UNSUPERVISED DEEP FEATURE EXTRACTION OF HYPERSPECTRAL IMAGES

Adriana Romero[*]*, Carlo Gatta*[†]*, and Gustavo Camps-Valls*[‡]

[*]Dept. MAiA (UB), Universitat de Barcelona, Spain. adriana.romero@ub.edu
[†]Computer Vision Center (CVC), Universitat Autònoma de Barcelona, Spain. cgatta@cvc.uab.es
[‡]Image Processing Laboratory (IPL), Universitat de València, Spain. http://isp.uv.es, gcamps@uv.es

ABSTRACT

This paper presents an effective unsupervised sparse feature learning algorithm to train deep convolutional networks on hyperspectral images. Deep convolutional hierarchical representations are *learned* and then used for pixel classification. Features in lower layers present less abstract representations of data, while higher layers represent more abstract and complex characteristics. We successfully illustrate the performance of the extracted representations in a challenging AVIRIS hyperspectral image classification problem, compared to standard dimensionality reduction methods like principal component analysis (PCA) and its kernel counterpart (kPCA). The proposed method largely outperforms the previous state-of-the-art results on the same experimental setting. Results show that single layer networks can extract powerful discriminative features *only* when the receptive field accounts for neighboring pixels. Regarding the deep architecture, we can conclude that: (1) additional layers in a deep architecture significantly improve the performance w.r.t. single layer variants; (2) the max-pooling step in each layer is mandatory to achieve satisfactory results; and (3) the performance gain w.r.t. the number of layers is upper bounded, since the spatial resolution is reduced at each pooling, resulting in too spatially coarse output features.

Index Terms— Convolutional networks, deep learning, sparse learning, feature extraction, hyperspectral image classification

1. INTRODUCTION

The high spectral resolution of hyperspectral images allows to characterize the objects of interest with unprecedented accuracy. However, the analysis of these images turns out to be more difficult than standard natural images, especially because of the high dimensionality of the pixels, the particular noise and uncertainty sources observed, the high spatial and spectral redundancy and collinearity, and their potential non-linear nature. Such non-linearities can be related to many factors, including multi-scattering in the acquisition process, heterogeneities at subpixel level, as well as atmospheric and geometric distortions. These characteristics of the imaging process lead to distinct non-linear feature relations since the pixels lie in high dimensional curved manifolds [1, 2].

Extracting expressive spatial-spectral features from hyperspectral images is thus of paramount relevance. While the classical Principal Component Analysis (PCA) [3] is still widely used in practice, a plethora of non-linear dimensionality reduction methods and dictionary learning algorithms have been introduced in the last decades. On one hand, we have witnessed the introduction of many *manifold learning* methods [4]: local approaches for the description of remote sensing image manifolds [5]; kernel-based and spectral decompositions that learn mappings optimizing for maximum variance, correlation, entropy, or minimum noise fraction [6]; neural networks that generalize PCA to encode non-linear data structures via autoassociative/autoencoding networks [7]; as well as projection pursuit approaches leading to convenient Gaussian domains [8]. On the other hand, in recent years, *dictionary learning* has emerged as an efficient way to learn image features in unsupervised settings, which are eventually used for image classification and object recognition: discriminative dictionaries have been proposed for spatial-spectral sparse-representation and image classification [9], sparse kernel networks have been recently introduced for classification [10], sparse representations over learned dictionaries for image pansharpening [11], saliency-based codes for segmentation [12], sparse bag-of-words codes for automatic target detection [13], and unsupervised learning of sparse features for aerial image classification [14].

This paper shows the applicability and potential of a new *unsupervised* method to learn hierarchical sparse feature representations of hyperspectral images: the method trains a deep convolutional model accounting for both spatial and spectral information simultaneously and optimizing for sparsity properties in the feature distribution. More precisely, the method optimizes for both *population* and *lifetime* sparsity, which are concepts drawn from the literature of computational neuroscience [15]. On one hand, population sparsity provides a simple interpretation of the data by representing samples with a large amount of outputs, from which only a small subset are active. On the other hand, lifetime sparsity expects each output to be active only for a few samples. In this paper, deep convolutional architectures are trained efficiently in a greedy layer-wise fashion [16] using the Enforcing Population and Lifetime Sparsity (EPLS) algorithm [17] to learn the filters. The algorithm learns discriminative features *without requiring any meta-parameter tuning*. Moreover, thanks to its computational efficiency, it can learn a large set of parameters. The learned hierarchical representations of the input hyperspectral images are used for pixel classification, where lower layers extract low-level features and higher layers exhibit more abstract and complex representations.

The rest of the paper is organized as follows. Section 2 introduces the main characteristics of the proposed algorithm for unsupervised hierarchical sparse feature extraction. Section 3 compares the proposed algorithm to PCA and kPCA in terms of classification accuracy and their expressive power. We end the paper with some concluding remarks in Section 4.

[*]The work of A. Romero is supported by an APIF-UB grant.
[†]The work of C. Gatta is supported by MICINN under a Ramón y Cajal Fellowship.
[‡]This work was partially funded by the Spanish Ministry of Economy and Competitiveness, under the LIFE-VISION project TIN2012-38102-C03-01.

2. UNSUPERVISED FEATURE LEARNING WITH CONVOLUTIONAL DEEP NETWORKS

This section introduces the field of deep convolutional networks and sparse feature coding. We will define the proposed architecture and unsupervised learning criteria used for the experimental setup.

2.1. Convolutional Deep learning

We use convolutional deep networks as the model for learning features. A convolutional deep network is based on the sequential application of a computation "module" where the output of the previous module is the input to the next one; these modules are called *layers*, see Fig. 1.

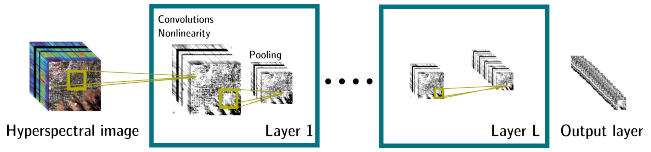

Fig. 1. A graphical representation of a deep architecture.

The input of the first layer is the given data, in this case a hyperspectral image. One layer is composed of three parts: (1) a set of convolutional linear filters, whose parameters can be learned by means of unsupervised or supervised techniques; (2) a point-wise non-linearity, e.g. the logistic function; and (3) a pooling stage, which normally reduces the size of the spatial support of the data and provides a certain local translational invariance, e.g. a non-overlapping 2×2 sliding window computing the maximum of its input (called max-pooling). The rationale of these three parts is (1) to provide a simple local feature extraction method based on convolutions; (2) to modify the result in a non-linear way to allow the deep architecture to learn non-linear representations of the data; and (3) to reduce the computational cost while allowing a local translational invariance in the combination of previously extracted features.

A convolutional deep network can be trained by means of supervised methods, such as the back-propagation algorithm, or greedy layer-wise pre-training as proposed in [16]. Usually, supervised methods require a large amount of labeled data. In the case of multi- and hyperspectral images, it is preferred to use an unsupervised learning strategy method given the typically few available labeled pixels per class. The most relevant parameters in a deep architecture are: (1) the algorithm to learn the convolutional filters; (2) the size of the convolutional filter; (3) the point-wise non-linearity; (4) the type of spatial pooling; (5) the number of layers; and (6) the number of outputs per layer. In the paper, we train deep convolutional networks by means of the criteria described in Section 2.2 using a maximum of 50000 pixels per layer, with a logistic non-linearity and the smallest possible max-pooling (2×2 pixels). For all the deep architectures, we employed the smallest possible symmetric receptive field (of size 3×3 pixels), and 200 outputs per layer. Other parameters are varied for the single layer case, as will be reviewed in the experimental section.

2.2. Unsupervised learning criteria

Unsupervised learning strategies have revealed to be helpful in greedy layer-wise pre-training of deep networks [16]. Methods such as Restricted Boltzmann Machines (RBM) [16], Sparse Auto-Encoders (SAE) [18], Sparse Coding (SC) [19] and Orthogonal Matching Pursuit (OMP-k) [20] have been successfully used in the literature to extract unsupervised feature representations. OMP-k trains a dictionary of bases by iteratively selecting an output of the code to be made non-zero in order to minimize the residual reconstruction error, until at most k outputs have been selected. The method achieves a sparse representation of the input data in terms of *population sparsity*. SAE trains the dictionary bases by minimizing the reconstruction error while ensuring similar activation statistics through all training samples among all outputs, thus not leading to dead outputs and ensuring a sparse representation of the data in terms of *lifetime sparsity*.

In this paper, we use the EPLS algorithm [17], which sets an output target with one "hot code" while ensuring the same mean activation among all outputs and optimizes for that specific target to learn the dictionary bases. Using this approach, we obtain a sparse feature representation of the data, in terms of *both population and lifetime sparsity*, able to discriminate well.

3. EXPERIMENTAL RESULTS

This section illustrates the performance of the proposed method in a challenging hyperspectral image classification problem. We compare the features extracted by networks of varying depth to the ones extracted by PCA and kPCA in terms of expressive power, classification accuracy, and robustness to the number of labeled examples.

3.1. Data Collection

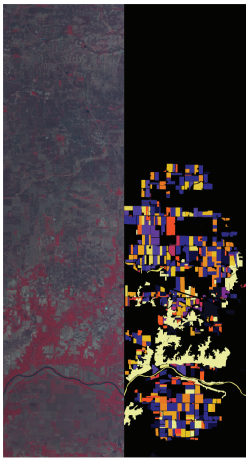

Fig. 2. Color composition (left) and the available reference data (right) for the AVIRIS Indian Pines data set.

The experiments are conducted on the well-known AVIRIS Indiana's Indian Pines test site acquired in June 1992. A small portion (145×145 pixels) of the original image has been extensively used as a benchmark image for comparing classifiers[1]. Here, however, we consider the whole image, which consists of 614×2166 pixels and 220 spectral bands, with a spatial resolution of 20 m. This data set represents a very challenging land-cover classification scenario.

From the 58 different land-cover classes available in the original ground truth, we discarded 20 classes since an insufficient number of training samples were available[2], and thus, this fact would dismiss the planned experimental analysis. The background pixels were not considered for classification purposes. We also removed 20 bands that are noisy or covering the region of water absorption, finally working with 200 spectral bands, cf. Fig. 2.

[1] ftp://ftp.ecn.purdue.edu/biehl/MultiSpec/92AV3C.lan

[2] i.e., less than 1000 samples

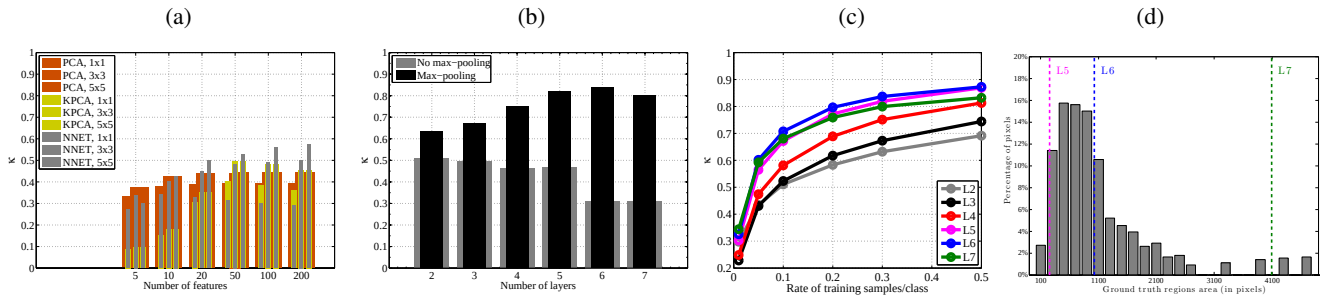

Fig. 3. Classification accuracy estimated with the kappa statistic for (a) several numbers of features, spatial extent of the receptive fields (for the single layer network) or the included Gaussian filtered features (for PCA and kPCA) using 30% of data for training; (b) impact of the number of layers on the networks with and without pooling stages; (c) for different rates of training samples, $\{1\%, 5\%, 10\%, 20\%, 30\%, 50\%\}$, with pooling; and (d) percentage of ground truth pixels as a function of labeled region areas (see text for details).

3.2. Experimental setup

We extract different numbers of features n_f with PCA, kPCA and different structures of the proposed network model, $n_f = \{5, 10, 20, 50, 100, 200\}$, and for different rates of training samples per class, $\{1\%, 5\%, 10\%, 20\%, 30\%, 50\%\}$. For each deep architecture, we train the layers both with and without the pooling stage to assess the effect of the downscaling factor. For kPCA, we use a RBF kernel and set the lengthscale parameter to the average distance between all training samples. Extracted features are then used for classification. For the sake of simplicity, we use the nearest neighbor classifier, and measure accuracy with the estimated Cohen's kappa statistic, κ, in the independent test set made of all the remaining examples.

3.3. Expressive and discriminative power

Figure 3(a) shows the κ statistic for several numbers of extracted features using PCA, kPCA and single layer networks. Both kPCA and the network yield poor results when a low number of features are extracted, and drastically improve their performance for more than 50 features. The neural network results stick around $\kappa = 0.3$ for pixel-wise classification, even with increased number of features. Nevertheless, there is a relevant gain when spatial information is considered. The best results are obtained for 200 features and 5×5 receptive fields. With these encouraging results, we decided to train deeper networks using 30% of the available training samples per class and 200 output features. Results with and without the max-pooling stage are shown in Fig. 3(b). Two main conclusions can be drawn: first, deeper networks improve the accuracy enormously (the 6-layer network reaches the highest accuracy of $\kappa = 0.84$), and second, including the max-pooling stage in each layer revealed extremely beneficial.

We should stress that this result clearly outperforms the previously reported state-of-the-art result ($\kappa = 0.75$) obtained with a Support Vector Machine (SVM) on the same experimental setting [21]. Furthermore, the proposed model largely outperforms SVMs in terms of sparsity computing the rate between model weights and size of the hypercube (24.5% versus 0.81%). Therefore, the learned representation is more accurate and reveals high expressive power.

3.4. Robustness w.r.t. number of training labels

Another question to be addressed is the robustness of the features in terms of training examples. Figure 3(c) reveals that using few samples for training a deep architecture can provide better results than training a single layer network with far more samples. Note, for instance, that the 6-layers net using 5% samples/class outperforms the best single layer net using 30% of the samples/class.

3.5. Need and limitation of spatial pooling

Special attention should be devoted to the 7-layers network. In this case, the accuracy decreases since the potential contribution of an additional layer is strongly counterbalanced by the heavily reduced spatial resolution of the additional max-pooling[3]. To corroborate this explanation, we created the histogram in Figure 3(d), which shows the percentage of ground truth pixels as a function of labeled region areas. As it can also be seen in Figure 2(right), the labeled regions are mainly rectangular with an average area around 500 pixels. Vertical lines in Figure 3(d) show the theoretical spatial resolution in the case the output layer is resized using a nearest neighbor interpolation. As it can be seen, when using 7 layers (L7, green) the resolution is too low to capture regions smaller than 4096 pixels (64×64). It has to be noted that we perform the upscaling of the output layer by means of a bilinear interpolation; this explains why, despite the lower spatial resolution, the result using 6 layers is still superior to the one with 5 layers.

3.6. Learned features

An important aspect of the proposed deep networks lies in the fact that they typically give rise to compact hierarchical representations. The best three features extracted by the networks according to the mutual information with labels are depicted in Fig. 4 for a subset of the whole image. It is worth stressing that the deeper we go, the more complicated and abstract features we retrieve, except for the seventh layer that provides spatially over-regularized features due to the downscaling impact of the max-pooling stages. Interestingly, it is also observed that, the deeper structures we use, the higher spatial decorrelation of the best features we obtain.

4. CONCLUSIONS

We introduced the use of the EPLS algorithm to train deep convolutional networks in a greedy layer-wise fashion and performed experiments to analyze the influence of depth and pooling of such networks on hyperspectral images. The method trains the network parameters

[3]The topmost layer has no max-pooling since it is used as output.

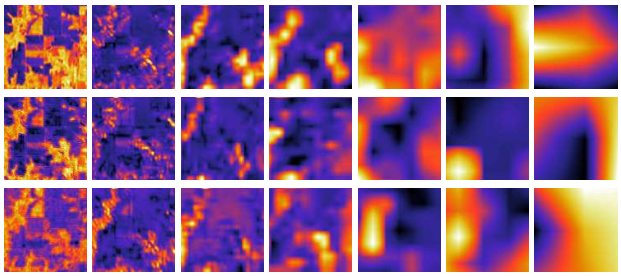

Fig. 4. Best three features (in rows) according to the mutual information with the labels for the outputs of the different layers 1st to 7th (in columns) for a subregion of the whole image.

to learn hierarchical sparse representations of the input hyperspectral images that can be used for classification. Results reveal that the trained networks are very effective at encoding spatio-spectral information of the image. Experiments show that (1) including spatial information is essential in order to avoid poor performance in single layer networks; (2) combining high numbers of output features and max-pooling steps in deep architectures is crucial to achieve excellent results; and (3) adding new layers to the deep architecture improves the kappa agreement score substantially, until the repeated max-pooling steps heavily reduce the features spatial resolution. Further work is tied to assessing generalization of the encoded features in multi-temporal and multi-angular image settings.

5. ACKNOWLEDGMENTS

The authors wish to thank Antonio Plaza from the University of Extremadura, Spain, for kindly providing the AVIRIS dataset used in this paper.

6. REFERENCES

[1] G. Camps-Valls, D. Tuia, L. Gómez-Chova, S. Jiménez, and J. Malo, editors. *Remote Sensing Image Processing*. Morgan & Claypool Publishers, LaPorte, CO, USA, Sept 2011.

[2] G. Camps-Valls, D. Tuia, L. Bruzzone, and J. Atli Benediktsson. Advances in hyperspectral image classification: Earth monitoring with statistical learning methods. *Signal Processing Magazine, IEEE*, 31(1):45–54, Jan 2014.

[3] I.T. Jolliffe. *Principal component analysis*. Springer, 2002.

[4] J. A. Lee and M. Verleysen. *Nonlinear dimensionality reduction*. Springer, 2007.

[5] C.M. Bachmann, T.L. Ainsworth, and R.A. Fusina. Improved manifold coordinate representations of large-scale hyperspectral scenes. *IEEE Trans. Geosci. Remote Sens.*, 44(10):2786–2803, 2006.

[6] J. Arenas-García, K. B. Petersen, G. Camps-Valls, and L. K. Hansen. Kernel multivariate analysis framework for supervised subspace learning: A tutorial on linear and kernel multivariate methods. *IEEE Sig. Proc. Mag.*, 30(4):16–29, 2013.

[7] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, July 2006.

[8] V. Laparra, G. Camps, and J. Malo. Iterative gaussianization: from ICA to random rotations. *IEEE Trans. Neur. Nets.*, 22(4):537–549, 2011.

[9] Z. Wang, N.M. Nasrabadi, and T.S. Huang. Spatial-spectral classification of hyperspectral images using discriminative dictionary designed by learning vector quantization. *IEEE Trans. Geosc. Rem. Sens.*, PP(99):1–15, 2013.

[10] S. Yang, H. Jin, M. Wang, Y. Ren, and L. Jiao. Data-driven compressive sampling and learning sparse coding for hyperspectral image classification. *IEEE Geosc. Rem. Sens. Lett.*, 11(2):479–483, Feb 2014.

[11] S. Li, H. Yin, and L. Fang. Remote sensing image fusion via sparse representations over learned dictionaries. *IEEE Trans. Geosc. Rem. Sens.*, 51(9):4779–4789, Sept 2013.

[12] I. Rigas, G. Economou, and S. Fotopoulos. Low-level visual saliency with application on aerial imagery. *IEEE Geosc. Rem. Sens. Lett.*, 10(6):1389–1393, Nov 2013.

[13] H. Sun, X. Sun, H. Wang, Y. Li, and X. Li. Automatic target detection in high-resolution remote sensing images using spatial sparse coding bag-of-words model. *IEEE Geosc. Rem. Sens. Lett.*, 9(1):109–113, Jan 2012.

[14] A.M. Cheriyadat. Unsupervised feature learning for aerial scene classification. *IEEE Trans. Geosc. Rem. Sens.*, 52(1):439–451, Jan 2014.

[15] B. Willmore and D. J. Tolhurst. Characterizing the sparseness of neural codes. *Network*, 12:255–270, 2001.

[16] G. E. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, July 2006.

[17] A. Romero, P. Radeva, and C. Gatta. No more meta-parameter tuning in unsupervised sparse feature learning. arXiv:1402.5766, 2014.

[18] M. A. Ranzato, C. Poultney, S. Chopra, and Y. Lecun. Efficient learning of sparse representations with an energy-based model. In *NIPS*, pages 1137–1144, 2006.

[19] B. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set: a strategy employed by V1? *Vision Research*, 37(23):3311–3325, 1997.

[20] A. Coates and A. Ng. The importance of encoding versus training with sparse coding and vector quantization. In *ICML*, pages 921–928, 2011.

[21] F. García-Vílchez, J. Muñoz-Marí and, M. Zortea, I. Blanes, V. González-Ruiz, G. Camps-Valls, A. Plaza, and J. Serra-Sagristà and. On the impact of lossy compression on hyperspectral image classification and unmixing. *IEEE Geosc. Rem. Sens. Lett.*, 8(2):253–257, Mar 2011.