutional neural network for fast hyperspectral image classification. ISPRS J. Photogramm. Remote Sens.
Paoletti, M.E., Haut, J.M., Plaza, J., Plaza, A., Liu, Q., Hang, R., July 2017b. Multicore implementation of the multi-scale adaptive deep pyramid matching model for remotely sensed image classification. In: 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS). pp. 2247–2250.
Paoletti, M.E., m. Haut, J., Plaza, J., Plaza, A., July 2018. An investigation on self-normalized deep neural networks for hyperspectral image classification. In: IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium. pp. 3607–3610.
Park, S., Kwak, N., 2017. Analysis on the dropout effect in convolutional neural networks. In: Lai, S.-H., Lepetit, V., Nishino, K., Sato, Y. (Eds.), Computer Vision – ACCV 2016. Springer International Publishing, Cham, pp. 189–204.
Patricia, N., Caputo, B., 2014. Learning to learn, from transfer learning to domain adaptation: A unifying perspective. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1442–1449.
Paul, S., Kumar, D.N., 2018. Spectral-spatial classification of hyperspectral data with mutual information based segmented stacked autoencoder approach. ISPRS J. Photogramm. Remote Sens. 138, 265–280.
Pearlman, J.S., Barry, P.S., Segal, C.C., Shepanski, J., Beiso, D., Carman, S.L., 2003. Hyperion, a space-based imaging spectrometer. IEEE Trans. Geosci. Remote Sens. 41 (6), 1160–1173.
Pedamonti, D., 2018. Comparison of non-linear activation functions for deep neural networks on mnist classification task. arXiv preprint arXiv:1804.02763.
Peerbhay, K.Y., Mutanga, O., Ismail, R., 2015. Random forests unsupervised classification: The detection and mapping of solanum mauritianum infestations in plantation forestry using hyperspectral data. IEEE J. Sel. Top. Appl. Earth Obser. Remote Sens. 8 (6), 3107–3122.
Penttilä, J., 2017. A method for anomaly detection in hyperspectral images, using deep convolutional autoencoders. Master's thesis. University of Jyväskylä.
Petersson, H., Gustafsson, D., Bergstrom, D., 2016. Hyperspectral image analysis using deep learning—a review. In: 2016 Sixth International Conference on Image Processing Theory, Tools and Applications (IPTA). IEEE, pp. 1–6.
Pezeshki, M., Fan, L., Brakel, P., Courville, A., Bengio, Y., 2016. Deconstructing the ladder network architecture. In: International Conference on Machine Learning, pp. 2368–2376.
Pignatti, S., Palombo, A., Pascucci, S., Romano, F., Santini, F., Simoniello, T., Umberto, A., Vincenzo, C., Acito, N., Diani, M., et al., 2013. The prisma hyperspectral mission: Science activities and opportunities for agriculture and land monitoring. In: 2013 IEEE International Geoscience and Remote Sensing Symposium-IGARSS. IEEE, pp. 4558–4561.
Plaut, E., 2018. From principal subspaces to principal components with linear autoencoders. CoRR abs/1804.10253. URL http://arxiv.org/abs/1804.10253.
Plaza, A., Benediktsson, J.A., Boardman, J.W., Brazile, J., Bruzzone, L., Camps-Valls, G., Chanussot, J., Fauvel, M., Gamba, P., Gualtieri, A., Marconcini, M., Tilton, J.C., Trianni, G., 2009. Recent advances in techniques for hyperspectral image processing. Remote Sens. Environ. 113 (1), S110–S122.
Plaza, A., Chang, C.-I., 2008. Clusters versus fpga for parallel processing of hyperspectral imagery. Int. J. High Performance Comput. Appl. 22 (4), 366–385.
Plaza, A., Du, Q., Chang, Y.-L., King, R.L., 2011a. High performance computing for hyperspectral remote sensing. IEEE J. Sel. Top. Appl. Earth Obser. Remote Sens. 4 (3), 528–544.
Plaza, A., Plaza, J., Paz, A., Sanchez, S., 2011b. Parallel hyperspectral image and signal processing [applications corner]. IEEE Signal Process. Mag. 28 (3), 119–126.
Qiu, F., Jensen, J., 2004. Opening the black box of neural networks for remote sensing image classification. Int. J. Remote Sens. 25 (9), 1749–1768.
Qiu, Q., Wu, X., Liu, Z., Tang, B., Zhao, Y., Wu, X., Zhu, H., Xin, Y., 2017. Survey of supervised classification techniques for hyperspectral images. Sensor Rev. 37 (3), 371–382.
Quirita, V.A.A., da Costa, G.A.O.P., Happ, P.N., Feitosa, R.Q., Ferreira, R.d.S., Oliveira, D.A.B., Plaza, A., 2017. A new cloud computing architecture for the classification of remote sensing data. IEEE J. Sel. Top. Appl. Earth Obser. Remote Sens. 10 (2), 409–416.
Ramachandran, P., Zoph, B., Le, Q.V., 2017. Swish: a self-gated activation function. arXiv preprint arXiv:1710.05941 7.
Ran, L., Zhang, Y., Wei, W., Zhang, Q., 2017. A hyperspectral image classification framework with spatial pixel pair features. Sensors 17 (10), 2421.
Randhe, P.H., Durbha, S.S., Younan, N.H., Aug 2016. Embedded high performance computing for on-board hyperspectral image classification. In: 2016 8th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS). pp. 1–5.
Rasmus, A., Berglund, M., Honkala, M., Valpola, H., Raiko, T., 2015. Semi-supervised learning with ladder networks. In: Advances in neural information processing systems. pp. 3546–3554.
Rasti, B., Scheunders, P., Ghamisi, P., Licciardi, G., Chanussot, J., 2018. Noise reduction in hyperspectral imagery: Overview and application. Remote Sens. 10 (3), 482.
Ratle, F., Camps-Valls, G., Weston, J., 2010. Semisupervised neural networks for efficient hyperspectral image classification. IEEE Trans. Geosci. Remote Sens. 48 (5), 2271–2282.
Rauber, P.E., Fadel, S.G., Falcao, A.X., Telea, A.C., 2017. Visualizing the hidden activity of artificial neural networks. IEEE Trans. Visualizat. Comput. Graphics 23 (1), 101–110.
Ravanelli, M., Bengio, Y., 2018. Interpretable convolutional filters with sincnet. arXiv preprint arXiv:1811.09725.
Resmini, R.G., Kappus, M.E., Aldrich, W.S., Harsanyi, J.C., Anderson, M., 1997. Mineral mapping with hyperspectral digital imagery collection experiment (hydice) sensor data at cuprite, nevada, u.s.a. Int. J. Remote Sens. 18 (7), 1553–1570.
Richter, R., 2005. Hyperspectral sensors for military applications. Tech. Rep., German Aerospace Center Wessling (DLR), Wessling (Germany).
Rickard, L.J., Basedow, R.W., Zalewski, E.F., Silverglate, P.R., Landers, M., 1993. Hydice: An airborne system for hyperspectral imaging. In: Imaging Spectrometry of the Terrestrial Environment. vol. 1937. International Society for Optics and Photonics, pp. 173–180.
Rifai, S., Vincent, P., Muller, X., Glorot, X., Bengio, Y., 2011. Contractive auto-encoders: Explicit invariance during feature extraction. In: Proceedings of the 28th International Conference on International Conference on Machine Learning. Omnipress, pp. 833–840.
Roberts, D.A., Dennison, P.E., Gardner, M.E., Hetzel, Y., Ustin, S.L., Lee, C.T., 2003. Evaluation of the potential of hyperion for fire danger assessment by comparison to the airborne visible/infrared imaging spectrometer. IEEE Trans. Geosci. Remote Sens. 41 (6), 1297–1310.
Roberts, D.A., Quattrochi, D.A., Hulley, G.C., Hook, S.J., Green, R.O., 2012. Synergies between vswir and tir data for the urban environment: An evaluation of the potential for the hyperspectral infrared imager (hyspiri) decadal survey mission. Remote Sens. Environ. 117, 83–101.
Rodríguez-Pérez, J.R., Riaño, D., Carlisle, E., Ustin, S., Smart, D.R., 2007. Evaluation of hyperspectral reflectance indexes to detect grapevine water status in vineyards. Am. J. Enology Viticulture 58 (3), 302–317.
Romero, A., Gatta, C., Camps-Valls, G., 2016. Unsupervised deep feature extraction for remote sensing image classification. IEEE Trans. Geosci. Remote Sens. 54 (3), 1349–1362.
Romero, A., Radeva, P., Gatta, C., 2015. Meta-parameter free unsupervised sparse feature learning. IEEE Trans. Pattern Anal. Mach. Intell. 37 (8), 1716–1722.
Roodposhti, M.S., Aryal, J., Lucieer, A., Bryan, B.A., 2019. Uncertainty assessment of

hyperspectral image classification: Deep learning vs. random forest. Entropy 21 (1), 78.
Roy, S.K., Krishna, G., Dubey, S.R., Chaudhuri, B.B., 2019. Hybridsn: Exploring 3d–2d cnn feature hierarchy for hyperspectral image classification. arXiv preprint arXiv:1902.06701.
Rußwurm, M., Körner, M., 2017. Temporal vegetation modelling using long short-term memory networks for crop identification from medium-resolution multi-spectral satellite images. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1496–1504.
Sabale, S.P., Jadhav, C.R., 2015. Hyperspectral image classification methods in remote sensing - a review. In: 2015 International Conference on Computing Communication Control and Automation, pp. 679–683.
Sabalel, S., Jadhav, C., 2014. Supervised, unsupervised, and semisupervised classification methods for hyperspectral image classification-a review. Int. J. Sci. Res. (IJSR) 3 (12).
Sabour, S., Frosst, N., Hinton, G.E., 2017. Dynamic routing between capsules. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (Eds.), Advances in Neural Information Processing Systems 30. Curran Associates, Inc., pp. 3856–3866.
Salimans, T., Kingma, D.P., 2016. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In: Advances in Neural Information Processing Systems. pp. 901–909.
Salman, M., Yüksel, S.E., May 2016. Hyperspectral data classification using deep convolutional neural networks. In: 2016 24th Signal Processing and Communication Application Conference (SIU). pp. 2129–2132.
Sánchez, S., Ramalho, R., Sousa, L., Plaza, A., 2015. Real-time implementation of remotely sensed hyperspectral image unmixing on gpus. J. Real-Time Image Proc. 10 (3), 469–483.
Santurkar, S., Tsipras, D., Ilyas, A., Madry, A., 2018. How does batch normalization help optimization? In: Advances in Neural Information Processing Systems. pp. 2483–2493.
Sanz, C., 2001. Der (dynamic evidential reasoning), applied to the classification of hyperspectral images. In: IGARSS 2001. Scanning the Present and Resolving the Future. Proceedings. IEEE 2001 International Geoscience and Remote Sensing Symposium (Cat. No. 01CH37217), vol. 4. IEEE, pp. 1904–1906.
Savage, S.H., Levy, T.E., Jones, I.W., 2012. Prospects and problems in the use of hyperspectral imagery for archaeological remote sensing: a case study from the faynan copper mining district, jordan. J. Archaeol. Sci. 39 (2), 407–420.
Scafutto, R.D.M., de Souza Filho, C.R., de Oliveira, W.J., 2017. Hyperspectral remote sensing detection of petroleum hydrocarbons in mixtures with mineral substrates: Implications for onshore exploration and monitoring. ISPRS J. Photogramm. Remote Sens. 128, 146–157.
Scherer, D., Müller, A., Behnke, S., 2010. Evaluation of pooling operations in convolutional architectures for object recognition. In: Diamantaras, K., Duch, W., Iliadis, L.S. (Eds.), Artificial Neural Networks – ICANN 2010. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 92–101.
Schmidhuber, J., 2015. Deep learning in neural networks: An overview. Neural Networks 61, 85–117.
Sellami, A., Farah, M., Farah, I.R., Solaiman, B., 2019. Hyperspectral imagery classification based on semi-supervised 3-d deep neural network and adaptive band selection. Expert Syst. Appl. 129, 246–259.
Seydgar, M., Alizadeh Naeini, A., Zhang, M., Li, W., Satari, M., 2019. 3-d convolution-recurrent networks for spectral-spatial classification of hyperspectral images. Remote Sens. 11 (7), 883.
Shaham, U., Stanton, K., Li, H., Nadler, B., Basri, R., Kluger, Y., 2018. Spectralnet: Spectral clustering using deep neural networks. arXiv preprint arXiv:1801.01587.
Shamsolmoali, P., Zareapoor, M., Yang, J., 2018. Convolutional neural network in network (cnnin): hyperspectral image classification and dimensionality reduction. IET Image Proc.
Shang, X., Chisholm, L.A., 2014. Classification of australian native forest species using hyperspectral remote sensing and machine-learning classification algorithms. IEEE J. Sel. Top. Appl. Earth Obser. Remote Sens. 7 (6), 2481–2489.
Sharma, A., Liu, X., Yang, X., 2018. Land cover classification from multi-temporal, multi-spectral remotely sensed imagery using patch-based recurrent neural networks. Neural Networks 105, 346–355.
Shi, C., Pun, C.-M., 2018. Multi-scale hierarchical recurrent neural networks for hyperspectral image classification. Neurocomputing 294, 82–93.
Shi, C., Wang, L., 2014. Incorporating spatial information in spectral unmixing: A review. Remote Sens. Environ. 149, 70–87.
Shi, X.-Z., Aspandiar, M., Oldmeadow, D., 2014. Using hyperspectral data and plsr modelling to assess acid sulphate soil in subsurface. J. Soils Sediments 14 (5), 904–916.
Shwartz-Ziv, R., Tishby, N., 2017. Opening the black box of deep neural networks via information. arXiv preprint arXiv:1703.00810.
Sidorov, O., Hardeberg, J.Y., 2019. Deep hyperspectral prior: Denoising, inpainting, super-resolution. arXiv preprint arXiv:1902.00301.
Signoroni, A., Savardi, M., Baronio, A., Benini, S., 2019. Deep learning meets hyperspectral image analysis: A multidisciplinary review. J. Imag. 5 (5), 52.
Sima, C., Dougherty, E.R., 2008. The peaking phenomenon in the presence of feature-selection. Pattern Recogn. Lett. 29 (11), 1667–1674.
Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
Simpson, A.J., 2015. Dither is better than dropout for regularising deep neural networks. arXiv preprint arXiv:1508.04826.
Slavkovikj, V., Verstockt, S., De Neve, W., Van Hoecke, S., Van de Walle, R., 2015. Hyperspectral image classification with convolutional neural networks. In: Proceedings of the 23rd ACM International Conference on Multimedia. ACM, pp. 1159–1162.
Smolensky, P., 1986. Information processing in dynamical systems: foundations of harmony theory. In: In: David, E., McLelland, J.L. (Eds.), Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol. 6. MIT Press, pp. 194–281 Ch. 6.
Song, B., Li, J., Mura, M.D., Li, P., Plaza, A., Bioucas-Dias, J.M., Benediktsson, J.A., Chanussot, J., 2014. Remotely sensed image classification using sparse representations of morphological attribute profiles. IEEE Trans. Geosci. Remote Sens. 52 (8), 5122–5136.
Song, W., Li, S., Fang, L., Lu, T., 2018. Hyperspectral image classification with deep feature fusion network. IEEE Trans. Geosci. Remote Sens. 56 (6), 3173–3184.
Sonoda, S., Murata, N., 2017. Neural network with unbounded activation functions is universal approximator. Appl. Comput. Harmonic Anal. 43 (2), 233–268.
Springenberg, J.T., Dosovitskiy, A., Brox, T., Riedmiller, M.A., 2014. Striving for simplicity: The all convolutional net. CoRR abs/1412.6806. URL http://arxiv.org/abs/1412.6806.
Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: A simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 15, 1929–1958.
Srivastava, R.K., Greff, K., Schmidhuber, J., 2015. Training very deep networks. In: Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R. (Eds.), Advances in Neural Information Processing Systems 28. Curran Associates, Inc., pp. 2377–2385.
Stein, D.W.J., Beaven, S.G., Hoff, L.E., Winter, E.M., Schaum, A.P., Stocker, A.D., 2002. Anomaly detection from hyperspectral imagery. IEEE Signal Process. Mag. 19 (1), 58–69.
Strachan, I.B., Pattey, E., Boisvert, J.B., 2002. Impact of nitrogen and environmental conditions on corn as detected by hyperspectral reflectance. Remote Sens. Environ. 80 (2), 213–224.
Su, J., Vargas, D.V., Sakurai, K., 2019. One pixel attack for fooling deep neural networks. IEEE Trans. Evol. Comput.
Sun, L., Wu, Z., Liu, J., Xiao, L., Wei, Z., 2015. Supervised spectral–spatial hyperspectral image classification with weighted markov random fields. IEEE Trans. Geosci. Remote Sens. 53 (3), 1490–1503.
Sutskever, I., Martens, J., Dahl, G., Hinton, G., 17–19 Jun 2013. On the importance of initialization and momentum in deep learning. In: Dasgupta, S., McAllester, D. (Eds.), Proceedings of the 30th International Conference on Machine Learning. vol. 28 of Proceedings of Machine Learning Research. PMLR, Atlanta, Georgia, USA, pp. 1139–1147.
Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going deeper with convolutions. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–9.
Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z., 2016. Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2818–2826.
Tan, K., Wu, F., Du, Q., Du, P., Chen, Y., 2019. A parallel gaussian–bernoulli restricted boltzmann machine for mining area classification with hyperspectral imagery. IEEE J. Sel. Top. Appl. Earth Obser. Remote Sens. 12 (2), 627–636.
Tao, C., Pan, H., Li, Y., Zou, Z., 2015. Unsupervised spectral-spatial feature learning with stacked sparse autoencoder for hyperspectral imagery classification. IEEE Geosci. Remote Sens. Lett. 12 (12), 2438–2442.
Tarabalka, Y., Benediktsson, J.A., Chanussot, J., 2009. Spectral-spatial classification of hyperspectral imagery based on partitional clustering techniques. IEEE Trans. Geosci. Remote Sens. 47 (8), 2973–2987.
Tarabalka, Y., Fauvel, M., Chanussot, J., Benediktsson, J.A., 2010. Svm-and mrf-based method for accurate classification of hyperspectral images. IEEE Geosci. Remote Sens. Lett. 7 (4), 736–740.
Teke, M., Deveci, H.S., Haliloglu, O., Gürbüz, S.Z., Sakarya, U., 2013. A short survey of hyperspectral remote sensing applications in agriculture. In: 2013 6th International Conference on Recent Advances in Space Technologies (RAST), pp. 171–176.
Theodoridis, S., Koutroumbas, K., 2003. Pattern Recognition. Elsevier Science.
Tian, K., Zhou, S., Guan, J., 2017. Deepcluster: A general clustering framework based on deep learning. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, pp. 809–825.
Tompson, J., Goroshin, R., Jain, A., LeCun, Y., Bregler, C., 2015. Efficient object localization using convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 648–656.
Transon, J., d'Andrimont, R., Maugnard, A., Defourny, P., 2017. Survey of current hyperspectral earth observation applications from space and synergies with sentinel-2. In: 2017 9th International Workshop on the Analysis of Multitemporal Remote Sensing Images (MultiTemp), pp. 1–8.
Transon, J., d'Andrimont, R., Maugnard, A., Defourny, P., 2018. Survey of hyperspectral earth observation applications from space in the sentinel-2 context. Remote Sens. 10 (2).
Tuia, D., Camps-Valls, G., Nov 2009. Recent advances in remote sensing image processing. In: 2009 16th IEEE International Conference on Image Processing (ICIP). pp. 3705–3708.
Tuia, D., Flamary, R., Courty, N., 2015. Multiclass feature learning for hyperspectral image classification: Sparse and hierarchical solutions. ISPRS J. Photogramm. Remote Sens. 105, 272–285.
Ulyanov, D., Lebedev, V., Vedaldi, A., Lempitsky, V., 2016a. Texture networks: Feed-forward synthesis of textures and stylized images. In: Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48. ICML'16. JMLR.org, pp. 1349–1357.
Ulyanov, D., Vedaldi, A., Lempitsky, V., 2016b. Instance normalization: The missing ingredient for fast stylization. arXiv preprint arXiv:1607.08022.
Ustin, S.L., DiPietro, D., Olmstead, K., Underwood, E., Scheer, G.J., June 2002a. Hyperspectral remote sensing for invasive species detection and mapping. In: IEEE

International Geoscience and Remote Sensing Symposium. vol. 3. pp. 1658–1660 vol 3.
Ustin, S.L., Roberts, D.A., Gardner, M., Dennison, P., 2002b. Evaluation of the potential of hyperion data to estimate wildfire hazard in the santa ynez front range, santa barbara, california. In: IEEE International Geoscience and Remote Sensing Symposium. vol. 2. pp. 796–798 vol 2.
Vane, G., Evans, D.L., Kahle, A.B., 1989. Recent advances in airborne terrestrial remote sensing with the Nasa airborne visible/infrared imaging spectrometer (aviris), airborne synthetic aperture radar (sar), and thermal infrared multispectral scanner (tims). In: 12th Canadian Symposium on Remote Sensing Geoscience and Remote Sensing Symposium. pp. 942–943.
Varshney, P., Arora, M., 2004. Advanced Image Processing Techniques for Remotely Sensed Hyperspectral Data. Springer.
Venkatesan, R., Prabu, S., 2019. Hyperspectral image features classification using deep learning recurrent neural networks. J. Med. Syst. 43 (7), 216.
Veraverbeke, S., Dennison, P., Gitas, I., Hulley, G., Kalashnikova, O., Katagis, T., Kuai, L., Meng, R., Roberts, D., Stavros, N., 2018. Hyperspectral remote sensing of fire: State-of-the-art and future perspectives. Remote Sens. Environ. 216, 105–121.
Wan, L., Zeiler, M., Zhang, S., Cun, Y.L., Fergus, R., 17–19 Jun 2013. Regularization of neural networks using dropconnect. In: Dasgupta, S., McAllester, D. (Eds.), Proceedings of the 30th International Conference on Machine Learning. Vol. 28 of Proceedings of Machine Learning Research. PMLR, Atlanta, Georgia, USA, pp. 1058–1066.
Wan, X., Zhao, C., Wang, Y., Liu, W., 2017. Stacked sparse autoencoder in hyperspectral data classification using spectral-spatial, higher order statistics and multifractal spectrum features. Infrared Phys. Technol. 86, 77–89.
Wang, C., Liu, Y., Bai, X., Tang, W., Lei, P., Zhou, J., 2017a. Deep residual convolutional neural network for hyperspectral image super-resolution. In: Zhao, Y., Kong, X., Taubman, D. (Eds.), Image and Graphics. Springer International Publishing, Cham, pp. 370–380.
Wang, C., Zhang, P., Zhang, Y., Zhang, L., Wei, W., 2016. A multi-label hyperspectral image classification method with deep learning features. In: Proceedings of the International Conference on Internet Multimedia Computing and Service. ACM, pp. 127–131.
Wang, L., Zhang, J., Liu, P., Choo, K.-K.R., Huang, F., 2017b. Spectral–spatial multi-feature-based deep learning for hyperspectral remote sensing image classification. Soft. Comput. 21 (1), 213–221.
Wang, Q., Li, Q., Liu, H., Wang, Y., Zhu, J., Oct 2014. An improved isodata algorithm for hyperspectral image classification. In: 2014 7th International Congress on Image and Signal Processing. pp. 660–664.
Wang, W., Dou, S., Jiang, Z., Sun, L., 2018a. A fast dense spectral–spatial convolution network framework for hyperspectral images classification. Remote Sens. 10 (7), 1068.
Wang, W., Dou, S., Wang, S., 2019. Alternately updated spectral–spatial convolution network for the classification of hyperspectral images. Remote Sens. 11 (15), 1794.
Wang, Y., Jiang, Y., Wu, Y., Zhou, Z.-H., 2010. Multi-manifold clustering. In: Pacific Rim International Conference on Artificial Intelligence. Springer, pp. 280–291.
Wang, Y., Mei, J., Zhang, L., Zhang, B., Zhu, P., Li, Y., Li, X., 2018b. Self-supervised feature learning with crf embedding for hyperspectral image classification. IEEE Trans. Geosci. Remote Sens.
Waske, B., van der Linden, S., Benediktsson, J.A., Rabe, A., Hostert, P., 2010. Sensitivity of support vector machines to random feature selection in classification of hyperspectral data. IEEE Trans. Geosci. Remote Sens. 48 (7), 2880–2889.
Wei, D., Zhou, B., Torralba, A., Freeman, W.T., 2017a. mneuron: A Matlab Plugin to Visualize Neurons From Deep Models. Institute of Technology, Massachusetts.
Wei, W., Zhang, L., Tian, C., Plaza, A., Zhang, Y., 2017b. Structured sparse coding-based hyperspectral imagery denoising with intracluster filtering. IEEE Trans. Geosci. Remote Sens. 55 (12), 6860–6876.
Williams, R.J., Zipser, D., 1989. A learning algorithm for continually running fully recurrent neural networks. Neural Comput. 1 (2), 270–280.
Williams, T., Li, R., 2018. Wavelet pooling for convolutional neural networks. In: International Conference on Learning Representations, . <https://openreview.net/forum?id=rkhlb8lCZ>.
Windrim, L., Melkumyan, A., Murphy, R.J., Chlingaryan, A., Ramakrishnan, R., 2018. Pretraining for hyperspectral convolutional neural network classification. IEEE Trans. Geosci. Remote Sens. 56 (5), 2798–2810.
Wold, S., Esbensen, K., Geladi, P., 1987. Principal Component Analysis. Chemometrics Intell. Laborat. Syst. 2 (1), 37–52.
Wu, H., Prasad, S., 2017. Convolutional recurrent neural networks for hyperspectral data classification. Remote Sens. 9 (3), 298.
Wu, H., Prasad, S., 2018. Semi-supervised deep learning using pseudo labels for hyperspectral image classification. IEEE Trans. Image Process. 27 (3), 1259–1270.
Wu, Y., He, K., 2018. Group normalization. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 3–19.
Wu, Z., Li, Y., Plaza, A., Li, J., Xiao, F., Wei, Z., 2016. Parallel and distributed dimensionality reduction of hyperspectral data on cloud computing architectures. IEEE J. Sel. Top. Appl. Earth Obser. Remote Sens. 9 (6), 2270–2278.
Wyatte, D., Herd, S., Mingus, B., O'Reilly, R., 2012. The role of competitive inhibition and top-down feedback in binding during object recognition. Front. Psychol. 3, 182.
Xiaoli Jiao, C.-I.C., 2007. Unsupervised hyperspectral image classification. Imaging Spectrometry XII, vol. 6661https://doi.org/10.1117/12.732614. pp. 6661 – 6661 – 10.
Xie, J., Girshick, R., Farhadi, A., 2016. Unsupervised deep embedding for clustering analysis. In: International Conference on Machine Learning, pp. 478–487.
Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K., 2017. Aggregated residual transformations for deep neural networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5987–5995.
Xie, W., Li, Y., Jia, X., 2018. Deep convolutional networks with residual learning for accurate spectral-spatial denoising. Neurocomputing 312, 372–381.
Xing, C., Ma, L., Yang, X., 2016. Stacked Denoise Autoencoder Based Feature Extraction and Classification for Hyperspectral Images. J. Sensors 2016.
Xu, B., Gong, P., 2008. Noise estimation in a noise-adjusted principal component transformation and hyperspectral image restoration. Can. J. Remote Sens. 34 (3), 271–286.
Xu, B., Wang, N., Chen, T., Li, M., 2015a. Empirical evaluation of rectified activations in convolutional network. arXiv preprint arXiv:1505.00853.
Xu, X., Lil, f., Plaza, A., 2016a. Fusion of hyperspectral and LiDAR data using morphological component analysis. In: 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), pp. 3575–3578.
Xu, Y., Du, B., Zhang, F., Zhang, L., 2018. Hyperspectral image classification via a random patches network. ISPRS J. Photogramm. Remote Sens. 142, 344–357.
Xu, Y., Du, J., Dai, L., Lee, C., 2015b. A regression approach to speech enhancement based on deep neural networks. IEEE/ACM Trans. Audio, Speech, Language Process. 23 (1), 7–19.
Xu, Y., Wu, Z., Li, J., Plaza, A., Wei, Z., 2016. Anomaly detection in hyperspectral images based on low-rank and sparse representation. IEEE Trans. Geosci. Remote Sens. 54 (4), 1990–2000.
Yang, C., Everitt, J.H., Bradford, J.M., Murden, D., 2004. Airborne hyperspectral imagery and yield monitor data for mapping cotton yield variability. Precision Agric. 5 (5), 445–461.
Yang, H., 1999. A back-propagation neural network for mineralogical mapping from aviris data. Int. J. Remote Sens. 20 (1), 97–110.
Yang, J., Zhao, Y., Chan, J.C., Yi, C., 2016. Hyperspectral image classification using two-channel deep convolutional neural network. In: 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), pp. 5079–5082.
Yang, J., Zhao, Y.-Q., Chan, J.C.-W., 2017. Learning and transferring deep joint spectral–spatial features for hyperspectral classification. IEEE Trans. Geosci. Remote Sens. 55 (8), 4729–4742.
Yang, S., Jin, H., Wang, M., Ren, Y., Jiao, L., 2014. Data-driven compressive sampling and learning sparse coding for hyperspectral image classification. IEEE Geosci. Remote Sens. Lett. 11 (2), 479–483.
Yang, X., Ye, Y., Li, X., Lau, R.Y.K., Zhang, X., Huang, X., 2018. Hyperspectral image classification with deep learning models. IEEE Trans. Geosci. Remote Sens. 56 (9), 5408–5423.
Yi, C., Zhao, Y.Q., Chan, J.C.W., 2018. Hyperspectral image super-resolution based on spatial and spectral correlation fusion. IEEE Trans. Geosci. Remote Sens. 56 (7), 4165–4177.
Yi, C., Zhao, Y.Q., Yang, J., Chan, J.C.W., Kong, S.G., 2017. Joint hyperspectral super-resolution and unmixing with interactive feedback. IEEE Trans. Geosci. Remote Sens. 55 (7), 3823–3834.
Yosinski, J., Clune, J., Bengio, Y., Lipson, H., 2014. How transferable are features in deep neural networks? In: Advances in Neural Information Processing Systems. pp. 3320–3328.
Younos, T., Parece, T., 2015. Advances in Watershed Science and Assessment. The Handbook of Environmental Chemistry. Springer International Publishing.
Yu, D., Seltzer, M.L., Li, J., Huang, J., Seide, F., 2013. Feature learning in deep neural networks - A study on speech recognition tasks. CoRR abs/1301.3605.
Yu, D., Wang, H., Chen, P., Wei, Z., 2014. Mixed pooling for convolutional neural networks. In: Miao, D., Pedrycz, W., Śl?zak, D., Peters, G., Hu, Q., Wang, R. (Eds.), Rough Sets and Knowledge Technology. Springer International Publishing, Cham, pp. 364–375.
Yu, S., Jia, S., Xu, C., 2017. Convolutional neural networks for hyperspectral image classification. Neurocomputing 219, 88–98.
Yuan, Q., Zhang, Q., Li, J., Shen, H., Zhang, L., 2019. Hyperspectral image denoising employing a spatial–spectral deep residual convolutional neural network. IEEE Trans. Geosci. Remote Sens. 57 (2), 1205–1218.
Yue, J., Mao, S., Li, M., 2016. A deep learning framework for hyperspectral image classification using spatial pyramid pooling. Remote Sens. Lett. 7 (9), 875–884.
Yue, J., Zhao, W., Mao, S., Liu, H., 2015. Spectral-spatial classification of hyperspectral images using deep convolutional neural networks. Remote Sens. Lett. 6 (6), 468–477.
Zabalza, J., Ren, J., Zheng, J., Zhao, H., Qing, C., Yang, Z., Du, P., Marshall, S., 2016. Novel segmented stacked autoencoder for effective dimensionality reduction and feature extraction in hyperspectral imaging. Neurocomputing 185, 1–10.
Zeiler, M.D., 2012. Adadelta: an adaptive learning rate method. arXiv preprint arXiv:1212.5701.
Zeiler, M.D., Fergus, R., 2013. Stochastic pooling for regularization of deep convolutional neural networks. CoRR abs/1301.3557. URL http://arxiv.org/abs/1301.3557.
Zhan, Y., Hu, D., Wang, Y., Yu, X., 2018. Semisupervised hyperspectral image classification based on generative adversarial networks. IEEE Geosci. Remote Sens. Lett. 15 (2), 212–216.
Zhang, D., Kang, J., Xun, L., Huang, Y., 2019a. Hyperspectral image classification using spatial and edge features based on deep learning. Int. J. Pattern Recognit Artif Intell.
Zhang, F., Du, B., Zhang, L., Zhang, L., 2016a. Hierarchical feature learning with dropout k-means for hyperspectral image classification. Neurocomputing 187, 75–82 recent Developments on Deep Big Vision.
Zhang, H., He, W., Zhang, L., Shen, H., Yuan, Q., 2014. Hyperspectral image restoration using low-rank matrix recovery. IEEE Trans. Geosci. Remote Sens. 52 (8), 4729–4743.
Zhang, H., Li, Y., 2016. Spectral-spatial classification of hyperspectral imagery based on deep convolutional network. In: 2016 International Conference on Orange Technologies (ICOT), pp. 44–47.
Zhang, H., Li, Y., Jiang, Y., Wang, P., Shen, Q., Shen, C., 2019. Hyperspectral

classification based on lightweight 3-d-cnn with transfer learning. IEEE Trans. Geosci. Remote Sens.
Zhang, H., Li, Y., Zhang, Y., Shen, Q., 2017. Spectral-spatial classification of hyperspectral imagery using a dual-channel convolutional neural network. Remote Sens. Lett. 8 (5), 438–447.
Zhang, L., Du, B., 2012. Recent advances in hyperspectral image processing. Geo-spatial Informat. Sci. 15 (3), 143–156.
Zhang, L., Zhang, L., Du, B., 2016b. Deep learning for remote sensing data: A technical tutorial on the state of the art. IEEE Geosci. Remote Sens. Mag. 4 (2), 22–40.
Zhang, L., Zhang, L., Tao, D., Huang, X., 2012. On combining multiple features for hyperspectral remote sensing image classification. IEEE Trans. Geosci. Remote Sens. 50 (3), 879–893.
Zhang, M., Gong, M., Mao, Y., Li, J., Wu, Y., 2018a. Unsupervised feature extraction in hyperspectral images based on wasserstein generative adversarial network. IEEE Trans. Geosci. Remote Sens. 57 (5), 2669–2688.
Zhang, P., Gong, M., Su, L., Liu, J., Li, Z., 2016c. Change detection based on deep feature representation and mapping transformation for multi-spatial-resolution remote sensing images. ISPRS J. Photogramm. Remote Sens. 116, 24–41.
Zhang, X., Sun, Y., Jiang, K., Li, C., Jiao, L., Zhou, H., 2018b. Spatial sequential recurrent neural network for hyperspectral image classification. IEEE J. Sel. Top. Appl. Earth Obser. Remote Sens. 1–15.
Zhao, C., Wan, X., Zhao, G., Cui, B., Liu, W., Qi, B., 2017a. Spectral-spatial classification of hyperspectral imagery based on stacked sparse autoencoder and random forest. Eur. J. Remote Sens. 50 (1), 47–63.
Zhao, R., Luk, W., Niu, X., Shi, H., Wang, H., 2017b. Hardware acceleration for machine learning. In: 2017 IEEE Computer Society Annual Symposium on VLSI (ISVLSI). IEEE, pp. 645–650.
Zhao, W., Du, S., 2016. Spectral-spatial feature extraction for hyperspectral image classification: A dimension reduction and deep learning approach. IEEE Trans. Geosci. Remote Sens. 54 (8), 4544–4554.
Zhao, W., Guo, Z., Yue, J., Zhang, X., Luo, L., 2015. On combining multiscale deep learning features for the classification of hyperspectral remote sensing imagery. Int. J. Remote Sens. 36 (13), 3368–3379.
Zheng, Z., Zhang, Y., Li, L., Zhu, M., He, Y., Li, M., Guo, Z., He, Y., Yu, Z., Yang, X., Liu, X., Luo, J., Yang, T., Liu, Y., Li, J., 2017. Classification based on deep convolutional neural networks with hyperspectral image. In: 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), pp. 1828–1831.
Zhong, P., Gong, Z., Li, S., Schönlieb, C.B., 2017a. Learning to diversify deep belief networks for hyperspectral image classification. IEEE Trans. Geosci. Remote Sens. 55 (6), 3516–3530.
Zhong, Y., Wang, X., Zhao, L., Feng, R., Zhang, L., Xu, Y., 2016a. Blind spectral unmixing based on sparse component analysis for hyperspectral remote sensing imagery. ISPRS J. Photogramm. Remote Sens. 119, 49–63.
Zhong, Z., Li, J., Luo, Z., Chapman, M., 2017b. Spectral-Spatial Residual Network for Hyperspectral Image Classification: A 3-D Deep Learning Framework. IEEE Trans. Geosci. Remote Sens. PP (99), 1–12.
Zhong, Z., Li, J., Ma, L., Jiang, H., Zhao, H., July 2017c. Deep residual networks for hyperspectral image classification. In: 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS). pp. 1824–1827.
Zhou, F., Hang, R., Liu, Q., Yuan, X., 2018a. Hyperspectral image classification using spectral-spatial lstms. Neurocomputing.
Zhou, F., Hang, R., Liu, Q., Yuan, X., 2018b. Integrating convolutional neural network and gated recurrent unit for hyperspectral image spectral-spatial classification. In: Chinese Conference on Pattern Recognition and Computer Vision (PRCV). Springer, pp. 409–420.
Zhou, J., Liang, J., Qian, Y., Gao, Y., Tong, L., 2015. On the sampling strategies for evaluation of joint spectral-spatial information based classifiers. In: 2015 7th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS). IEEE, pp. 1–4.
Zhou, P., Han, J., Cheng, G., Zhang, B., 2019a. Learning compact and discriminative stacked autoencoder for hyperspectral image classification. IEEE Trans. Geosci. Remote Sens.
Zhou, S., Xue, Z., Du, P., 2019b. Semisupervised stacked autoencoder with cotraining for hyperspectral image classification. IEEE Trans. Geosci. Remote Sens.
Zhou, X., Li, S., Tang, F., Qin, K., Hu, S., Liu, S., 2017. Deep learning with grouped features for spatial spectral classification of hyperspectral images. IEEE Geosci. Remote Sens. Lett. 14 (1), 97–101.
Zhou, X.M., Wang, N., Wu, H., Tang, B.H., Li, Z.L., 2011. Estimation of precipitable water from the thermal infrared hyperspectral data. In: 2011 IEEE International Geoscience and Remote Sensing Symposium, pp. 3241–3244.
Zhu, J., Fang, L., Ghamisi, P., 2018a. Deformable convolutional neural networks for hyperspectral image classification. IEEE Geosci. Remote Sens. Lett. 15 (8), 1254–1258.
Zhu, J., Wu, L., Hao, H., Song, X., Lu, Y., June 2017a. Auto-encoder based for high spectral dimensional data classification and visualization. In: 2017 IEEE Second International Conference on Data Science in Cyberspace (DSC). pp. 350–354.
Zhu, L., Chen, Y., Ghamisi, P., Benediktsson, J.A., 2018b. Generative adversarial networks for hyperspectral image classification. IEEE Trans. Geosci. Remote Sens. 56 (9), 5046–5063.
Zhu, X.X., Tuia, D., Mou, L., Xia, G.-S., Zhang, L., Xu, F., Fraundorfer, F., 2017. Deep learning in remote sensing: A comprehensive review and list of resources. IEEE Geosci. Remote Sens. Mag. 5 (4), 8–36.
Zuo, Z., Shuai, B., Wang, G., Liu, X., Wang, X., Wang, B., Chen, Y., 2015. Convolutional recurrent neural networks: Learning spatial dependencies for image representation. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 18–26.