Segmentation-aided classification of hyperspectral data using spatial dependency of spectral bands

Annalisa Appice[a,b,c,*], Donato Malerba[a,b,c]

[a] *Department of Informatics, Universitá degli Studi di Bari Aldo Moro, via Orabona, Bari 4 - 70125, Italy*
[b] *Consorzio Interuniversitario Nazionale per l'Informatica - CINI, Italy*
[c] *Centro Interdipartimentale di Logica e Applicazioni - CILA, Italy*

Abstract

Classifying every pixel of a hyperspectral image with a certain land-cover type is the cornerstone of hyperspectral image analysis. In the present study a segmentation-aided methodology for the spectral-spatial classification of hyperspectral data is proposed. It considers the spatial dependence of the spectral bands, deals with the curse of dimensionality and handles the spectral variability. A local spatial regularization of spectral information is used, in order to derive an informative joint spectral-spatial representation of the data. A contiguity-based segmentation algorithm is formulated, in order to build the object-wise texture that can aid classifier learning. The hybrid use of the segmentation texture is evaluated in both pre-processing (i.e selecting representative pixels to learn the classifier) and post-processing (i.e. refining predicted labels and removing possible outlier classifications). The experiments performed with the proposed methodology provide encouraging results, also compared to several recent state-of-the-art approaches.

Keywords: spectral-spatial classification, segmentation, local spatial dependency analysis, curse of dimensionality

[*] Corresponding author
Email addresses: annalisa.appice@uniba.it (Annalisa Appice), donato.malerba@uniba.it (Donato Malerba)

1. Introduction

Hyperspectral (HS) remote sensing, also known as imaging spectroscopy, is a major breakthrough in remote sensing technology. HS sensors, mounted on aircraft or satellites, produce digital images (HyperSpectral Images - HSI) of an observed scene, by recording reflected light in hundreds of narrow frequencies covering the visible, near-infrared and shortwave infrared bands (pixel spectrum). Such an abundance of spectral data represents an invaluable source of knowledge regarding the physical nature of the different materials possibly observed. In particular, the high-dimensionality of the measured spectrum is useful for pixel classification, in order to distinguish different landscapes in the image scene [1].

Land cover classification in HSI data is a hot topic, that is still challenging [2]. One of the major reasons is the curse of dimensionality [3]. This is related to the human-supervised effort needed to collect only few labeled imagery pixels (training set) that should also be properly distributed among the classes [4]. In fact, the commonly low number of collected ground-truth labels, compared to the high number of spectral bands (curse of dimensionality), is not always sufficient for a reliable estimate of the classifier parameters. As discussed in [5], this is prone to problems of over-fitting (i.e. the classification model exactly fits the training data without accounting for a wider generalization) or under-fitting (i.e. the model complexity is constrained excessively), which cause a reduction in the classifier's ability. To mitigate the curse of dimensionality, Support Vector Machine is considered an effective method [5], while a data reduction of HSI is widely used [6, 7, 8, 9, 10], in order to decrease the number of spectral bands and select (extract) non-redundant informative features that preserve the discriminative properties of the data. However, recent studies [11, 12] point out that when the number of labels is too scarce to rely on, a supervised classifier may suffer from under-fitting rather than over-fitting. This is due to the fact that the complexity of the learnable parameters far exceeds the limited amounts of training samples. To handle the under-fitting problems that often occur under small sample conditions, one possibility is to reconsider classification as a process of choosing an optimal set of training samples for supervised classifiers, under limited sample conditions.

In the early stages of HSI classification, many methods have concentrated on handling the curse of dimensionality when the abundant spectrum of a pixel is exclusively considered to determine its class [1]. In any case the most recent search trend has definitely highlighted that the presence of spectral variability is a further complexity factor. Spectral variability is caused by many conditions such as incident illumination, atmospheric effects, unwanted shade and shadow, natural spectrum variation and instrument noises. These conditions may provoke two main difficulties which hinder classification [2]. On the one hand, high intra-class spectrum variability makes the identification of a given class very difficult. On the other hand, low inter-class spectral variability makes the discrimination of different classes hard. All such difficulties lead to unsatisfactory classification performances with pixel-wise methods [2]. In general, the consideration of spatial information, which originates from homogeneous areas of contiguous pixels in HSI, provides complementary information to spectral bands, by offering the possibility to boost the pixel-wise classification. Early attempts to incorporate spatial information into hyperspectral classification can be traced back over the last decade [13]. Since then spectral-spatial classification has witnessed a great surge of interest. A wide plethora of successful studies has definitely proved the ability to enhance the performance of spectral pixel-wise classification with object (neighborhood)-based spatial information in precise land-cover mapping, forest inventory or urban-area monitoring (see [2] for a survey on the recent state-of-the-art in spectral-spatial HSI classification).

The use of feature engineering is one of the most popular spectral-spatial approaches. It couples the spatial information to the spectral knowledge by pre-extracting spatial feature profiles of the spectrum, which are then pushed into a subsequent classifier. The morphological [14] and extended morphological [15] profiles are the most frequently computed, due to the excellent spatial features created. Morphological profiles are also computed via collective inference, as a way to express the spectral-aware label dependency [16, 17]. Markov random fields [18], Gabor wavelet transforms [19] and Extinction profiles [20] are also introduced for a better understanding of the spectral scenes, by using spatial and contextual properties. In addition, multi-objective optimization-based sparse unmixing methods are proposed, in order to take full advantage of the spectral characteristics and exploit regularization by the spectral correlations among different individuals [21]. Very recently deep learning-based methods have finally shown promising performance in HSI classification and achieved high accuracy in obtaining discriminative spectral-spatial features [22, 23, 24, 25, 26, 27]. However, the feature representation ability of deep learning is commonly at the cost of a very complex learning process.

Alternatively, some spectral-spatial classification methods incorporate spatial information via segmentation, often in post-processing, after a spectral-based classification has been conducted. In the image segmentation process, an image is partitioned into non-overlapping homogeneous regions, based on one or more homogeneity criteria. Depending on the features employed in the segmentation process, the methods can be categorized into the following groups: spatial-based, spectral-based and spatial-spectral-based (see [28] for a survey). Spatial-based methods search for homogenous regions of spatially connected pixels with the defined criteria. Representative methods in this group include region growing, split-merge methods and watershed methods. Spectral-based methods, such as thresholding and partitional clustering, group pixels into spectral clusters based on spectral similarity measures, without considering the spatial locations of these pixels. Spatial-spectral-based methods exploit spectral-spatial information in the segmentation process and allow for the merging of spatially disjoint regions. One of the first HSI classification methodologies, that uses segmentation as spatial post-processing, is adopted in [29]. It combines the output of a pixel-wise classifier with the morphological watershed transformation. This method is extended in [30] with the incorporation of the segmentation into a multiple spectral-spatial classification that uses an ensemble of classifiers. Subsequent studies also confirm that segmentation-based post-processing can lead to a better delineation of object borders, as well as to a refinement of spatial features through both outlier removal and refinement of the classification results [31]. Finally, a recent study [32] shows that spatial features can be constructed from segmentation regions and segments can be used in combination with iterative active learning, in order to select the pixels to expand the training set. Additional spectral-spatial studies resort to a Markov Random field regularizer [33] and a Loopy Belief propagation [34] to perform pixel-wise classification post-processing. A few hybrid methods also use a mixture of feature engineering and post-processing [35, 36].

In this paper, we revamp the hybrid spectral-spatial philosophy and propose an HSI classification methodology, named SoCRATE (Spectral-spatial COrRelation SegmenTAtion-based ClassifiEr), that includes:

- A specific combination of dimensionality reduction and spectral-spatial feature engineering, which deals with the curse of dimensionality and directly introduces a spatial perspective of the spectrum in the classifier to learn. Dimensionality reduction is performed using Principal Component Analysis (PCA) of spectral data. Feature engineering is performed relying on the geostatistics theory and building spectral-spatial features that measure the degree of spatial dependency of each spectral component.

- A contiguity-based algorithm for the unsupervised data segmentation, that divides the sensed scene into regions, characterized by high spatial dependency over the observed spectrum representation. It exploits contiguity constraints to speed up the segmentation process and considers spectral-spatial features to model spatial dependency of spectral data during the segmentation process. This segmentation knowledge is used twice in the methodology: (1) to sample representative pixels for segments whose labels are acquired by querying a human-supervised oracle, so that they can be used as a training set to learn an accurate supervised classifier under small sample conditions, and (2) to post-process the pixel labeling produced by the learned classifier, so that outlier classifications, which are potentially wrong, can be removed.

- A two-level spectral-spatial classification pattern. The first-level applies the learned classifier and uses the spectral-spatial profile to determine the classifications pixel-wise. The second-level exploits the image objects revealed by segmentation to isolate and correct outlier predicted labels.

We note that principal component analysis, segmentation, spatial dependency analysis and object-wise post-processing of predicted labels have already been explored in the literature. However, to the best of our knowledge, the novelty of this study is the specific formulation adopted for these components (in particular, for segmentation and feature construction), as well as the effectiveness of the combination of these components in a methodology that actually outperforms the classification performance of several state-of-the-art competitors on various hyperspectral data sets. In particular, this study contributes to proving that the proposed formulation of a contiguity-constrained algorithm for segmentation is an effective means to delineate, in an unsupervised manner, the spectral objects which may help in both the training dataset definition and outlier classification removal. In general, our methodology gains in accuracy compared to various classifiers, which also aid segmentation knowledge to yield the final classifications. Another contribution is the use of the geostatistics theory in the HSI classification scenario. In addition, the empirical study proves that the proposed methodology, with features constructed using local spatial indicators of spectral dependency, achieves competitive representation ability also compared to recent deep learning models, without requiring computational-demanding learning architectures and achieving improvement of classification performance. It is interesting to note that the proposed methodology also outperforms several state-of-the-art methods, such as those adopting the widely used morphological and extended morphological methods.

The remainder of this paper is organized as follows. In Section 2 preliminary concepts are introduced, while in Section 3 the proposed methodology is illustrated. Section 4 provides the details of the experiments, which are carried out in this study, and their results, along with important discussions, are reported. In particular, the experiments described show the effectiveness of each component of the proposed methodology, analyze the sensitivity of the performance of the methodology to the set-up of input parameters and compare the performance of the performed classification to that of various recent state-of-the-art spectral-spatial classifiers. Finally, Section 5 summarizes the conclusions.

2. Preliminary concepts

Let $\mathbf{D}$ be an HSI dataset, that is, a collection of n pixels. Each pixel is a region of around a few square meters of the Earth's surface and a function of a hyperspectral sensor spatial resolution. It is associated to the spatial coordinates XY in the image, it is characterized by an m-dimensional vector of spectral bands $\boldsymbol{S} = S_1, S_2, \dots, S_m$ (spectrum) and it can, in principle, be labeled according to an unknown target function, whose range is a finite set of k distinct labels, i.e. $\mathrm{C} = C_1, C_2, \dots, C_k$. As pixels are in general equally-space distributed over a regular grid, an HSI is represented as a matrix. Thus, the spatial coordinate X is associated with the row index, while the spatial coordinate Y is associated with the column index of the matrix. Every spectral feature S_i is numeric and expresses how much the radiation is reflected, on average, at the i-th band of the considered spectrum, from the resolution cell of the considered pixel. Every class C_i represents a distinct theme (i.e. type of Earth's surface object). A spatial neighborhood is a set of pixels q surrounding a center pixel p in the imagery matrix. In the imagery analysis literature, spatial neighborhoods frequently have a square shape [13], although alternative shapes like a circle or a cross can also be considered. Let R be a positive, integer-valued radius, the square neighborhood $\mathrm{N}(i, R)$ of a center pixel i is defined as follows:

$$\mathrm{N}(i, R) = \bigcup_{X=-R}^{X=+R} \bigcup_{Y=-R}^{Y=+R} \{(x_j, y_j) | x_j = x_i + X, y_j = y_i + Y\}, \qquad (1)$$

where (x_i, y_i) are the spatial coordinates of i.

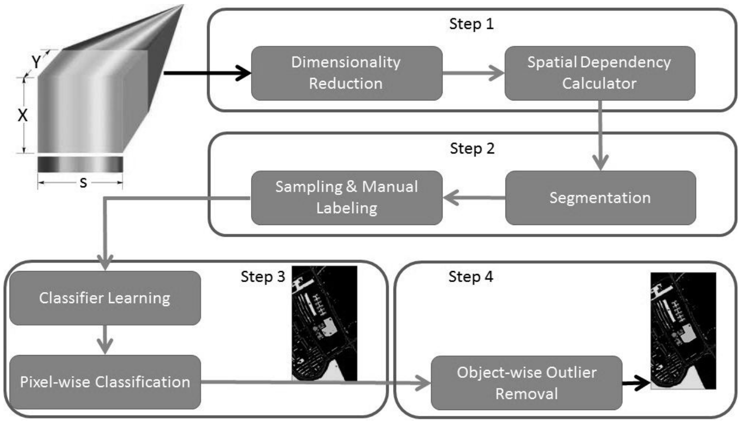
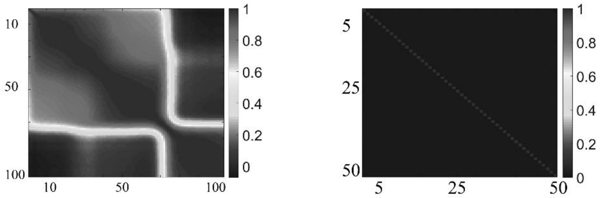

Y
X
S
Step 1
Dimensionality Reduction
Spatial Dependency Calculator
Step 2
Sampling & Manual Labeling
Segmentation
Step 3
Classifier Learning
Pixel-wise Classification
Step 4
Object-wise Outlier Removal

Figure 1: Block diagram of SoCRATE (Spectral-spatial COrRelation SegmenTAtion-based ClassifiEr).

3. Methodology

SoCRATE is a four-stepped methodology for HSI classification (see Figure 1). The first step (see Section 3.1) performs spectral-spatial pre-processing that applies dimensionality reduction followed by spectral-spatial feature engineering, in order to deal with the curse of dimensionality and derive an informative spectral-spatial profile of the HSI dataset. The second step (see Section 3.2) performs a contiguity-constrained segmentation of the HSI dataset over the newly engineered spectral-spatial space, in order to separate material objects in an unsupervised fashion. The segmentation output is subsequently considered to identify a few representative pixels (e.g. centers of detected objects), whose labels can be manually acquired, in order to learn an accurate classifier. The third step (see Section 3.3) learns the classifier from the segmentation-driven training set, as it is spanned over the spectral-spatial features produced. The learned classifier is used to predict pixel-wise the unknown labels of the HSI dataset under study. Finally, the fourth step (see Section 3.4) performs a post-processing that re-uses the segmentation knowledge, in order to better delineate object borders via outlier removal. The input parameters are: (1) the number of N dimensions returned with dimensionality reduction, (2) the similarity threshold



(a) Spectra bands (b) Spectral PCs

Figure 2: Pavia University data (see Section 4.1 for further details): heatmap of the Pearson correlation matrix of spectral bands (Figure 2(a)) and heatmap of Pearson correlation matrix of top-fifty Principal Components, expressing a partially orthogonal reduced representation of the entire spectrum (Figure 2(b)).

ϵ, (3) the number of training samples K and (4) the radius of the square-shaped neighborhood system R.

3.1. Spectral-spatial pre-processing

In the pre-processing phase, a vector of spectral-spatial features is built by reducing the dimensionality of the spectral signature and calculating local indicators of the spatial dependency of the spectral information. We note that these spectral-spatial features will be considered in place of the original spectral features in all the subsequent steps of this methodology (segmentation, pixel-wise classification and outlier post-processing).

3.1.1. Dimensionality reduction

Principal Component Analysis (PCA) is one of the most widely used linear feature extraction techniques, which has been proved to be a powerful HSI data reduction strategy [37, 38, 20]. Specifically, PCA seeks to reduce the dimension of the data and drops the curse of dimensionality by finding a few orthogonal linear combinations (the Principal Components – PCs) of the original spectral bands with the largest variance. In HSI analysis, the preference for the use of PCA for data reduction is also motivated by its ability to derive a collinearity-free characterization of the spectrum. The spectral bands are strongly contemporaneously correlated with each other in the near spectrum, while the spectral principal components are contemporaneously uncorrelated with each other. An illustration of this phenomenon can be seen in Figures 2(a) and 2(b). We note that the collinearity phenomenon among near spectral bands may not be simply neglected, as it leads to a series of problems, such as unreliable coefficients and predictions, as well as aggravated data redundancy and computational complexity [39]. In general, as discussed in [40, 41], PCA is a mandatory step in improving the learning performance, by removing collinearity, speeding up the learning process and reducing the data storage requirements.

There are also valid alternatives to PCA. For example, autoencoders, that belong to the neural network family, are similar to PCA as they can be used for finding a low-dimensional representation of input data [42]. They minimize the same objective function as PCA, but they are more flexible than PCA, due to the activation function that can introduce non-linearities in the encoding. Although autoencoders are really a big class of potentially extremely complex models, the advantage of PCA is that it is simple and efficient to train in comparison. Assuming that the linear transformation of PCA fits the spectral data accurately, it is much better to train PCA than try to select some complex deep model. In the empirical study (see Section 4.2), we evaluate the viability of both PCA and autoencoders in various benchmark HSI scenarios, in order to confirm empirically the suitability of PCA for the methodology presented.

3.1.2. Spatial dependency indicators

Once the HSI spectrum is transformed into its PCs and the HSI data are reduced to N top-ranked PCs, new spectral-spatial features are computed by borrowing a local indicator of spatial dependency from the geostatistics theory [43]. In particular, for each selected PC, a spectral-spatial feature is built by calculating a local indicator of the spatial dependency of the considered PC. Widely speaking, a local indicator of spatial dependency allows us to represent spectral information with a spatial regularization, which is introduced with the goal of easing the discovery of deviations from global patterns of spatial association, as well as hot spots like local clusters. Specifically the regularization is applied, in order to reduce the labeling uncertainty due to the possible salt-and-pepper noise that may appear in the spectral measurements (and again in their principal components). In particular, the spectral PCs of a center pixel are regularized by considering the spectral information from the neighbor pixels. Neighbors are processed to express the degree to which the spectral PC of the pixel under consideration is part of a specific contiguous object (see [44] for a survey). For every PC the local indicator returns one value for each pixel; this value expresses the degree to which that pixel is part of a cluster (object). We note that recovering this kind of information is desirable for accurately achieving the objectives of both segmentation and classifications (steps 2 and 3 of this methodology), i.e. distinguishing objects from spectral data and recognizing the class (material) associated to the detected objects.

Several local indicators of spatial dependency are formulated in the geostatistics literature and commonly used in geophysical data elaboration [45]. The standardized Getis and Ord local GI^* [46] is considered here. It is a local indicator of spatial dependency that has gained wide acceptance in clustering studies already conducted in the geostatistics literature [47, 48, 45]. Let us consider a spectral principal component PC and a center pixel i, then standardized Getis and Ord local $GI^*(PC,i)$ is computed as follows:

$$GI^*(PC,i) = \frac{1}{\sqrt{\frac{S^2}{n-1}\left(n\sum_{j=1,j\neq i}^{n} w(ij)^2 - W(i)^2\right)}}\left(\sum_{j=1,j\neq i}^{n} w_{ij}PC(j) - \overline{PC}\;W(i)\right), \tag{2}$$

where $PC(i)$ is the value of PC collected at pixel i, $\overline{PC} = \frac{\sum\limits_{i\in\mathbf{D}} PC(i)}{cardinality(\mathbf{D})}$ is the mean of the data measured for PC over the entire dataset, $w(ij)$ is a spatial (Gaussian or bi-square) weight between the pixels i and j over the neighborhood structure of the HSI, $W(i) = \sum\limits_{j=1,j\neq i}^{n} w(ij)$ and $S^2 = \frac{\sum\limits_{j=1}^{n}(PC(j)-\overline{PC})^2}{n}$. In this study $w(i,j)$ is computed as in [45], that is, $w(i,j) = \frac{1}{\sqrt{d(i,j)}}$, if j belongs to the square-shaped neighborhood of i (i.e. $j \in N(i, R)$ according to Formula 1) , 0 otherwise. $d(\cdot, \cdot)$ is the Euclidean distance computed between the spatial coordinates XY of the considered pixels. According to the theory reported in [46] a positive value for $GI^*(PC, i)$ indicates clusters of high values around i, while a negative value for $GI^*((PC, i)$ indicates clusters of low values around i. This theory inspires our understanding that the calculus of this peculiar spatial dependency indicator can contribute to derive a joint spectral-spatial feature space that will be especially appropriate for the specific objective of both the segmentation step (see Section 3.2) and classification step (see Section 3.3). The foundation of this intuition will also be empirically evaluated in Section 4.2.

Finally, every new computed joint spectral-spatial feature is scaled into the range [0, 1] according to the transformation:

$$\hat{GI}^*(PC, i) = \frac{GI^*(PC, i) - \min\limits_{i \in \mathbf{D}} GI^*(PC, i)}{\max\limits_{i \in \mathbf{D}} GI^*(PC, i) - \min\limits_{i \in \mathbf{D}} GI^*(PC, i)}. \tag{3}$$

This transformation is applied to solve the problem of combining features, whose range, that potentially ranges between $-\infty$ and $+\infty$, may also differ in its orders of magnitude.

3.2. Segmentation and training sampling

The segmentation of the HSI dataset is performed in the space of the joint spectral-spatial features, which are computed as described in Section 3.1. It aims at discovering the image regions (namely segmentation objects) , where the distribution of spectral information is smoothly continuous over space, with boundaries possibly marked by sharp discontinuities, which can be observed in the spatial dependency of the spectrum. The segments are used to identify representative pixels throughout the HSI dataset. The labels of these sampled pixels are acquired by querying a human-supervised oracle and used to populate the training set to learn an accurate supervised classifier. The introduction of this segmentation-stratified sampling is based upon the idea that the unknown class of a pixel is a spectral response of the pixel itself. In particular, the expected outcome is that the segmentation objects, although discovered in an unsupervised manner (i.e. without the label information), reasonably delineate the distribution of the unknown classes over the space. Hence, sampling per segment allows the safe acquisition of labels, which are plausibly stratified among the various unknown classes, without knowing apriori how the classes are actually distributed throughout the HSI dataset.

The segmentation process is two-stepped. The first step (details in Section 3.2.1) divides the HSI dataset into micro-objects, in order to reduce the number of samples for the segmentation and to speed up the entire process. For this step a contiguity-constrained algorithm is adopted. Procedurally it performs a region-growing process that grows each segementation region from a seed pixel by evaluating the similarity of the joint spectral-spatial data at contiguous pixels. We note that the region-growing approach is being widely used for remote sensing applications, as it guarantees the creation of closed regions [49]. It is preferred in this study, as it is a conceptually simple and yet effective and robust technique for performing segmentation. It performs well also in the presence of noise and gives very good segmentations, which correspond well to the object edges. In particular, it can fit the requirements of learning under spatial dependency and takes advantage of contiguity constraints between pixels to reduce the number of possible solutions. It can also force the algorithm to converge fast onto largely similar areal boundaries as shown in [50, 51].

In this study, the region-growing approach embeds a homogeneity function that allows us to use the model spatial dependency of spectral data during the segmentation process. It is controlled by a spectral similarity threshold ϵ (i.e., a user-defined threshold, ranging between 0 and 1). The higher the similarity threshold, the lower the number of discovered micro-objects. The second step (details in Section 3.2.2) derives the final segmentation objects by resorting to an agglomerative procedure that aggregates contiguous, similar micro-objects to form K final objects. This reduces the possible over-segmentation in the region-growing results by allowing us to control the final number of segments.

We note that both steps use contiguity constraints defined on the basis of a contiguity relationship between pixels. Let i and j be two pixels of the HSI dataset $(i, j \in \mathbf{D})$. They are contiguous if and only if j belongs to the square-shaped neighborhood of i built with radius $R = 1$ ($\mathrm{N}(i, 1)$, defined according to Formula 1), that is:

$$contiguous(i, j)\ iff\ j \in \mathrm{N}(i, 1). \tag{4}$$

Once the segmentation objects are identified, they are used to sample pixels for the oracle (details in Section 3.2.2) .

3.2.1. Micro-object discovery

The discovery starts assigning $\iota = 1$, where ι enumerates the computed micro-objects. The construction of a new micro-object O_ι starts with a seed i that is an HSI pixel still un-assigned to any micro-object. The seed is selected by exploring the imagery matrix row by row (from left to right and top to bottom). The pixel i is added to O_ι, while O_ι is expanded by using i as the seed of the expansion process. The expanded micro-object O_ι is added to the output structure $\mathcal{O}$. ι is incremented by one and the discovery process is iteratively repeated until all the pixels are assigned to a micro-object. The expansion of O_ι is driven by a seed pixel i and is recursively defined. First, the contiguity pixel set $\mathrm{E}(i)$ having a seed i is constructed by considering pixels j, which are contiguous to i according to Formula 4 and still un-assigned to any micro-object. Formally,

$$\mathrm{E}(i) = \{j \in \mathbf{D} | un - assigned(j) \wedge contiguous(i, j)\}. \tag{5}$$

Then candidate micro-object $\mathrm{tempO} = \mathrm{O}_\iota \cup \mathrm{E}(i)$ is built. The average dissimilarity $diss(\cdot)$ is computed on candidate micro-object tempO spanned on the joint spectral-spatial feature space. Formally,

$$diss(\text{tempO}) = \frac{\sum_{p=1}^{N} \lambda(PC_p) \left(\max_{i \in \text{tempO}} \hat{GI}^*(PC_p, i) - \min_{i \in \text{tempO}} \hat{GI}^*(PC_p, i) \right)}{\sum_{p=1}^{N} \lambda(PC_p)}, \tag{6}$$

where $\lambda(PC_p)$ is the eigenvalue-based rank assigned to PC_p during the PCA. Two cases are distinguished: in the former case, tempO satisfies the dissimilarity condition (i.e., $diss(\text{tempO}) \leq \epsilon$) and then pixels of $\text{E}(i)$ are definitely assigned to O_ι. In the latter case, tempO does not satisfy the dissimilarity condition and the addition of each pixel of $\text{E}(i)$ to O_ι is evaluated pixel-by-pixel. In both cases pixels newly assigned in O_ι are iteratively chosen as seeds to continue the expansion process. The expansion process stops if no new pixel is added to the micro-object.

3.2.2. Agglomerative segmentation

The agglomerative segmentation starts elaborating the object set $\mathcal{O}$ as it is constructed by the micro-object discovery.[1] It iteratively merges a pair of objects which are selected over current $\mathcal{O}$, until the number of objects left in $\mathcal{O}$ is less than or equal to K (i.e. the iterative step stops when the HSI dataset is segmented into K segmentation objects).[2] At each iteration, the objects $(\text{O}'_i, \text{O}'_j)$, which are identified for the merge operation, represent the two objects of $\mathcal{O}$, which are contiguous across space, whose merge (the object $\text{O}'_i \cup \text{O}'_j$) achieves the minimum dissimilarity over current $\mathcal{O}$, that is:

$$(\text{O}'_i, \text{O}'_j) = \operatorname*{argmin}_{\text{O}_i \in \mathcal{O}, \text{O}_j \in \mathcal{O}, \text{O}_i \neq \text{O}_j, contiguous(\text{O}_i, \text{O}_j)} diss(\text{O}_i \cup \text{O}_j). \tag{7}$$

[1] If the micro-object discovery is excluded, $\mathcal{O}$ is naively populated with one micro-object formed for every pixel in the HSI dataset.

[2] If $\mathcal{O}$ initially collects less than K micro-objects, no iteration of the agglomerative segmentation is actually performed.

Objects O_i and O_j are contiguous across space if and only if a pixel $i \in \mathrm{O}'_i$ and a pixel $j \in \mathrm{O}'_j$ exist, so that the contiguity relationship between i and j (i.e., $contiguous(i, j)$) is verified according to Formula 4.

3.2.3. Segmentation-based sampling

The sampling procedure is geometrically defined by accounting for the shape of the segmentation. In this study the shape of a segmentation object $\mathrm{O} \in \mathcal{O}$ is represented through the Minimum Bounding Rectangle (MBR) that envelopes the pixels enclosed in O. The center of the MBR is determined, in order to represent the internal part of the object, while the cardinal point vertices of the MBR are determined, in order to represent the boundary of the object. The pixels to query the oracle are then sampled as the closest to these vertices.

Procedurally, for each segmentation object $\mathrm{O} \in \mathcal{O}$, the coordinates $minX$, $maxX$, $minY$ and $maxY$ are firstly computed as follows:

$$\begin{aligned} minX &= \min_x \{x|x = x_i, i \in \mathrm{O}\} \quad & minY &= \min_y \{y|y = y_i, i \in \mathrm{O}\}, \\ maxX &= \max_x \{x|x = x_i, i \in \mathrm{O}\} \quad & maxY &= \max_y \{y|y = y_i i \in \mathrm{O}\}. \end{aligned} \tag{8}$$

They are used to determine the coordinates of the center $(\frac{minX+maxX}{2}, \frac{minY+maxY}{2})$, as well as the coordinates of the north-west vertex $(minX, maxY)$, the south-east vertex $(maxX, minY)$, the north-east vertex $(maxX, maxY)$, the south-west vertex $(minX, minY)$, the north vertex $(\frac{minX+maxX}{2}, maxY)$, the south vertex $(\frac{minX+maxX}{2}, minY)$, the west vertex $(minX, \frac{minY+maxY}{2})$ and the east vertex $(maxX, \frac{minY+maxY}{2})$ of $mbr(O)$. Then the segmentation objects are sorted by their cardinality (the number of pixels they enclose) and repeatedly explored until K labels are acquired for the oracle. In the first exploration, for each object $\mathrm{O} \in \mathcal{O}$, the object pixel closest to the center of $mbr(\mathrm{O})$ is selected for the oracle. Ideally $\mathcal{O}$ contains K objects, so querying the oracle to acquire the central label of each segmentation object will be sufficient to complete the construction of the training set (and acquire K labels). However, there are two cases where this ideal condition may not be satisfied: (1) $\mathcal{O}$ contains less than K objects (due to the use of high ϵ in the micro-object discovery step) or (2) the oracle is unable to provide the label for a query pixel. In both cases, the exploration of $\mathcal{O}$ is iterated until K labels are acquired or all the pixels are queried. Each new exploration of $\mathcal{O}$ is driven by one of the cardinal points, so that, for each segmentation object, the object pixel un-queried before and closest to the cardinal point vertex under consideration is sampled for the oracle.

3.3. Pixel-wise classification

Let us consider the training set L, that is the set of K pixels, whose labels are acquired by querying the oracle (see details in Section 3.2). L is spanned over the joint spectral-spatial feature space (see details in Section 3.1) and the class feature. A supervised classifier γ: $\hat{GI}^*(PC_1) \times \ldots, \times \hat{GI}^*(PC_N) \mapsto C$ is learned from L. This is a pixel-wise classifier, as it allows us to assign a class to every pixel. This is a spectral-spatial classifier, as it is learned on a feature space that synthesizes the spectral information with a spatial regularization. This classifier can be used to determine pixel-wise the unknown label of any pixel of an HSI dataset.

Technical remarks. The inductive Support Vector Machine (SVM) with the Gaussian kernel [52] is selected as the base algorithm to learn the pixel-wise classifier. This choice is motivated by several studies reported in the literature (e.g. [13, 31, 53, 16, 17]), which show that inductive SVMs with a Gaussian kernel are applied to hyperspectral image classification with great success, outperforming several other inductive classifiers. In particular, experiments in [16] prove that the Gaussian kernel is better than the Linear kernel in the considered scenario. As the SVM is defined in the literature for binary classification problems, the “one-against-all” strategy is used, in order to adapt the binary classifier to the multi-class problem. Previous investigations [16, 17] have also shown that the accuracy of SVMs, learned in the hyperspectral scenarios, varies significantly with γ, while it is approximately stable with C. Based upon this observation, SVMs are learned with parameter $C = 64$, while parameter γ is optimally selected according to a grid-search method and a three-fold cross validation (3-CV) of the labeled set. Specifically, the grid search is used, in order to find the value of γ that yields the best average 3-CV OA on the configurations with γ ranging among $2^{-4}, 2^{-3}, 2^{-2}, 2^{-1}, 1, 2^1, 2^2, 2^3, 2^4$ and 2^5.

3.4. Object-wise post-processing

This pixel-wise classification can be subsequently refined, in order to remove possible outlier classifications. The outlier removal is object-wise, as it uses the texture introduced by the discovered segmentation and forces the assignment of the predominant class in the segmentation object to each pixel in the object under consideration. Procedurally, let us consider a segmentation object $\mathrm{O} \in \mathcal{O}$, then the class $C(\mathrm{O})$ is the most frequent label associated to a pixel of O. This class is forcefully assigned to each pixel in O, if the class entropy measured over the collection of pixel-wise labels acquired/predicted in O is greater than 0.5. This entropy-based condition is imposed to apply the object-wise refinement only to the outliers, which emerge in those objects where there is a class that is actually predominant with respect to the others.

4. Experimental evaluation and discussion

SoCRATE, whose implementation is publicly available,[3] is written in Java. The Java implementation of SVM, included in the 3.6 WEKA toolkit, is used. Three benchmark hyperspectral images (see Subsection 4.1) are considered, in order to validate its effectiveness in terms of accuracy and efficiency. The accuracy performance is evaluated with Overall Accuracy (OA), Average Accuracy (AA) and Cohen's kappa coefficient (κ) [54].[4] These accuracy metrics are computed on the testing ground-truth samples, which are overlooked when the supervised classifier is learned. The efficiency performance is evaluated with the computation time (TIME), spent in seconds completing the learning process

[3] `http://www.di.uniba.it/~appice/software/SoCRATE/index.htm`

[4] Overall Accuracy measures the percentage of correctly classified samples. Average Accuracy measures the average percentage of correctly classified samples for an individual class. Cohen's kappa coefficient measures the percentage agreement corrected by the level of agreement that could be expected by chance alone.

on ReCaS cloud, CPU 1:8 @ 2Ghz 2,16.0 GB RAM, running Ubuntu 14.04.4 (GNULinux 3.13.0-39-generic x86–64). The presentation of the results is organized as follows. Initially the effectiveness of the pre-processing (PCA and spatial dependency analysis), post-processing (outlier removal via object-wise classification) and segmentation components is evaluated. To this purpose the performance of SoCRATE is compared to that of its baselines which are derived without using the PCA, the spatial dependency analysis, the segmentation knowledge and/or the object-wise classification (see Subsection 4.2). Subsequently, the sensitivity of the performance of SoCRATE is evaluated along the parameter configuration – the number of principal components, the size of the neighborhood, the segmentation similarity threshold and the base classification algorithm (see Subsection 4.3). Finally, a brief discussion of recent evaluation results, reported in the hyperspectral image classification literature, is illustrated (see Subsection 4.4).

4.1. Hyperspectral datasets

Three well-known publicly available real hyperspectral datasets, namely Indian Pines, Pavia University and Salinas Valley,[5] are used in this experimental study. These three datasets have different spatial resolutions and they were also acquired from different types of land cover (agriculture and urban areas), using two different sensors (AVIRIS and ROSIS). Therefore, the three datasets are selected for the present study to demonstrate the general applicability of the proposed approach. These datasets are also considered in the majority of recent, relevant studies on hyperspectral image classification (e.g. [17, 22, 23, 24, 20, 55, 25, 26]).

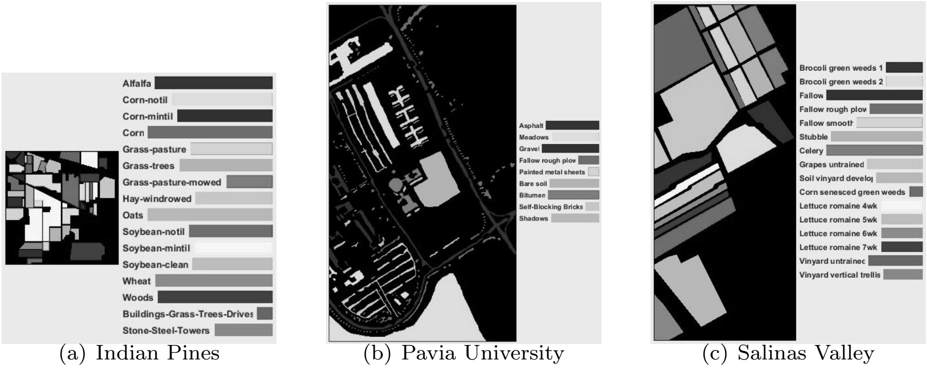
4.1.1. Indian Pines

This dataset was obtained by the Airborne Visible Infrared Imaging Spectrometer (AVIRIS) sensor over the Indian Pines region, in Northwestern Indiana,

[5] `http://www.grss-ieee.org/community/technical-committees/data-fusion/`

Table 1: Details of the class ground-truth for Indian Pines (columns 1-2), Pavia University (columns 3-4) and Salinas Valley (columns 5-6) datasets.

Indian Pines		Pavia University		Salinas Valley	
Class name	Pixels	Class name	Pixels	Class name	Pixels
Alfalfa	46	Asphalt	6631	Brocoli green weeds 1	2009
Corn-notill	1428	Meadows	18649	Brocoli green weeds 2	3726
Corn-mintill	830	Gravel	2099	Fallow	1976
Corn	237	Trees	3064	Fallow rough plow	1394
Grass-pasture	483	Painted metal sheets	1345	Fallow smooth	2678
Grass-trees	730	Bare Soil	5029	Stubble	3959
Grass-pasture-mowed	28	Bitumen	1330	Celery	3579
Hay-windrowed	478	Self-Blocking Bricks	3682	Grapes untrained	11271
Oats	20	Shadows	947	Soil vinyard develop	6203
Soybean-notill	972	-	-	Corn weeds	3278
Soybean-mintill	2455	-	-	Lettuce romaine 4wk	1068
Soybean-clean	593	-	-	Lettuce romaine 5wk	1927
Wheat	205	-	-	Lettuce romaine 6wk	916
Woods	1265	-	-	Lettuce romaine 7wk	1070
Building-Grass-Trees	386	-	-	Vineyard untrained	7268
Stone-Steel-Towers	93	-	-	Vineyard vertical trellis	1807
Unknown	10776	Unknown	164624	Unknown	56975
Total	21025	Total	207400	Total	111104



(a) Indian Pines (b) Pavia University (c) Salinas Valley

Figure 3: The ground-truth map (with known classes) of Indian Pines (Figure 3(a)), Pavia University (Figure 3(b)) and Salinas Valley (Figure 3(c)) datasets.

in 1992. The dataset covers mostly an agricultural area. It consists of 145×145 pixels with 20 m spatial resolution and 224 spectral bands in the 0.4–2.5 μm wavelength range . The number of bands is reduced to 200 by removing water absorption bands ([104–108], [150–163], 220). The dataset contains 10249 labeled pixels, which are classified in 16 mutually exclusive land-cover classes. A description of the dataset is reported in Table 1, while a map of the land-cover is shown in Figure 3(a). As reported in [13], this dataset represents a very challenging land-cover classification scenario, in which the primary crops of the area (mainly corn and soybeans) were very early in their growth cycle, with only about 5% canopy cover. Discriminating among the major crops under these circumstances can be a very difficult task. This scenario is also made more complex by the imbalanced number of available labeled pixels per class.

4.1.2. Pavia University

This dataset was obtained by the Reflective Optics System Imaging Spectrometer (ROSIS) sensor during a flight campaign over the Engineering School at the University of Pavia in 2003. The dataset covers an urban area with some buildings and large meadows. It comprises 610×340 pixels with a spatial resolution of 1.3 m and reflectance information of 103 spectral bands in the 0.43–0.86 μm wavelength range . The dataset contains 42776 labeled pixels, which belong to 9 different urban-cover classes. A description of the dataset is reported in Table 1, while a map of the ground-truth urban-cover classes is shown in Figure 3(b).

4.1.3. Salinas Valley

This dataset was acquired by AVIRIS over Salinas Valley, Southern California, in 1998. The dataset covers mostly an agricultural area. It contains 512×217 pixels with 20 m spatial resolution and 224 spectral bands in the 0.4–2.5 μm wavelength range. As with the Indian Pines scene, the number of bands is reduced to 200 by removing water absorption bands. The dataset contains 54129 labeled pixels, which belong to 16 different classes, including vegetables, bare soils and vineyard fields. A description of the dataset is reported in Table 1, while a map of the ground-truth land-cover classes is shown in Figure 3(c).

4.2. Comparative analysis

The three datasets described in Subsection 4.1 are considered. The dimensionality reduction is run with the number of dimensions $N = 50$, the spectral-spatial dependency analysis is run with radius $R = 7$, the segmentation is run with the similarity threshold $\epsilon = 0.05$. The accuracy and efficiency of the evaluated learning methodology is analyzed with the training sample size K ranging between 1% and 5% of the ground-truth sample size.

4.2.1. Pre-processing analysis

We start by evaluating the performance of the proposed methodology with respect to the two steps of the pre-processing phase, specifically the dimensionality reduction and the spatial dependency analysis. To this purpose, we compare SoCRATE to different baseline configurations, namely SoCRATE-GI, SoCRATE-PCA, SoCRATE-AUTO+GI, SoCRATE-AUTO and SoCRATE-NONE (see details in Table 2). SoCRATE performs the PCA, in order to reduce the dimensionality of the spectral data and computes the local Getis and Ord indicator of the extracted principal components to derive the spatial information. Baselines SoCRATE-AUTO+GI and SoCRATE-AUTO replace the PCA with an autoencoder. The input size of the autoencoder is the spectral signature size, while the output size is 50 (as for the PCA). The autoencoder considered for this study includes 2 hidden layers with RELU as the activation function. The optimizer ADAM is adopted with MSE as the loss. The fitting procedure is iterated on

Table 2: Compared pre-processing configurations of SoCRATE. They are defined along the dimensionality reduction (principal component analysis – PCA, Autoencoder or No dimensionality reduction – NONE) and the spatial dependency analysis (Local Getis and Ord Indicator – GI or No spatial dependency analysis – NONE).

Configuration	Dimensionality reduction ($N = 50$)	Spatial dependency analysis ($R = 7$)
SocRATE	PCA	GI
SoCRATE-GI	NONE	GI
SoCRATE-PCA	PCA	NONE
SoCRATE-AUTO+GI	Autoencoder	GI
SoCRATE-AUTO	Autoencoder	None
SoCRATE-NONE	NONE	NONE

Table 3: Time spent in seconds computing the dimensionality reduction (DR TIME) via PCA or autoencoder, as well as accuracy performance (OA, AA and k) of classification done with both dimensionality reduction techniques considered, $N = 50$, $R = 7$ and $\epsilon = 0.03$. The training sample size is equal to 5% of the ground-truth sample size. The autoencoder is run with three different layer architectures, that is, 256 × 128, 512 × 256 and 1024 × 512.

Dataset/DR	Metric	PCA	Autoencoder		
			256 × 128	512 × 256	1024 × 512
Indian Pines	DR TIME	48.061	46.955	97.954	326.827
	OA	.9818	.9579	.9474	.9634
	AA	.9793	.9606	.9376	.9453
	k	.9792	.9520	.9401	.9582
Pavia University	DR TIME	260.699	378.444	862.679	2999.018
	OA	.9984	.9986	.9990	.9951
	AA	.9971	.9606	.9981	.9935
	k	.9979	.9520	.9987	.9935
Salinas Valley	DR TIME401.280	568.826	1058.663	1778.336	
	OA	.9994	.9982	.9810	.9928
	AA	.9993	.9990	.9915	.9971
	k	.9993	.9980	.9787	.9920

100 epochs. Three two-layer architectures, namely, 256 × 128, 512 × 256 and 1024 × 512, are evaluated. For each experiment, the autoencoder architecture that maximizes the OA of the object-wise classification is finally considered for the subsequent comparative analysis.

Setting-up apriori the architecture of the autoencoder is neither computation-free nor negligible as a problem, since the quality of the features, that changes with the architecture, may impact on the accuracy of the entire classification process. These considerations are supported by the results illustrated in Table 3. They represent the computation time (in secs) spent computing either the PCA or the autoencoder, as well as the accuracy (OA, AA and k) of the object-wise classifications yielded when the features computed by either the PCA or the autoencoder are considered for the spatial dependency analysis. These results confirm that computing the autoencoder is less efficient than deriving the principal components. In addition, the more complex the architecture, the higher the computation cost of the autoeconder. In any case, the final classification accuracy depends on the architecture of the autoencoder, while the architecture with the best performance may change in each experiment.[6] On the other hand, the final classification accuracy achieved with the principal components is always comparable to the best accuracy achieved with the autoencoder. This reveals that the linear transformation is commonly sufficient to accurately fit the hypespectral data. This motivated our decision to consider the principal components. To complete the comparative study, we also consider baselines SoCRATE-GI and SoCRATE-NONE, which do not use any dimensionality reduction technique, while baselines SoCRATE-PCA, SoCRATE-AUTO and SoCRATE-NONE do not perform the spatial dependency analysis. For the six configurations, we evaluate the accuracy of the object-wise classification achieved after the segmentation-aided post-processing.

Table 4 reports the classification performance (OA, AA, k and TIME) achieved with training sample size equal to 5% of the ground-truth sample size. In addition, Figures 6(a)-6(c), Figures 6(d)-6(f) and Figures 6(g)-6(i) show the change in OA, AA and TIME with the training sample size growing from 1% to 5%. The collected results confirm that the dimensionality reduction, coupled with the spatial dependency analysis, actually contributes to a gain in both accuracy and efficiency. SoCRATE and SoCRATE-AUTO+GI always achieve the highest accuracy in this comparative study. As anticipated in the preliminary analysis of the autoeconder, SoCRATE (with principal components) commonly outperforms SoCRATE-AUTO+GI (with an autoencoder) in Indian Pines and Salinas Valley, while SoCRATE-AUTO+GI is slightly more accurate (but still less efficient) than SoCRATE in Pavia University. This consideration is valid independently of the training sample size evaluated. The accuracy performance is motivated by the fact that the principal components (as well as the autoencoder features) reduce data dimensionality and derive a collinearity-free characterization of the correlation in the spectral signature, while the collinearity phenomenon, if overlooked,

[6] Due to limited, the results of this analysis are only reported for the experiments run with training sample size equal to 5% of the ground-truth sample size. In any case, the reported considerations are independent of the training sample size.

may lead to a series of problems, such as unreliable predictions, that aggravate data redundancy and computational complexity [56].

On the other hand, the spatial dependency information introduces a spatial smoothing of the spectral signature (see the heatmaps in Figures 4(a)-4(j)), which can correct potential spurious inference [57]. In any case, the spatial dependency analysis contributes more than the PCA (or autoencoder) to the observed gain in accuracy. This is confirmed by the observation that both the PCA and the autoencoder, separated from the spatial dependency analysis (SoCRATE-PCA and SoCRATE-AUTO), even achieve the lowest accuracy in this comparative analysis, while the spatial dependency analysis, separated from both types of dimensionality reduction (SoCRATE-GI), still achieves the runner-up accuracy (after SoCRATE and SoCRATE-AUTO+GI). We also note that, although the principal components coupled with the spatial dependency analysis (SoCRATE) generally outperform the autoencoder coupled with the spatial dependency analysis (SoCRATE-AUTO+GI), the autoencoder, separated by the spatial dependency analysis (SoCRATE-AUTO), outperforms the PCA, separated by the spatial dependency analysis (SoCRATE-AUTO). This empirically proves that the proposed spatial smoothing, performed with the local indicator computation, is especially effective coupled with the linear transformation introduced by the principal components, as proposed in this study.

Coherent conclusions can also be drawn from the analysis of the efficiency performance. In fact, coupling the dimensionality reduction (both PCA and autoencoder) with the spatial dependency analysis (SoCRATE and SoCRATE-AUTO+GI) makes the computation faster, although the autoencoder is slower than the PCA. On the other hand, neglecting both these analysis components – dimensionality reduction and spatial dependency analysis (SoCRATE-NONE) – makes the computation slower. A more detailed interpretation of this performance can be derived by analyzing how the computation time is progressing on the learning process. Figures 5(a)-5(c) show the progress report of the cumulative computation time in each step of the learning process (dimensionality reduction, spatial dependency analysis, scaling, segmentation and classification) for the compared configurations. The majority of the computation time is spent performing the segmentation. This also explains the observation that the computation time spent completing the learning process commonly decreases as the training sample size increases (see Figure 6(g)-6(g)). In fact, the training sample size defines the segmentation granularity, so that the smaller the training sample size, the finer-grained the segmentation output and, consequently, the longer the time spent completing the segmentation step and then the entire learning process. On the other hand, the spatial smoothing of the spectral information (performed by SoCRATE, SoCRATE-GI and SoCRATE-AUTO+GI), thanks to a reduction of possible spectral noise, actually diminishes the time spent computing the segmentation of the dataset and, consequently, makes the entire learning process faster. Finally, there is a computational cost for performing the dimensionality reduction. However, in both SoCRATE and SoCRATE-AUTO+GI, this cost is regained when the spectral variables make way for their principal components (or autoencoder features) in the subsequent steps of the learning process. All the previous considerations contribute to assessing that all the learning components of the pre-processing of the defined methodology actually characterize the effectiveness of SoCRATE, expressed in terms of both accuracy and efficiency.

4.2.2. Post-processing analysis

We proceed by investigating the benefit of the object-wise refinement applied to the pixel-wise classifications during the post-processing step. For this investigation P-SoCRATE denotes the performance of the pixel-wise classifier without any object-wise refinement. Table 5 reports the classification performance achieved with training sample size equal to 5% of the ground-truth sample size. In addition, Figures 7(a)-7(c), Figures 7(d)-7(f) and Figures 7(g)-7(i) show the change in OA, AA and ΔTIME according to the training sample size. ΔTIME denotes the time spent performing the object-wise refinement after the pixel-wise classifications have been determined (i.e. the difference in TIME between SoCRATE and P-SoCRATE).

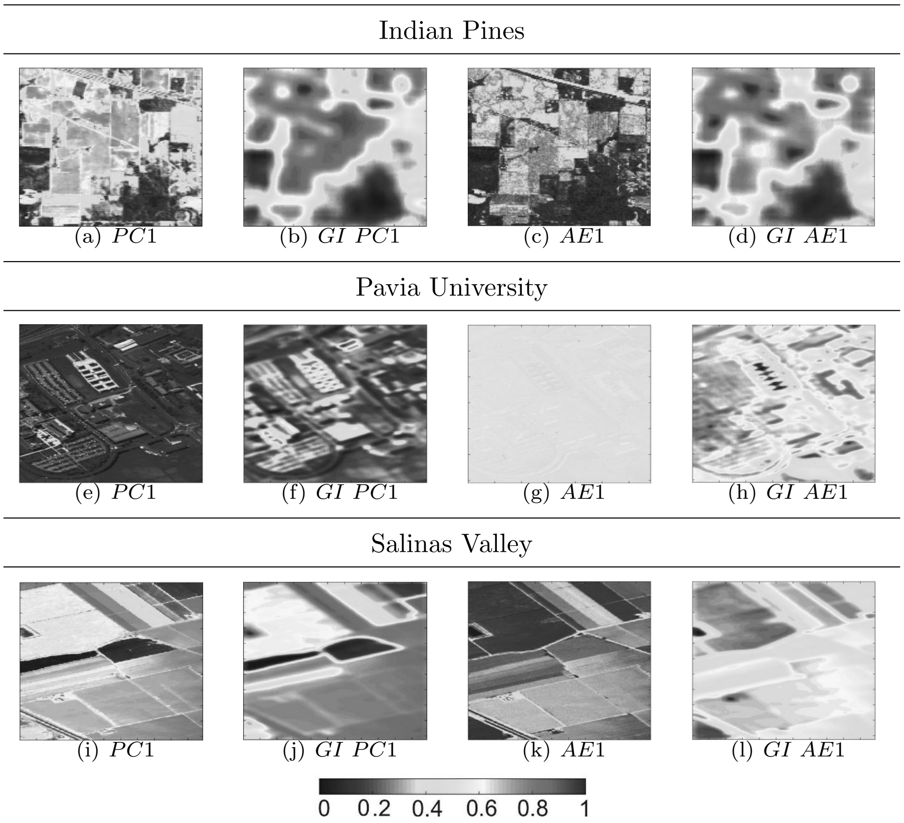
These results show that the time spent refining the pixel-wise classifications

Table 4: Performance (OA, AA, k and TIME) of SoCRATE, SoCRATE-GI, SoCRATE-PCA, SoCRATE-AUTO+GI, SoCRATE-AUTO and SoCRATE-None with $N = 50$, $R = 7$, $\epsilon = 0.03$ for the three datasets. The training sample size is equal to 5% of the ground-truth sample size.

Dataset	Configuration	OA	AA	κ	TIME(secs)
Indian Pines	SoCRATE	.9818	.9793	.9792	474.694
	SoCRATE-GI	.9564	.9172	.9502	957.969
	SoCRATE-PCA	.5390	.5071	.4872	2580.045
	SoCRATE-AUTO+GI	.9634	.9453	.9582	717.944
	SoCRATE-AUTO	.6191	.5542	.5721	2416.036
	SoCRATE-NONE	.6302	.5621	.5881	4048.111
Pavia University	SoCRATE	.9984	.9971	.9979	5928.607
	SoCRATE-GI	.9753	.9592	.9667	11891.068
	SoCRATE-PCA	.7192	.6623	.6306	20579.616
	SoCRATE-AUTO+GI	.9990	.9981	.9987	16656.222
	SoCRATE-AUTO	.9068	.8348	.8736	70461.307
	SoCRATE-NONE	.9021	.8157	.8680	74405.708
Salinas Valley	SoCRATE	.9994	.9993	.9993	608.847
	SoCRATE-GI	.9957	.9972	.9952	434.175
	SoCRATE-PCA	.8653	.8690	.8502	3868.436
	SoCRATE-AUTO+GI	.9982	.9990	.9980	824.780
	SoCRATE-AUTO	.9109	.9506	.9007	1171.230
	SoCRATE-NONE	.9317	.9576	.9241	6393.436

with the segmentation-aided information is negligible (see ΔTIME). On the other hand, the use of the object-wise refinement improves the accuracy of the initial pixel-wise classification, although the strength of this gain in accuracy increases as the training sample size diminishes (see, for example, the maps in Figures 8 per class for Indian Pines). This phenomenon can be easily explained by observing that the accuracy of the pixel-wise classifier naturally increases with the size of the labeled training sample. Therefore, starting from more accurate pixel-wise classifiers makes the request of exploiting object-aided information less substantial, in order to correct fictional spurious pixel-wise classifications. This consideration is also confirmed by analyzing the accuracy (true positive rate - recall) computed per class.

The recall results are reported in Table 6 for P-SoCRATE and SoCRATE, Indian Pines dataset, with both 1% and 5% of ground-truth samples selected during the segmentation-aided training phase. Once again these results high-



(a) $PC1$ (b) $GI\ PC1$ (c) $AE1$ (d) $GI\ AE1$

(e) $PC1$ (f) $GI\ PC1$ (g) $AE1$ (h) $GI\ AE1$

(i) $PC1$ (j) $GI\ PC1$ (k) $AE1$ (l) $GI\ AE1$

Figure 4: Heatmaps of $PC1$ (Figures 4(a),4(e) and 4(i)), $GI\ PC1$ (Figures 4(b),4(f) and 4(j)), $AE1$ (Figures 4(c),4(g) and 4(k)), $GI\ AE1$ (Figures 4(d),4(h) and 4(l)), for Indian Pines, Pavia University and Salinas Valley datasets.

light the benefit of the object-wise refinement when the training sample size is equal to only 1% of the ground-truth sample size. In this case, the object-wise refinement outperforms (or performs equally to) the pixel-wise classification for each class. On the other hand, the object-wise refinement slightly worsens the classifications for six out of sixteen land-cover classes ("Alfaalfa", "Corn-notill", "Corn-mintill","Grass pasture", "Soybean-notill" and "Stone-Steel-Towers"), when the training sample size is equal to 5% of the ground-truth sample size. It outperforms (or performs equally to) the pixel-wise classification of the pixels in the remaining land cover classes. We note that two groups of worsened classifi-

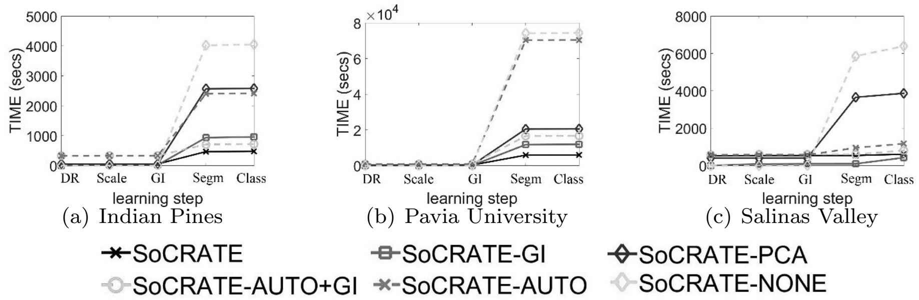
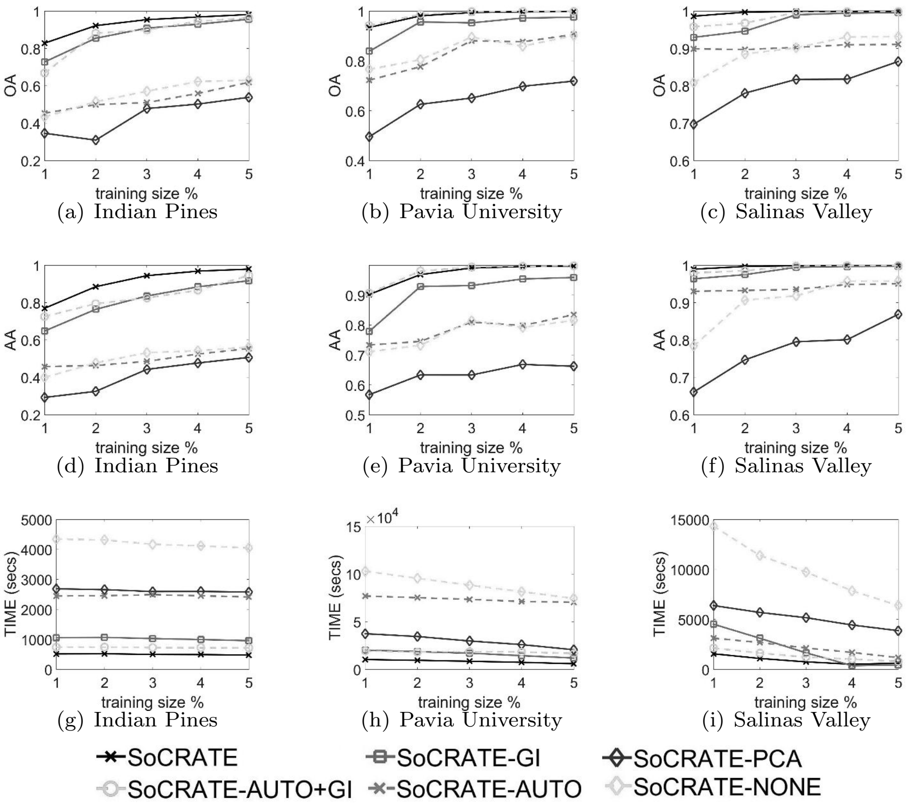


Figure 5: Progress report of the cumulative computation time (axis Y) with the steps of the learning process (axis X - *DR* denotes the dimensionality reduction (PCA or autoencoder), *GI* denotes the spectral-spatial analysis, *Scale* denotes the scaling step, *Segm* denotes the segmentation and *Class* denotes the classification) performed by the compared configurations (SoCRATE, SoCRATE-GI, SoCRATE-PCA, SoCRATE-AUTO+GI, SoCRATE-AUTO and SoCRATE-NoPCAGI) for Indian Pines, Pavia University and Salinas Valley datasets.

Table 5: Performance (OA, AA, k, TIME and ΔTIME) of both SoCRATE and P-SoCRATE with $N = 50$, $R = 7$ and $\epsilon = 0.03$ for the three datasets. The training sample size is equal to 5% of the ground-truth sample size.

Dataset	Configuration	OA	AA	κ	TIME(secs)	ΔTIME
Indian Pines	SoCRATE	.9818	.9793	.9792	474.518	.176
	P-SoCRATE	.9816	.9780	.9790	474.694	
Pavia University	SoCRATE	.9984	.9971	.9979	5928.607	.431
	P-SoCRATE	.9982	.9968	.9976	5928.176	
Salinas Valley	SoCRATE	.9994	.9993	.9993	608.584	.246
	P-SoCRATE	.9994	.9994	.9993	570.270	

cations concern pixels of minority classes (“Alfaalfa” and “Stone-Steel-Towers”). They contribute to the calculation of the final average accuracy, independently of the class cardinality. This analysis confirms that, also in this particular case, the object-wise refinement may yield an overall advantage. However, it also suggests the need for further investigation to improve the object-wise set-up in the presence of minority classes.



(a) Indian Pines (b) Pavia University (c) Salinas Valley

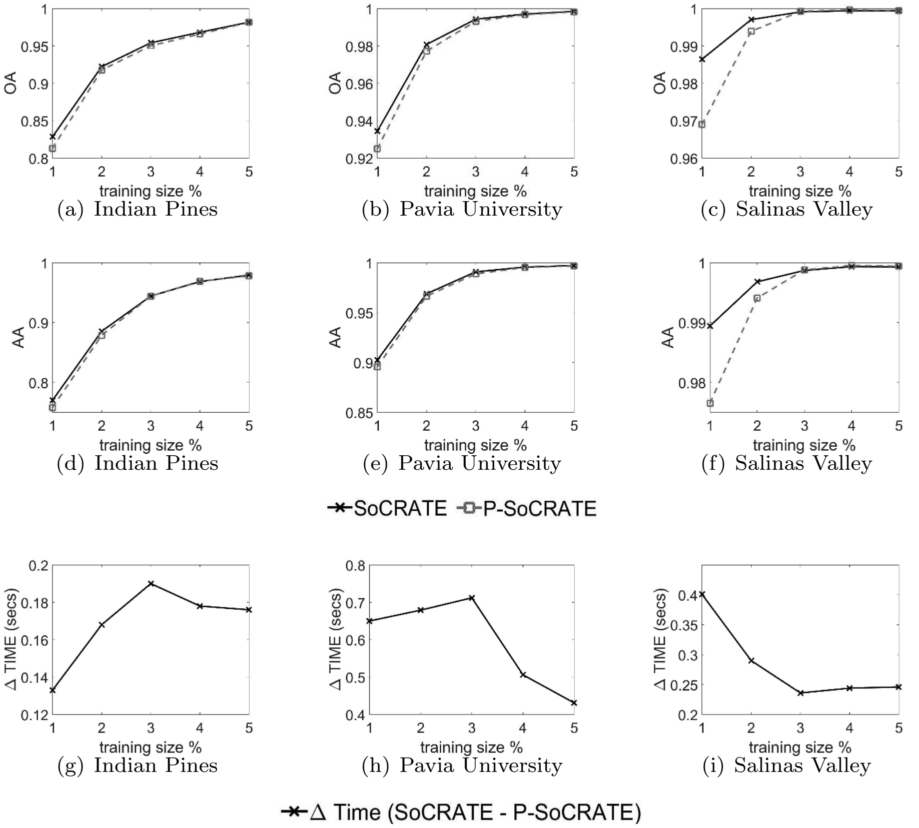
(d) Indian Pines (e) Pavia University (f) Salinas Valley

(g) Indian Pines (h) Pavia University (i) Salinas Valley

Figure 6: OA (axis Y - Figures 6(a)-6(c)), AA (axis Y - Figures 6(d)-6(f)) and TIME (axis Y - Figures 6(g)-6(i)) of SoCRATE, SoCRATE-GI, SoCRATE-PCA, SoCRATE-AUTO+GI, SoCRATE-AUTO and SoCRATE-NoPCAGI, run with $N = 50$, $R = 7$, $\epsilon = 0.03$ and training sample size (axis X) ranging among 1%, 2%, 3%, 4% and 5% of the ground-truth sample size for the three datasets.

4.2.3. Training sample selection

We investigate how the use of the segmentation-aided information actually conditions the training sample set definition. In the absence of segmentation information and without any prior knowledge of the ground-truth class distribution, the training pixels should be *randomly* selected across the image dataset. So, we repeat the random sampling procedure on ten trials and compare the average performance achieved by the pixel-wise classifiers, learned from the ran-



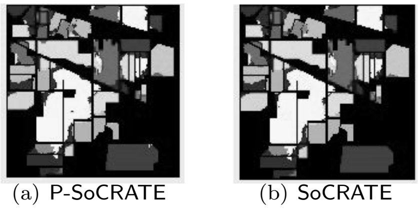
(a) Indian Pines (b) Pavia University (c) Salinas Valley

(d) Indian Pines (e) Pavia University (f) Salinas Valley

(g) Indian Pines (h) Pavia University (i) Salinas Valley

Figure 7: OA (axis Y - Figures 7(a)-7(c)), AA (axis Y - Figures 7(d)-7(f)) and ΔTIME (axis Y - Figures 7(g)-7(i)) of both SoCRATE and P-SoCRATE, run with $N = 50$, $R = 7$, $\epsilon = 0.03$ and training sample size (axis X) ranging among 1%, 2%, 3%, 4% and 5% of the ground-truth sample size for the three datasets.

dom training sample sets, to the performance of both the pixel-wise classifier (P-SoCRATE) and the object-wise refinement (P-SoCRATE), which account for the segmentation-aided information in the selection of the training samples. We note that, as the object-wise refinement is unavailable without the knowledge embedded in the segmentation output, the performance of the pixel-wise classifier is the only option that can be considered with the random sampling procedure.



(a) P-SoCRATE (b) SoCRATE

Figure 8: Pixel-wise classification map (P- SoCRATE - Figure 8(a)) and object-wise classification map (SoCRATE - Figure 8(b)) for the Indian Pines dataset, with training sample size equal to 1% of the ground-truth size.

Table 6: Number of training samples ($\#Tr$) automatically selected using the segmentation-stratified sampling procedure; true positive rate (recall) per class of both SoCRATE and P-SoCRATE, with N= 50, $R = 7$, $\epsilon = 0.03$ for Indian Pines. Training size =1% (columns 2-4) and 5% (columns 5-7). The highest accuracy is in bold.

Class	$\#Tr$	P-SoCRATE	SoCRATE	$\#Tr$	P-SoCRATE	SoCRATE
Training size		1%			5%	
Alfaalfa	2	**1.00**	**1.00**	4	**.952**	.928
Corn-notill	14	.848	**.860**	76	**.978**	.975
Corn-mintill	6	.507	**.529**	34	**.948**	.944
Corn	2	**.782**	**.782**	13	**1.0**	**1.0**
Grass-pasture	7	.829	**.831**	27	**.993**	.991
Grass-trees	10	.844	**.847**	38	.966	**.968**
Grass-pasture-mowed	1	**.851**	**.851**	2	**.961**	**.961**
Hay-windrowed	9	.950	**.961**	36	**1.00**	**1.00**
Oats	0	**.000**	**.000**	2	.944	**1.00**
Soybean-notill	8	.677	**.692**	44	**.984**	.983
Soybean-mintill	16	.871	**.888**	92	.983	**.986**
Soybean-clean	7	.726	**.781**	44	.961	**.967**
Wheat	1	**.504**	**.504**	9	**1.00**	**1.00**
Woods	14	.984	**.992**	62	.990	**1.00**
Building-Grass-Trees	4	.801	**.853**	23	**.997**	**.997**
Stone-Steel-Towers	2	**.945**	**.945**	7	**.976**	.965
AA	103	.757	**.770**	513	.978	**.979**

This experiment is performed with the baseline parameter configuration ($N = 50$, $R = 7$, $\epsilon = 0.03$). The accuracy results, reported in Table 7, show that the object information available with segmentation significantly contributes to optimally selecting the training samples which actually increase the level of accuracy of the pixel-wise classifier. The only exception occurs with Salinas Valley when only 1% of ground-truth data is selected to populate the training sample set. Although the segmentation-based sampling still yields the higher AA by augmenting the number of correct classifications for the minority classes, the OA of the pixel-wise classifier learned with random sampling is slightly greater than the OA of the pixel-wise classifier learned with segmentation-aided sampling. In any case, also in this isolated scenario, the post-processing procedure allows us to correct the spurious classifications by contributing to restoring the superiority of the performance of classification achieved with segmentation. This confirms that the use of the segmentation information yields twin advantages in the presented methodology: it contributes to the selection of appropriate pixels for training sample sets, aiding accurate learning, and it helps in the correction of spurious pixel-wise misclassifications that are more frequent, as the classifiers are learned from the smaller training sample sets.

Further considerations are derived from a qualitative analysis of the pixels identified by the segmentation-stratified sampling. Table 6 (columns 2 and 5) reports the class distribution of the pixels sampled per segments in Indian Pines when 1% and 5% of the pixels are selected for the training set. Segmentation knowledge is sufficient to select also pixels that belong to several minority classes for the training set even when a very small training set (i.e. 1% of the ground-truth) is considered. In this extreme configuration, only the minority class “Oat” will not have any samples for the training, while the class “Wheat” will have only one sample, which is not sufficient to avoid under-fitting, as proved by the small recall (see Table 6). Instead the minority classes “Alfaalfa”, “Grass-pasture-mowed” and “Stone-Steel” are sufficiently represented also in the very small training set, reducing the risk of a serious under-fitting of the training classifier for these classes. On the other hand, the training classifier learned using the very small training set unexpectedly suffers from under-fitting with classes “Corn-mintill” and “Soybean-notill”. These classes cannot be properly considered minority classes as they theoretically have a sufficient number of ground-truth samples in the HSI dataset (830 for ‘Corn-mintill” and 972 for “Soybean-notill”). However, the segmentation-stratified sampling identifies a small number of training samples for them (6 for “Corn-mintill” and 8 for “Soybean-notill”). As pointed out in [13], corn and soybean are very challenging land-cover classes to discriminate as they are very early in their growth cycle. This suggests that an under-segmentation phenomenon occurs during the agglomerative merging phase of the segmentation, when this is performed in the configuration requested to populate the very small training set. This undersegmentation reasonably fails in the correct identification of the segments that correctly delineate these crop objects and, consequently, in the selection of a sufficient number of samples, in order to learn an accurate classifier also for these challenging classes. This analysis paves the way for future investigations, in order to improve the performance of both the segmentation and the entire classification methodology, when the learning is done under very small sample conditions and in the presence of a very challenging task.

To conclude this study of the training sample selection, we note that additional empirical evidence of the effectiveness of the segmentation-stratified training sample selection is also yielded by the analysis of the performance of several state-of-the-art spectral-spatial competitors, reported in Section 4.4. In fact, according to the author guidelines, the accuracy performance of these competitors is commonly evaluated considering training sets that are ideally constructed by selecting pixels per class (using the ground-truth knowledge). Our analysis highlights that SoCRATE, even without using the ground-truth on the class distribution to select the training set, generally outperforms these competitors.

4.2.4. Final remarks

We conclude this analysis by summarizing the conclusions drawn from the study on the actual gain in accuracy yielded by the spectral-spatial feature construction and the segmentation knowledge processing in the proposed methodology. Table 8 collects the results of OA and AA with training sample size equal to 5% of the ground-truth sample size. The results show that the configuration with both components (Spectral-spatial - Training+Post) effectively con-

Table 7: Random sampling vs Segmentation-aided sampling. The learning process is performed with $N = 50$, $R = 7$ and $\epsilon = 0.03$. The random sampling is repeated on 10 trials. The training sample size ranges between 1% and 5% of the ground-truth sample size for the three datasets.

Dataset	Training size	Random (avg±stdev)		P-SoCRATE		SoCRATE	
		OA	AA	OA	AA	OA	AA
Indian Pines	1%	.7278±.0096	.5874±.0373	.8130	.7579	.8286	.7701
	2%	.8460±.0125	.7508±.0212	.9177	.8786	.9223	.8854
	3%	.8953±.0123	.8186±.0264	.9502	.9442	.9542	.9443
	4%	.9257±.0085	.8466±.0376	.9659	.9690	.9682	.9684
	5%	.9407±.0056	.9017±.0338	.9816	.9780	.9818	.9793
Pavia University	1%	.9191±.0088	.8377±.0223	.9249	.8955	.9344	.9023
	2%	.9529±.0058	.9017±.0199	.9772	.9664	.9807	.9688
	3%	.9709±.0023	.9357±.0079	.9932	.9888	.9943	.9908
	4%	.9783±.0037	.9531±.0086	.9967	.9953	.9970	.9956
	5%	.9834±.0014	.9639±.0035	.9982	.9968	.9984	.9971
Salinas Valley	1%	.9769±.0048	.9748±.0060	.9690	.9765	.9864	.9894
	2%	.9891±.0020	.9894±.0021	.9939	.9941	.9970	.9968
	3%	.9944±.9E-5	.9928±.0010	.9992	.9988	.9991	.9987
	4%	.9970±.4E-5	.9956±.0010	.9996	.9995	.9994	.9993
	5%	.9969±.6E-5	.9956±.0013	.9994	.9994	.9994	.9993

tribute to a gain in accuracy with respect to the baseline, that eliminates both the feature construction and the contiguity-constrained segmentation (Spectral+NoSegm). In any case, the spectral-spatial feature construction clearly introduces more effectiveness in the methodology (see results of Spectral vs Spectral-spatial), while the segmentation introduces an improvement only if it is computed over the spectral-spatial features. This is a consequence of the fact that the proposed segmentation procedure, decoupled from the spatial smoothing of the spectral information, introduced with the local indicator computation, may suffer in the presence of possible spectral noise. Therefore, the segmentation patterns, learned without the spatial smoothing, fail to identify the optimal training sample set for learning the classifier (Spectral-NoSegm vs Spectral-Training), although they still help to correct a few spurious classifications in the post-processing phase (Spectral-Training vs Spectral-Training+Post).

Table 8: Performance (OA and AA) of configurations of SoCRATE generated along the feature construction (spectral – no feature construction, spectral-spatial - feature construction with PCA and GI) and the segmentation knowledge processing (NoSegm – random training set sampling and no outlier post processing, Training – segmentation-aided training set sampling and no outlier post processing, Training+Post – segmentation-aided set sampling and no outlier post processing).

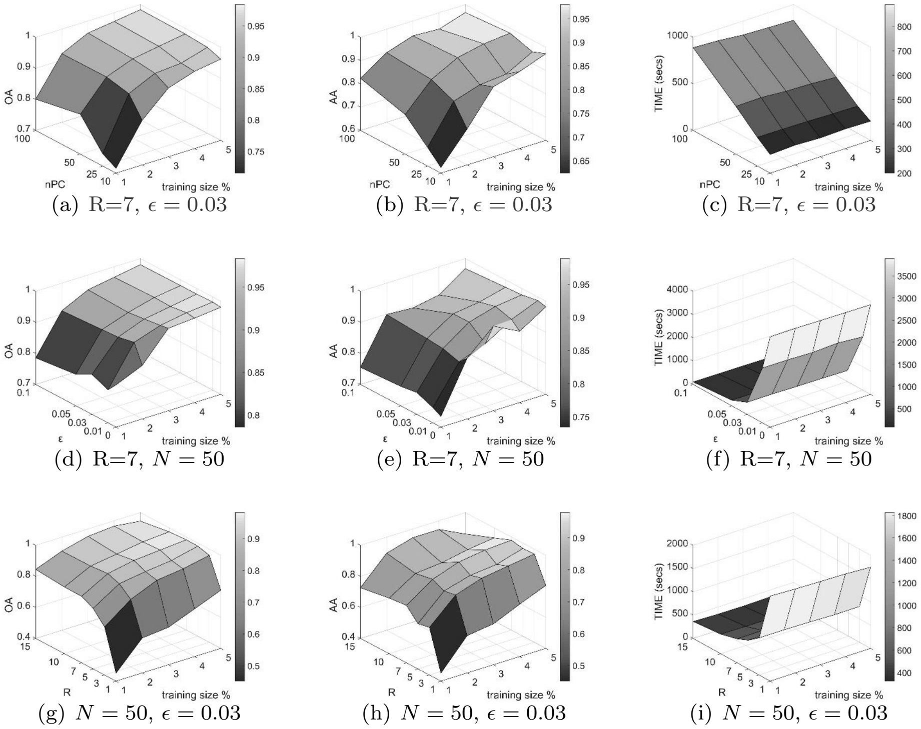
Feature / Segmentation	dataset	Metric	NoSegm	Training	Training+Post
Indian Pines	OA	Spectral	.6672	.6087	.6302
		Spectral-spatial	.9407	.9816	.9818
	AA	Spectral	.5762	.5460	.5621
		Spectral-spatial	.9017	.9780	.9793
Pavia University	OA	Spectral	.9237	.8795	.9021
		Spectral-spatial	.9834	.9982	.9984
	AA	Spectral	.8878	.8027	.9021
		Spectral-spatial	.9639	.9968	.9971
Salinas Valley	OA	Spectral	.9199	.9235	.9317
		Spectral-spatial	.9969	.9994	.9994
	AA	Spectral	.9549	.9235	.9576
		Spectral-spatial	.9956	.9994	.9993

4.3. Sensitivity analysis

For this analysis we consider the Indian Pines data set that, according to considerations formulated by Plaza et al. [13], is a very challenging classification problem. We perform a sensitivity analysis of the performance of SoCRATE along the number of principal components (with $N = 10$, 25, 50 (baseline) and 100), the size of the spatial neighborhood (with $R = 1, 3, 5, 7$ (baseline), 10 and 15), the segmentation similarity threshold (with $\epsilon = 0.0, 0.03$ (baseline), 0.05 and 0.1) and the base classification learner (J48, Naive Bayes - NB, Random Forests - RF, Logistic Regression - LR and Support Vector Machine - SVM (baseline))[7]. We analyze the change of OA, AA and TIME on the training sample size, ranging between 1% and 5% of the ground-truth sample size.

[7] The Java implementation of J48, NB, RF, LOG and SVM, included in the 3.6 WEKA toolkit

N, R and ϵ. For this study we consider SVM as the base classification learner of SoCRATE. Initially we vary N among 10, 25, 50 (baseline) and 100, while we run SoCRATE with baselines $R = 7$ and $\epsilon = 0.03$. The computed metrics are reported in Figures 9(a) (OA), 9(b) (AA) and 9(c) (TIME). We note that the dimensionality reduction contributes to improving the efficiency of the entire learning process - the lower the number of principal components, the lower the computation time spent completing the learning process. According to the considerations already reported in Section 4.2, this gain in efficiency is due to the fact that processing the principal components speeds up the segmentation and classification steps by recovering the computational time spent performing the PCA. On the other hand, the accuracy increases with the number of principal components, although the gain in accuracy is approximately stable after at least 50 principal components have been saved. Even though we observe that 10 PCs explain 99% variance of the HSI dataset, this experiment shows that the consideration of a higher number of PCs in the proposed methodology may increase not only the complexity of the model but also the classification accuracy. This gain in accuracy is particularly notable with the smaller training sets. Therefore, $N = 50$ is considered a reasonable realization of the trade-off between efficiency and accuracy. Then we vary ϵ among 0.0, 0.01, 0.03, 0.05 and 0.1, while we run SoCRATE with baselines $R = 7$ and $N = 50$. The computed metrics are reported in Figures 9(d) (OA), 9(e) (AA) and 9(f) (TIME). These results show that the segmentation similarity threshold has an important impact on the efficiency - the higher the segmentation threshold, the lower the computation time. However, the segmentation threshold has only a slight impact on the accuracy. In any case, independently of the training sample size, the metrics measuring the accuracy are approximately stable with $\epsilon \geq .03$. Finally, we vary R among 1, 3, 5, 7, 10 and 15, while we run SoCRATE with baselines $N = 50$ and $\epsilon = 0.03$. The computed metrics are reported in Figures 9(g) (OA), 9(h) (AA) and 9(i) (TIME). Both the efficiency and accuracy generally improve with R, that is, the augmented consideration of the smoothing effect in the representation of the spectral-spatial information contributes to speeding



(a) R=7, $\epsilon = 0.03$ (b) R=7, $\epsilon = 0.03$ (c) R=7, $\epsilon = 0.03$

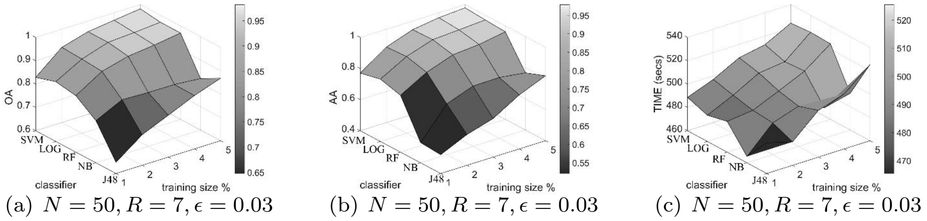
(d) R=7, $N = 50$ (e) R=7, $N = 50$ (f) R=7, $N = 50$

(g) $N = 50$, $\epsilon = 0.03$ (h) $N = 50$, $\epsilon = 0.03$ (i) $N = 50$, $\epsilon = 0.03$

Figure 9: Sensitivity of OA, AA and TIME of SoCRATE to N with fixed $R = 7$ and $\epsilon = 0.03$ (Figures 9(a), 9(b) and 9(c)). Sensitivity to ϵ with $R = 7$ and $N = 50$ (Figures 9(d), 9(e) and 9(f)). Sensitivity to R with $N = 50$ and $\epsilon = 0.03$ (Figures 9(g), 9(h) and 9(i)). The training sample size ranges among 1%, 2%, 3%, 4% and 5% of the ground-truth sample size of the Indian Pines dataset.

up the learning process and generally improves the accuracy. However, there are configurations where the accuracy slightly decreases after the peak between $R = 7$ and $R = 10$. Once again $R = 7$ can produce the trade-off between efficiency and accuracy.

Base classification learner. We consider the baseline parameter configuration ($N = 50$, $R = 7$ and $\epsilon = 0.03$) and evaluate the change in the performance of SoCRATE to the selection of the base classification learner. The computed metrics are reported in Figures 10(a) (OA), 10(b) (AA) and 10(c) (TIME). These



(a) $N = 50, R = 7, \epsilon = 0.03$ (b) $N = 50, R = 7, \epsilon = 0.03$ (c) $N = 50, R = 7, \epsilon = 0.03$

Figure 10: Sensitivity of OA - Figure 10(a), AA - Figure 10(b) and TIME - Figure 10(c) of SoCRATE to the selection of the base learner. SoCRATE is run with the baseline parameter configuration $N = 50$, $R = 7$, $\epsilon = 0.03$ and the training sample size ranging among 1%, 2%, 3%, 4% and 5% of the ground-truth sample size of the Indian Pines dataset.

results show that SoCRATE with SVM outperforms SoCRATE with J48, NB, RF and LOG. In any case, the higher accuracy observed is achieved at the expense of a higher computational cost. The analysis also reveals that LOG and RF can be considered valuable options for the hyperspectral classification methodology presented here. Both are slightly less accurate than SVM, but their computation is faster. This final consideration is not surprising as LOG has already been used as the base classification learner of the spectral-spatial iterative methodology presented in [58], while RF has been used as the base classification learner of both the spectral-spatial ensemble methodology described in [20] and the spectral-spatial segmented stacked autoencoder illustrated in [55].

4.4. Comparison with state-of-the-art competitors

Hyperspectral classification has received a great deal of attention in the recent literature. In this section various recent competitive spectral-spatial classification methods [17, 22, 23, 24, 20, 55, 25, 26, 15, 32, 30, 59, 60, 61] are considered (see a short description in Table 9). These competitors have been evaluated on Indian Pines, Pavia University and Salinas Valley in the referenced studies. The optimal accuracy performance described in the literature for each competitor is reported in Table 10. We note that each competitor has been evaluated starting with a specific number of training samples selected per class, except for [32], where an active learning strategy is developed, in order to acquire the labels. In general, the training sample sets of these state-of-the art experiments have been populated by accounting for the prior information available in the ground-truth.

For each competitor and for every experimental configuration considered by the competitor, the accuracy performance of SoCRATE is here evaluated by considering the training sample size equal to the total number of training samples already used to evaluate the competitor. However, according to the methodology presented in this paper, SoCRATE selects the samples that will populate the training sample set by accounting for the segmentation information, while neglecting the ground-truth that is properly used in the evaluation phase (i.e. to compute the accuracy of the classification of the testing pixels). This analysis is an additional empirical proof of the effectiveness of the entire methodology proposed here.

SoCRATE achieves a better performance (highlighted in Table 10 for the various configurations of the three datasets) compared to all the other competitors tested in this study. In particular, SoCRATE outperforms the performance of various deep learning architectures (e.g. [22, 23, 24, 25, 26, 59]), multi-view iterative collective inference methodologies [17], segmentation-aided classifiers [55, 32, 30], multi-profile-aware ensembles [20, 17], as well as spectral-spatial classifiers with the widely used extended morphological profile - EMP [15, 60, 61]. There are only four configuration settings where one of the competitors outperforms SoCRATE. Specifically, SoCRATE loses average accuracy (.886 vs .907) when processing a very small training set of Indian Pines (i.e. 240 pixels), although it still gains overall accuracy in this configuration compared to the ensemble with the extended multiextinction profiles [20] (.938 vs .881). This accuracy performance means that SoCRATE fails in the correct classification of a few minority class pixels, while it improves the accuracy of the classifications yielded for pixels in the remaining majority classes compared to its competitor. In any case, this competitor can be fully outperformed if we change the baseline parameter set-up and run SoCRATE with a segmentation threshold

$\epsilon = 0.05$ (instead of 0.03). In this configuration SoCRATE achieves OA=.935 and AA= .914. A similar accuracy performance is observed with a competitor deploying deep learning on multi-grained networks [26]. This competitor is also evaluated with a very small training set of Indian Pines (i.e. 304 pixels). In this configuration SoCRATE loses average accuracy (.934 vs .954) by failing in the classification of a few minority class pixels, but it gains overall accuracy (.947 vs .906) by correctly classifying pixels belonging to the majority classes. The convolutional neural network with diverse region-based inputs and feature learning [25] achieves higher overall accuracy than SoCRATE only in one out of the five configurations tested for Pavia University (i.e. when only 450 pixels are selected for training). On the other hand, SoCRATE outperforms this specific competitor in all remaining configurations of Pavia University and all tested configurations for Indian Pines and Salinas Valley. Finally, we note that the competitor deploying segmentation combined with active and iterative learning [32] outperforms SoCRATE in Pavia University, although the same competitor still performs worse than SoCRATE in Indian Pines. While SoCRATE exploits the geometric shape of the segments, in order to identify all pixels whose labels are acquired from the oracle, the active learner in [32] derives an estimate of the reliability of predictions across segments, in order to identify iteratively the pixels to query the oracle. Interestingly this solution, that allows the competitor to gain in accuracy with few labels acquired in Pavia University, has difficulty classifying Indian Pines under the same conditions. Probably, the presence of an imbalanced number of pixels per class in Indian Pines compromises the reliability estimation, thus weakening the active learning. In any case, this result paves the way for further investigation of active learning applied to the methodology described in this study.

In general, the combination of PCA and spatial dependency analysis, described in this paper, engineers features as accurately as complex deep learning architectures, without suffering from the high computational costs of deep neural networks. This is an interesting milestone achieved in this study, considering that nowadays deep learning represents one the most prominent frontiers of re-

Table 9: State-of-the-art compared competitors: spectral-spatial competitor reference (column 1) and description (column 2).

Competitor	Description
[17](2017)	Multi-view collective inference, semi-supervised learning, frequency and morphological profiles,
[22](2018)	Convolutional neural networks, spectral-spatial feature learning
[23](2018)	End-to-end residual network
[24](2018)	ConvDeconv network, unsupervised, spectral-spatial feature learning
[20](2018)	Random Forest ensembles, extended multiextinction profile
[55](2018)	Segmented stacked autoencoder, segmentation-based mutual information
[25](2018)	Convolutional neural networks, diverse region-based inputs, feature learning
[26](2017)	Deep learning, multi-grained network
[15](2014)	Spectral-spatial classification, extended morphological profiles
[32](2016)	Segmentation, active learning, semi-supervised learning
[30](2010)	Segmentation, multiple spectral-spatial classification
[59] (2017)	Autoencoder, deep learning, self-taught learning
[60] (2016)	Multi-kernel learning, extended morphological profiles
[61] (2015)	Extreme earning machine, extended morphological profiles

cent research in machine learning and remote sensing. In addition, it elegantly selects the training pixels without requiring prior information on ground-truth class distribution. In any case, further investigation is advocated, in order to avoid possibly incorrect classifications that may still arise in the presence of minority classes with very small training sets.

5. Conclusion

A new methodology for spectral-spatial classification of HS data is described. The use of a local indicator of spatial dependency of HS bands for segmentation-aided classification is proposed for the first time. The local indicators are extracted from the principal components of the spectral bands. In particular, this study also promotes the computation of a local indicator of spatial dependency to incorporate a spatial regularization of the spectral information in the segmentation and classification approach, which was not investigated earlier.

There are several reasons for using PCA, local indicators of spatial dependency and segmentation in this study. Firstly, PCA contributes to dealing with the curse of dimensionality, handling collinearity at near spectral bands, re-

Table 10: Comparison with the recent state-of-the-art hyperspectral classification competitors (details in Table 9). The set-up includes the competitor reference, the dataset (IP - Indian Pines, IP8 - Indian Pines, where the smallest eight classes are discarded, while the eight largest classes are selected, PU - Pavia University and SV - Salinas Valley) and the training sample size. A star (*) is reported if the competitor outperforms SoCRATE.

Set-up			Competitor		SoCRATE		Set-up			Competitor		SoCRATE	
			OA	AA	OA	AA				OA	AA	OA	AA
[17]	IP	513	.961	.898	.981	.979	[22]	IP	2715	.975	.985	.999	.998
	PU	2139	.985	.958	.998	.997		PU	2250	.996	.996	.998	.998
	SV	2707	.994	.997	.999	.999		SV	4800	.983	.993	.999	.998
[23]	IP	3080	.991	.989	.998	.998	[24]	IP	695	.857	.922	.988	.988
	PU	8562	.997	.996	.999	.999		PU	3921	.873	.843	.999	.999
[20]	IP	695	.922	.946	.988	.988	[55]	IP	1021	.966	.974	.995	.997
	IP	240	.881	.907*	.938	.886		IP	2042	.982	.986	.997	.997
	PU	3912	.963	.979	.999	.999		PU	2138	.966	.962	.998	.997
	PU	2139	.910	.932	.998	.997		PU	4276	.974	.970	.999	.999
[25]	IP8	400	.887	-	.976	.973	[25]	PU	1350	.992	-	.995	.992
	IP8	800	.949	-	.993	.995		PU	1800	.995	-	.997	.995
	IP8	1200	.974	-	.997	.998		SV	800	.934	-	.973	.985
	IP8	1600	.985	-	.997	.998		SV	1600	.955	-	.999	.998
	PU	450	.969*	-	.947	.921		SV	2400	.973	-	.999	.999
	PU	900	.986	-	.984	.973		SV	3200	.983	-	.999	.999
[26]	IP	304	.906	.954*	.947	.934	[15]	PU	3921	.988	.990	.999	.999
[32]	IP	250	.827	.859	.929	.876	[30]	IP	695	.923	.942	.988	.988
	PU	250	.922*	.926*	.869	.876		PU	3921	.979	.985	.999	.999
[59]	IP	513	.966	.945	.981	.979	[60]	IP13	509	.764	-	.984	.988
	PU	2139	.995	.990	.998	.997		IP13	1017	.825	-	.995	.995
	SV	2707	.983	.985	.999	.999		PU	440	.921	-	.946	.920
[61]	IP	695	.928	.950	.988	.988		PU	1318	.951	-	.995	.992
	PU	3850	.996	.996	.999	.999		SV	542	.883	-	.986	.989
								SV	1624	.912	-	.999	.998

ducing the computation time for the segmentation and classification, as well as improving the accuracy for the classification. Secondly, local indicators of dependency extract the most apt representative spectral-spatial features by smoothing the spectral variability at near locations and holding them in discriminating textures with high spectral values from textures with low spectral values. These features have proved to be the most effective component in improving the performance of the proposed methodology. Thirdly, segmentation is a reasonable means for delineating in an unsupervised manner (i.e. without the label information) the distribution of the unknown classes over the imagery scene. Contiguity constraints are used to converge fast on largely similar areal boundaries. The segmentation knowledge helps in the label acquisition for the classification and in the outlier removal performed in the post-classification.

The experiments are performed on three widely used HS datasets, considering only 1%, 2%, 3%, 4% and 5% of the labeled samples from each dataset. They prove that every component of the methodology contributes to the gain in classification accuracy. These experiments investigate the sensitivity of the performance to the set-up of input parameters. The results also reveal that the proposed methodology is able to provide high accuracy for most of the land cover classes, even if few labels are acquired for the classification. Finally, the entire methodology achieves competitive accuracy compared to recent state-of-the-art models (including deep learning models), without requiring computational-demanding learning architectures and achieving improvement of classification performance. In fact, with the encouraging performance of the proposed methodology, precise land use land cover (or cropping pattern) maps can be prepared.

Some directions for further work are still to be explored. Appropriate post-classification mechanisms can be considered, in order to improve the outlier removal for the classification of minority classes. New parallel computation mechanisms can be investigated as an alternative to computation architecture for spectral-spatial feature engineering and segmentation. Finally, segmentation can be integrated into iterative active learning mechanisms for the label acquisition step. This improves the selection of training samples, limiting the risk of under-fitting the training classifier also in the presence of spectral-close land-cover classes, which may be particularly difficult to discriminate.

Acknowledgment

The authors wish to thank Lynn Rudd for her help in reading the manuscript, Nicola Di Mauro for his useful discussion on agglomeration segmentation and for providing the Python code to run AutoEncoding and ReCaS-Bari resource team for providing the infrastructure to run the experimental study.

References

[1] F. Melgani, L. Bruzzone, Classification of hyperspectral remote sensing images with support vector machines, IEEE Transactions on Geoscience and Remote Sensing 42 (8) (2004) 1778–1790.

[2] L. He, J. Li, C. Liu, S. Li, Recent advances on spectral-spatial hyperspectral image classification: An overview and new guidelines, IEEE Transactions on Geoscience and Remote Sensing 56 (3) (2018) 1579–1597.

[3] G. Hughes, On the mean accuracy of statistical pattern recognizers, IEEE Transactions on Information Theory 14 (1) (1968) 55–63.

[4] B. Pan, Z. Shi, X. Xu, Multi-objective based sparse representation classifier for hyperspectral imagery using limited samples, IEEE Transactions on Geoscience and Remote Sensing PP (2018) 1–11.

[5] L. Bruzzone, M. Chi, M. Marconcini, Semisupervised support vector machines for classification of hyperspectral remote sensing images, in: C. Chang (Ed.), Hyperspectral Data Exploitation: Theory and Applications, Wiley Online Library, 2007, pp. 275–311.

[6] S. B. Serpico, L. Bruzzone, A new search algorithm for feature selection in hyperspectral remote sensing images, IEEE Trans. Geoscience and Remote Sensing 39 (7) (2001) 1360–1367.

[7] S. Patra, P. Modi, L. Bruzzone, Hyperspectral band selection based on rough set, IEEE Trans. Geoscience and Remote Sensing 53 (10) (2015) 5495–5503.

[8] F. Feng, W. Li, Q. Du, B. Zhang, Dimensionality reduction of hyperspectral image with graph-based discriminant analysis considering spectral similarity, Remote Sensing 9 (4) (2017) 323.

[9] H. Pu, Z. Chen, B. Wang, G. Jiang, A novel spatial-spectral similarity measure for dimensionality reduction and classification of hyperspectral imagery, IEEE Trans. Geoscience and Remote Sensing 52 (11) (2014) 7008–7022.

[10] D. Akbari, Improving spectral–spatial classification of hyperspectral imagery using spectral dimensionality reduction based on weighted genetic algorithm, Journal of the Indian Society of Remote Sensing 45 (6) (2017) 927–937.

[11] J. Cao, B. Wang, Embedding learning on spectralspatial graph for semisupervised hyperspectral image classification, IEEE Geoscience and Remote Sensing Letters 14 (10) (2017) 1805–1809.

[12] H. Gao, S. Lin, Y. Yang, C. Li, M. Yang, Convolution neural network based on two-dimensional spectrum for hyperspectral image classification, Journal of Sensors 2018 (2018) 13.

[13] A. Plaza, J. A. Benediktsson, J. W. Boardman, J. Brazile, L. Bruzzone, G. Camps-Valls, J. Chanussot, M. Fauvel, P. Gamba, A. Gualtieri, M. Marconcini, J. C. Tilton, G. Trianni, Recent advances in techniques for hyperspectral image processing, Remote Sensing of Environment 113(1) (2009) 110 – 122.

[14] J. Benediktsson, M. Pesaresi, K. Amason, Classification and feature extraction for remote sensing images from urban areas based on morphological transformations, IEEE Transactions on eoscience and Remote Sensing 41 (9) (2003) 1940–1949.

[15] P. Quesada-Barriuso, F. Argello, D. B. Heras, Spectralspatial classification of hyperspectral images using wavelets and extended morphological profiles, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 7 (4) (2014) 1177–1185.

[16] A. Appice, P. Guccione, D. Malerba, Transductive hyperspectral image classification: toward integrating spectral and relational features via an iterative ensemble system, Machine Learning 103 (3) (2016) 343–375.

[17] A. Appice, P. Guccione, D. Malerba, A novel spectral-spatial co-training algorithm for the transductive classification of hyperspectral imagery data, Pattern Recognition 63 (2017) 229–245.

[18] Y. Tarabalka, M. Fauvel, J. Chanussot, J. Benediktsson, Svm- and mrf-based method for accurate classification of hyperspectral images, Geoscience and Remote Sensing Letters, IEEE 7 (4) (2010) 736–740.

[19] Y. Y. Tang, Y. Lu, H. Yuan, Hyperspectral image classification based on three-dimensional scattering wavelet transform, IEEE Transactions on Geoscience and Remote Sensing 53 (5) (2015) 2467–2480.

[20] J. Xia, P. Ghamisi, N. Yokoya, A. Iwasaki, Random forest ensembles and extended multiextinction profiles for hyperspectral image classification, IEEE Transactions on Geoscience and Remote Sensing 56 (1) (2018) 202–216.

[21] X. Xu, Z. Shi, B. Pan, l0-based sparse hyperspectral unmixing using spectral information and a multi-objectives formulation, ISPRS Journal of Photogrammetry and Remote Sensing 141 (2018) 46 – 58.

[22] Q. Gao, S. Lim, X. Jia, Hyperspectral image classification using convolutional neural networks and multiple feature learning, Remote Sensing 10 (2).

[23] Z. Zhong, J. Li, Z. Luo, M. Chapman, Spectral-spatial residual network for hyperspectral image classification: A 3-d deep learning framework, IEEE Transactions on Geoscience and Remote Sensing 56 (2) (2018) 847–858.

[24] L. Mou, P. Ghamisi, X. X. Zhu, Unsupervised spectral-spatial feature learning via deep residual conv-deconv network for hyperspectral image classifi-

cation, IEEE Transactions on Geoscience and Remote Sensing 56 (1) (2018) 391–406.

[25] M. Zhang, W. Li, Q. Du, Diverse region-based cnn for hyperspectral image classification, IEEE Transactions on Image Processing 27 (6) (2018) 2623–2634.

[26] B. Pan, Z. Shi, X. Xu, Mugnet: Deep learning for hyperspectral image classification using limited samples, ISPRS Journal of Photogrammetry and Remote Sensing.

[27] B. Pan, Z. Shi, X. Xu, R-VCANet: A new deep-learning-based hyperspectral image classification method, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 10 (5) (2017) 1975–1986.

[28] L. Wang, C. Shi, C. Diao, W. Ji, D. Yin, A survey of methods incorporating spatial information in image classification and spectral unmixing, International Journal of Remote Sensing 37 (16) (2016) 3870–3910.

[29] Y. Tarabalka, J. Chanussot, J. Benediktsson, Segmentation and classification of hyperspectral images using watershed transformation, Pattern Recognition 43 (7) (2010) 2367 – 2379.

[30] Y. Tarabalka, J. A. Benediktsson, J. Chanussot, J. C. Tilton, Multiple spectralspatial classification approach for hyperspectral data, IEEE Transactions on Geoscience and Remote Sensing 48 (11) (2010) 4122–4132.

[31] M. Fauvel, Y. Tarabalka, J. Benediktsson, J. Chanussot, J. Tilton, Advances in spectral-spatial classification of hyperspectral images, Proceedings of the IEEE 101 (3) (2013) 652–675.

[32] Z. Zhang, E. Pasolli, M. M. Crawford, J. C. Tilton, An active learning framework for hyperspectral image classification using hierarchical segmentation, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 9 (2) (2016) 640–654.

[33] W. Li, S. Prasad, J. E. Fowler, Hyperspectral image classification using gaussian mixture models and markov random fields, IEEE Geoscience and Remote Sensing Letters 11 (1) (2014) 153–157.

[34] J. Li, J. Bioucas-Dias, A. Plaza, Spectral-spatial classification of hyperspectral data using loopy belief propagation and active learning, IEEE Transactions on Geoscience and Remote Sensing 51 (2) (2013) 844–856.

[35] G. Zhang, X. Jia, J. Hu, Superpixel-based graphical model for remote sensing image mapping, IEEE Transactions on Geoscience and Remote Sensing 53 (11) (2015) 5861–5871.

[36] M. Golipour, H. Ghassemian, F. Mirzapour, Integrating hierarchical segmentation maps with mrf prior for classification of hyperspectral images in a bayesian framework, IEEE Transactions on Geoscience and Remote Sensing 54 (2) (2016) 805–816.

[37] G. Licciardi, P. R. Marpu, J. Chanussot, J. A. Benediktsson, Linear versus nonlinear pca for the classification of hyperspectral data based on the extended morphological profiles, IEEE Geoscience and Remote Sensing Letters 9 (3) (2012) 447–451.

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

[39] T. Howley, M. G. Madden, M.-L. OConnell, A. G. Ryder, The effect of principal component analysis on machine learning accuracy with high-dimensional spectral data, Knowledge-Based Systems 19 (5) (2006) 363 – 370, aI 2005 SI.

[40] S. Pravilovic, M. Bilancia, A. Appice, D. Malerba, Using multiple time series analysis for geosensor data forecasting, Inf. Sci. 380 (2017) 31–52.

[41] S. Pravilovic, A. Appice, D. Malerba, Leveraging correlation across space and time to interpolate geophysical data via cokriging, International Journal of Geographical Information Science 32 (1) (2018) 191–212.

[42] D. Charte, F. Charte, S. Garca, M. J. del Jesus, F. Herrera, A practical tutorial on autoencoders for nonlinear feature fusion: Taxonomy, models, software and guidelines, Information Fusion 44 (2018) 78 – 96.

[43] A. Getis, A history of the concept of spatial autocorrelation: A geographer's perspective, Geographical Analysis 40 (3) (2008) 297–309.

[44] B. Boots, Local measures of spatial association, Ecoscience 9 (2) (2002) 168–176.

[45] A. Appice, D. Malerba, Leveraging the power of local spatial autocorrelation in geophysical interpolative clustering, Data Min. Knowl. Discov. 28 (5-6) (2014) 1266–1313.

[46] A. Getis, J. K. Ord, The analysis of spatial association by use of distance statistics, Geographical Analysis 24 (3) (1992) 189–206.

[47] Z. A. Holden, J. S. Evans, Using fuzzy c-means and local autocorrelation to cluster satellite-inferred burn severity classes, International Journal of Wildland Fire 19 (7) (2010) 853–860.

[48] A. Appice, S. Pravilovic, D. Malerba, A. Lanza, Enhancing regression models with spatio-temporal indicator additions, in: M. Baldoni, C. Baroglio, G. Boella, R. Micalizio (Eds.), Proceedings of AI*IA 2013: Advances in Artificial Intelligence - XIIIth International Conference of the Italian Association for Artificial Intelligence, Vol. 8249 of Lecture Notes in Computer Science, Springer, 2013, pp. 433–444.

[49] Y. Gao, J. F. Mas, N. Kerle, J. A. N. Pacheco, Optimal region growing segmentation and its effect on classification accuracy, International Journal of Remote Sensing 32 (13) (2011) 3747–3763.

[50] A. Appice, P. Guccione, D. Malerba, A. Ciampi, Dealing with temporal and spatial correlations to classify outliers in geophysical data streams, Inf. Sci. 285 (2014) 162–180.

[51] A. Appice, A. Ciampi, D. Malerba, Summarizing numeric spatial data streams by trend cluster discovery, Data Min. Knowl. Discov. 29 (1) (2015) 84–136.

[52] C. Cortes, V. Vapnik, Support-vector networks, Machine Learning 20 (3) (1995) 273–297.

[53] C. Chen, W. Li, H. Su, K. Liu, Spectral-spatial classification of hyperspectral image based on kernel extreme learning machine, Remote Sensing 6 (6) (2014) 5795–5814.

[54] J. A. Richards, Remote Sensing Digital Image Analysis: An Introduction, 2nd Edition, Springer-Verlag New York, Inc., 1993.

[55] S. Paul, D. N. Kumar, Spectral-spatial classification of hyperspectral data with mutual information based segmented stacked autoencoder approach, ISPRS Journal of Photogrammetry and Remote Sensing 138 (2018) 265 – 280.

[56] Y. Chen, J. Quan, W. Zhan, Z. Guo, Enhanced statistical estimation of air temperature incorporating nighttime light data, Remote Sensing 8 (656) (2016) 1–23.

[57] L. Anselin, Y. W. Kim, I. Syabri, Web-based analytical tools for the exploration of spatial data, Journal of Geographical Systems 6 (2) (2004) 197–218.

[58] P. Guccione, L. Mascolo, A. Appice, Iterative hyperspectral image classification using spectral-spatial relational features, IEEE Transactions on Geoscience and Remote Sensing 53 (7) (2015) 3615–3627.

[59] R. Kemker, C. Kanan, Self-taught feature learning for hyperspectral image classification, IEEE Transactions on Geoscience and Remote Sensing 55 (5) (2017) 2693–2705.

[60] Y. Gu, T. Liu, X. Jia, J. A. Benediktsson, J. Chanussot, Nonlinear multiple kernel learning with multiple-structure-element extended morphological profiles for hyperspectral image classification, IEEE Transactions on Geoscience and Remote Sensing 54 (6) (2016) 3235–3247.

[61] F. Arguello, D. B. Heras, ELM-based spectral-spatial classification of hyperspectral images using extended morphological profiles and composite feature mappings, International Journal of Remote Sensing 36 (2015) 645–664.