

[Home](#) > [Image and Graphics](#) > Conference paper

Unsupervised Representation Learning with Deep Convolutional Neural Network for Remote Sensing Images

| Conference paper | First Online: 29 December 2017

| pp 97–108 | [Cite this conference paper](#)




[Image and Graphics](#)

(ICIG 2017)

[Yang Yu](#), [Zhiqiang Gong](#), [Ping Zhong](#)  & [Jiaxin Shan](#)

 Part of the book series: [Lecture Notes in Computer Science](#) ((LNIP, volume 10667))

 Included in the following conference series:
[International Conference on Image and Graphics](#)

 7654 Accesses  77 [Citations](#)

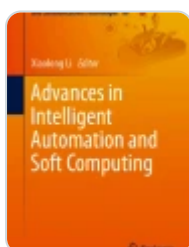
Abstract

With the rapid growth in quantity and quality of remote sensing images, extracting the useful information in them effectively and efficiently becomes feasible but also challenging. Convolutional neural network (CNN) is a suitable method to deal with such

challenge since it can effectively represent and extract the information. However, the CNN can release their potentials only when enough labelled data provided for the learning procedure. This is a very time-consuming task and even infeasible for the applications with non-cooperative objects or scenes. Unsupervised CNN learning methods, which relieve the need for the labels in the training data, is a feasible solution for the problem. In this work, we investigate a real-world motivated sparsity based unsupervised deep CNN learning method. At first, the method formulates a balanced data driven population and lifetime sparsity prior and thus construct the unsupervised learning method through a layerwise mean. Then we further perform the method on the deep model with multiple CNN layers. Finally, the method is used for the remote sensing image representation and scenes classification. The experimental results over the public UC-Merced Land-use dataset demonstrate that the developed algorithm obtained satisfactory results compared with the recent methods.

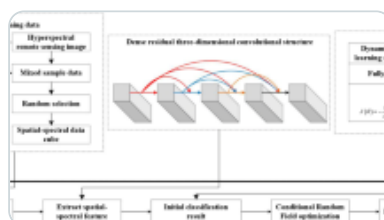
You have full access to this open access chapter, [Download conference paper PDF](#) 

Similar content being viewed by others



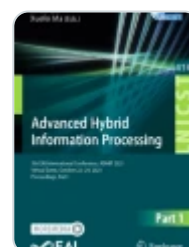
Remote Sensing Scene Classification by Multi-granularity Fused CNN

Chapter | © 2022



Hyperspectral remote sensing image classification based on dense residual three-...

Article | 10 September 2020



Deep Residual Network with Transfer Learning for High Spatial Resolution Remote...

Chapter | © 2022

Keywords

[Unsupervised representation learning](#)

[Convolutional neural network](#)

1 Introduction

In the last decades, remote sensing imaging has become increasingly important in environmental monitoring, military reconnaissance, precision farming and other domains. Both the quantity and quality of remote sensing images are growing rapidly. How to mine the useful information in such huge image repositories has been a challenging task. Traditional machine learning methods including the feature representation and classification algorithms cannot tackle this challenge well for their limited representation ability.

Recently, the emergence of deep learning combined with hierarchical feature representation has brought changes for the analysis of huge remote sensing images. Convolutional neural networks (CNN) [1], as one of the famous deep learning architectures, demonstrated impressive performances in the literature of computer vision. CNN can not only capture and represent more complicated and abstract images by powerful fitting capacity coming from deep structures but also make full use of a great number of training samples to achieve better results. The representation ability of CNN derives main from its deep model structure, and also a large number of model parameters. Estimating the model parameters usually need huge labelled data in the supervised learning framework. Preparing the numerous labelled data is a very time-consuming task. Moreover, in the applications for non-cooperative objects and scenes, it is impossible to get the labels for the observed data. Semi-supervised and unsupervised learning methods could deal with this problem. In this work, we mainly investigate the unsupervised learning method for the CNN model and aim to relieve the heavy consumption of the numerous labelled training data and meanwhile to maintain its performance on representation ability.

The goal of unsupervised deep learning is utilizing the observed data only (no need for the corresponding semantic labels) to learn a rich representation that exposes relevant semantic features as easily decodable factors [2]. The deep learning community has completed primary researches on this topic. There are three kinds of methods to implement the unsupervised deep learning. The first is to design specific models which

naturally allow the unsupervised learning. The conventional models are autoencoders [3] and recently their variants, such as denoising autoencoders (DAE [3]) and ladder network [4]. The restricted Boltzman machines (RBMs) can also be trained in an unsupervised way [5]. Besides, stacking multiple autoencoders layers or RBMs produce the well-known deep belief network (DBN) [6], which naturally has the ability from autoencoders and RBMs on allowing unsupervised learning.

The second kind of unsupervised deep learning methods is implemented through a particular model structure and learning strategy. The generative adversarial networks (GANs) is the recent popular method for unsupervised learning [7]. The GAN method trains a generator and a discriminator by the learning strategy as rule of minimax game. Following the research direction of GANs, deep convolutional generative adversarial networks (DCGAN [8]) also obtained a good performance on the generation and feature learning of images, and Wasserstein GAN [9] improved the defects of GAN and achieved a more robust framework. From the application view, the structure of DCGAN has been successfully introduced into the remote sensing literature and obtained impressive remote sensing image representation and scene classification results [2]. But the difficulty in training procedure is the GAN' defect that cannot be ignored.

The third kind of unsupervised deep learning method is substantially embedded into the learning process through formulating specific loss objectives, which usually follow the real-world rules and can be computed with no needs of semantic labels in the training samples. The principles of sparsity and diversity can be used to construct loss function for deep unsupervised learning. Enforcing population and lifetime sparsity (EPLS) [10] is such a method with simple ideas but remarkable performance. The EPLS enforces the output of each layer conform to population sparsity and lifetime sparsity. It is an entirely unsupervised method that can be trained with unlabeled image patches layer by layer and does not need a fine-tuning process. In contrast to other deep learning methods, especially GAN, EPLS is very easy to train and also very robust to convergence.

Considering the easy implementation and also the superior representation ability obtained through only the unsupervised learning, this work investigates the EPLS based unsupervised representation learning with Deep CNN for remote sensing images. Especially this work investigates a balanced data driven sparsity (BDDS) [11] EPLS algorithm in the deep CNN. In this method, the CNN is trained through the extended EPLS

with a balanced data driven sparsity, and the multiple convolutional layers are stacked to form the deep network obtained through entirely unsupervised learning method.

2 EPLS

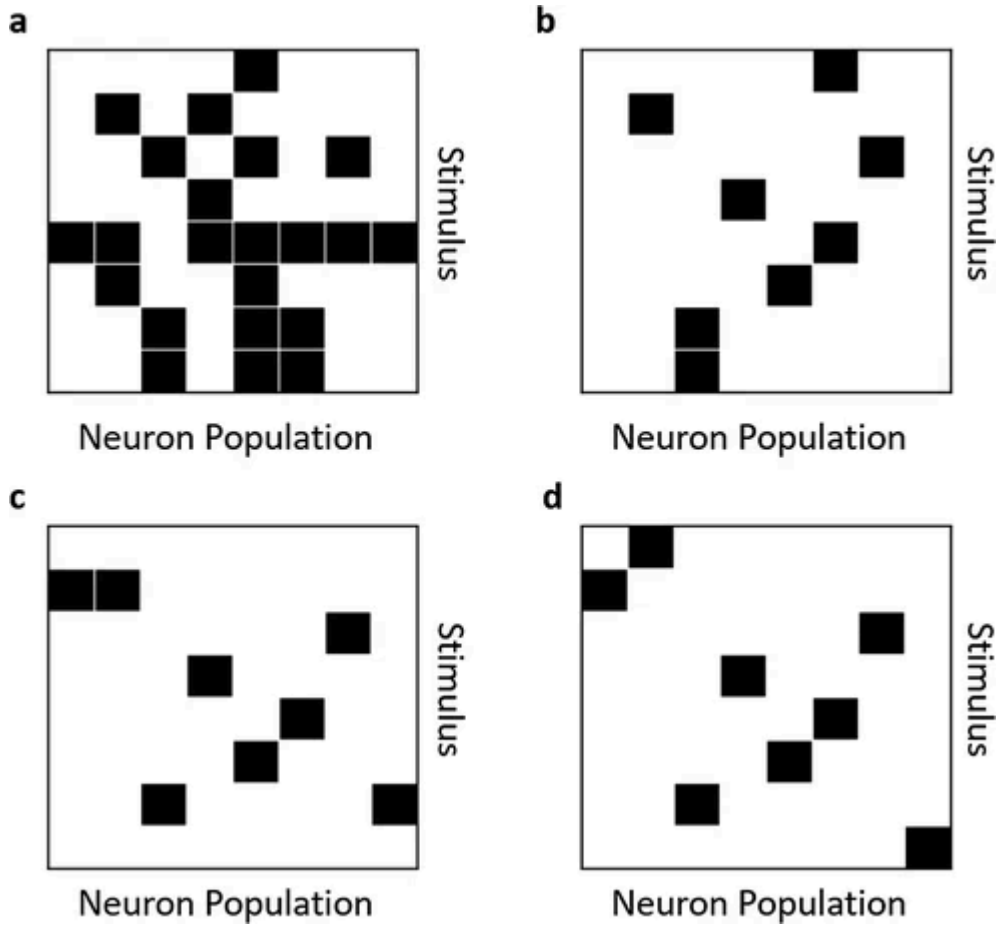
EPLS builds a sparse objective through enforcing lifetime sparsity and population sparsity to filters output and optimizes the parameters by minimizing the error between the filters output and the sparse target. The reasons why enforcing lifetime and population sparsity will be explained in Subsect. [2.1](#). Subsection [2.2](#) will present the algorithm details.

2.1 Population and Lifetime Sparsity

Sparsity is one of the desirable properties of a good network's output representation. Its primary purpose is to reduce network's redundancy and improve its efficiency and diversity. Sparsity can be described in terms of population sparsity and lifetime sparsity [[12](#)]. Population sparsity means that the fraction of neurons activated by a particular stimulus should be relatively small. The sparsity assumption can reduce the number of redundant neurons and enhance the networks' ability of description and efficiency. Lifetime sparsity expresses that a neuron is constrained to respond to a small number of stimuli and each neuron must have a response to some stimuli. So lifetime sparsity plays a significant role in preventing bad solutions such as dead outputs.

The degree of population and lifetime sparsity can be adjusted for specific task and requirement for better performances. In our task at hand, the sparsity degrees are set as follows. On the one hand, strong population sparsity is demanded that for each training sample (stimulus) in a mini-batch only one neuron must be activated as active or inactive (no intermediate values are allowed between 1 and 0). On the other hand, we enforce a strict lifetime sparsity since each neuron must be activated only one time by a certain training simple in one mini-batch. So the ideal outputs of a mini-batch in training EPLS are one-hot matrices, as shown in Fig. [1](#)(d). Besides, Fig. [1](#) also shows other three situations of output according to different kinds of sparsity for comparisons.

Fig. 1.



Understanding strong population and lifetime sparsity. (a) Outputs that disobey the rule of sparsity. (b) Strong population sparsity. (c) Strong lifetime sparsity. (d) Outputs that conform both population and lifetime sparsity.

2.2 EPLS Algorithm

EPLS is usually used for the unsupervised learning in a layer-wise way. For the layer l in CNN, EPLS builds a one-hot target matrix T^l (as shown in Fig. 1(d)) to enforce both lifetime and population sparsity of the output H^l from a mini-batch. The parameters of this layer will be optimized by minimizing the L_2 norm of the difference between H^l and T^l ,

$$H^l = \sigma(D^l W^l + b^l)$$

(1)

$$\{W^l, b^l\} = \underset{\{W^l, b^l\}}{\operatorname{argmin}} \left\| H^l - T^l \right\|_2^2$$

(2)

where D^l is the mini-batch matrix of layer l , W^l and b^l are weights and bias of the layer, $\sigma(\cdot)$ is the pointwise nonlinearity, $H^l, T^l \in \mathbb{R}^{m \times r}$, m is the number of patches in a mini-batch, r is the number of neurons of layer l , N is the total number of training patches. For the optimization of parameters in layer l , the mini-batch stochastic gradient descent algorithm will be performed.

Algorithm 1 presents the steps to implement EPLS. The primary objective of EPLS is to build the target matrix T^l , which will be assigned to be a one-hot matrix depending on the maximum values in H^l . EPLS will process rows of H^l one by one iteratively. In each row, the algorithm selects the neuron k of the n th row that has the maximal activation value (h_j minus an inhibitor a_j) to be set as the only one “hot code”, and thus ensure population sparsity [10]. Once the neuron k has been selected, the corresponding inhibitor a_k will be increased by r/N . The inhibitor which can measure the number of each neuron has been selected, and the phenomenon of dead outputs or one neuron form being selected too many times is prevented, and thus ensure lifetime sparsity. More details on the EPLS algorithm can be found in [10].

Algorithm 1 EPLS [11]

Input: H^l, a, N
Output: T^l, a

 1: $T^l = 0$

 2: $H^l = (H^l - \min(H^l)) / (\max(H^l) - \min(H^l))$

 3: **for** $n = 1 \rightarrow m$ **do**

 4: $h_j = H^l_{n,j} \quad \forall j \in \{1, 2, \dots, r\}$

 5: $k = \operatorname{argmax}_j (h_j - a_j)$

 6: $T^l_{n,k} = 1$

 7: $a_k = a_k + (r/N)$

 8: **end for**

 9: Remap T^l to active/inactive values

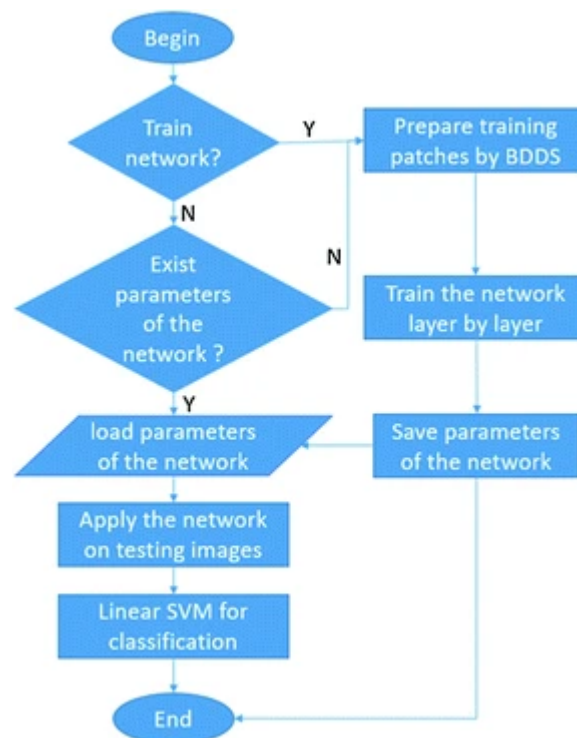
Though EPLS tactfully enforce lifetime sparsity through the inhibitors, a defect will exist when the inhibitor of a certain neuron k is too big, wrong neurons will be activated by patches corresponding to the neuron k . It will influence the performance of EPLS, and we call it “neuron saturation” that will be discussed in the next section.

3 Unsupervised Learning of Deep CNN with BDDS Based EPLS

3.1 Flowchart of the Method

The method contains both training and testing processes, as shown in Fig. 2. The testing process is similar with the one in other deep learning algorithms, which applies the deep networks on testing samples to get discriminative features. Because of the feature of EPLS' loss function (2) training process is different, parameters are independent of each layer. It is impossible to train all the parameters at one time, an algorithm to update the parameters layer by layer is needed and it will be discussed in Subsect. 3.3. It is also noteworthy that we perform a BDDS operation before training the network, which makes up the shortcomings of EPLS and will be discussed in Subsect. 3.2.

Fig. 2.



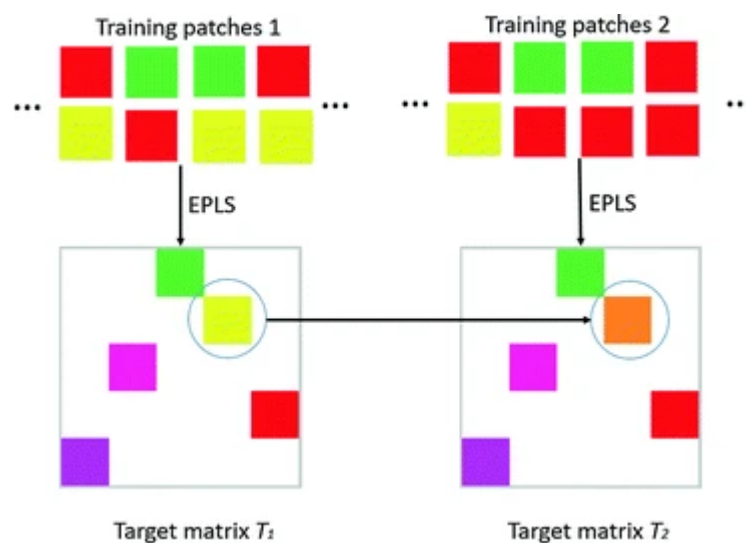
Flow chart of the method.

3.2 Balanced Data Driven Sparsity (BDDS) Based EPLS

“Neuron Saturation” Phenomenon.

EPLS achieves lifetime sparsity by enforcing a neuron being selected only once in a mini-batch through inhibitors. Such operation must face two drawbacks. On the one hand, training image patches randomly abstracted from remote sensing images are unbalanced, i.e., the numbers of patches from different potential classes could be very different, some are huge while some are small. Just part of such kinds of patches with a huge number will greatly improve value of the inhibitor corresponding to a neuron, thus the neuron will be saturated. Other neurons that respond to the different patches will be enforced to respond to the remaining part of that kind of training patches. We call such phenomenon “neuron saturation” which will reduce the diversity of filters. Figure 3 shows how the unbalanced data influences the target matrix. Training patch set 1 is a balanced training set corresponding to a “healthy” target matrix T_1^l . If there are too many red patches and too little yellow patches just like training patch set 2, a “sick” target matrix T_2^l will appear. It is because additional red patches may enforce the yellow neuron make the weights that were learned for yellow patches turn to respond to the red patches.

Fig. 3.



The effect of the unbalanced training data [11]. (Color figure online)

On the other hand, even though the whole training patches is balanced, inputting too many similar patches to the network at the same time will also lead to a local “neuron saturation”. Patches enforced to activate the wrong neuron will be wasted, and unrelated neuron will be “contaminated”. It will reduce the efficiency of the training process and decrease the network’s performance. A balanced data driven sparsity (BDDS) based EPLS

will be introduced to solve the problem from local and global “neuron saturation” phenomenon in next subsection.

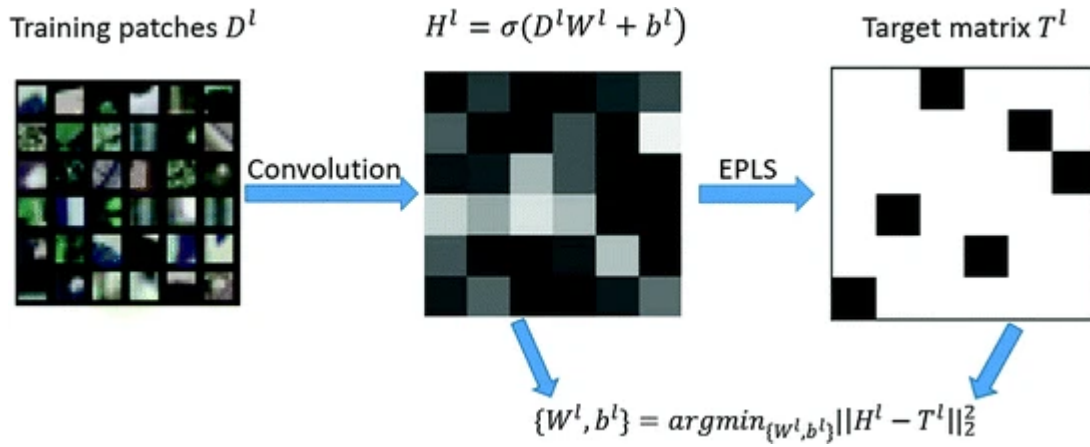
BDDS Based EPLS.

To address the issue of local and global “neuron saturation” phenomenon and achieve a more natural sparsity, we construct the balanced training samples for EPLS. There are four steps to implement the proposed method. (1) **Patches Extraction**. The patches are randomly extracted from remote sensing image datasets. (2) **Clustering**. All patches are clustered into n classes (Sect. 4 will set an experiment to explore the suitable value of n). In this work, we perform the clustering over the color LBP features of training patches. We perform LBP [13] to all three channels of training patches and concatenate all these features to LBP texture features, thus color LBP features is obtained. The classes without enough patches will be supplemented at random. (3) **Arrangement**. This step extracts one sample per category to constitute a balanced mini-batch. As a result, every mini-batch contains samples covering all classes, and the numbers of samples from different classes are same. (4) **EPLS**. This step uses the balanced mini-batches to train EPLS and the problems from local and global “neuron saturation” phenomenon could be solved.

3.3 Applying BDDS Based EPLS on Multiple Layers

BDDS-EPLS is usually performed layerwise. Moreover, a single layer trained by BDDS-EPLS could obtain impressive results. In this work, we further investigate to apply the BDDS-EPLS method to multiple layers for more powerful representation ability. Because global backpropagation cannot be used on EPLS' layerwise loss function, perform EPLS training on deep CNN model is different from the training of other deep networks. Since EPLS needs to be trained layer by layer, a greedy layerwise unsupervised pretraining [14] will be performed to implement the EPLS training over the deep model. It is based on the idea that a layerwise unsupervised criterion can be applied to pretrain the network's parameters, allowing the use of large amounts of unlabeled data. Figure 4 shows the layerwise training process of a BDDS-EPLS network. All parameters in every layer need to be optimized by greedy layerwise unsupervised training.

Fig. 4.



The layerwise training process of a BDDS-EPLS network

Algorithm 2 shows the detail steps to train BDDS-EPLS. For each layer, parameters are updated independently. The updating process as follows: firstly, the method performs BDDS algorithm for balanced training patches. Secondly, the feature matrix H^l is obtained through (1) and the target matrix through EPLS algorithm. Finally, mini-batch stochastic gradient descent algorithm is performed on minimizing (2) to optimize parameters in layer l . Repeat the training process until the stop condition is reached.

Algorithm 2 Greedy layerwise unsupervised training of BDDS-EPLS

```

1: for  $l = 1 \rightarrow L$  do
2:    $D^l \leftarrow \text{BDDS}(l)$ 
3:   do
4:      $H^l \leftarrow \text{EPLS}(D^l)$ 
5:      $\{W^l, b^l\} \leftarrow$  updated by minimising  $\|H^l - T^l\|_2^2$ 
6:   while achieving stop condition
7: end for

```

4 Experiments

4.1 Experimental Setup

In this section, experiments are set to show the effects of the single layer and multiple layers BDDS-EPLS in different scenarios of image classification on Ucmersed dataset [15]. We randomly select 80 images per class for training and leave the remaining 20 ones for

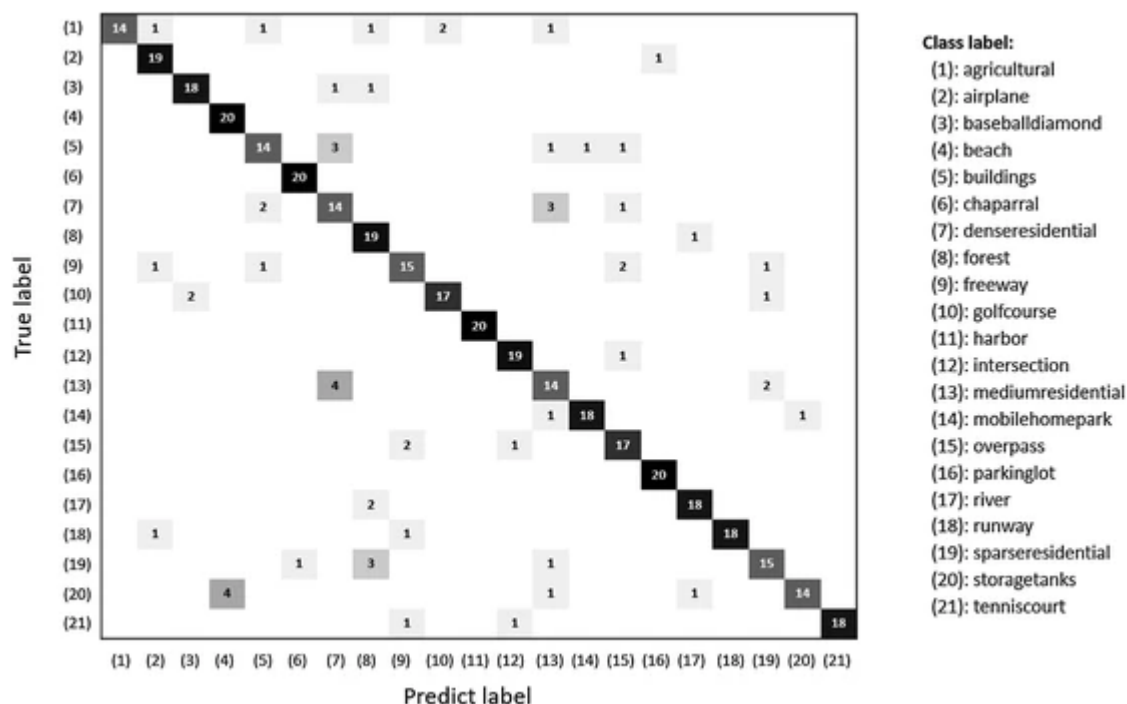
testing. Both the number of neurons and the size of mini-batch are set to 500 to all experiments and the number of training samples will be set to 105000 in every layer. The receptive field is set to 7×7 with stride 1 pixel. And we applied a non-overlapping max-pooling of 2×2 pixels at each representation layer, except for the last layer, which divides the output feature map into 2×2 pixels and feeds into a linear SVM.

4.2 General Performance

Representation and Classification Performance.

The two-layer BDDS-EPLS network obtains a classification accuracy on 85.95% that significantly improves the performance of two-layer EPLS network reported in [10]. Figure 5 shows the confusion matrices of the representation features of the two-layer BDDS-EPLS learned. Errors concentrate on close classes such as dense-residential, building and mobile-home-park, and the other classes are classified well even 100% accuracy appears in some classes. It shows that BDDS-EPLS has the powerful capacity of unsupervised feature representation only with unlabeled image patches as training samples.

Fig. 5.

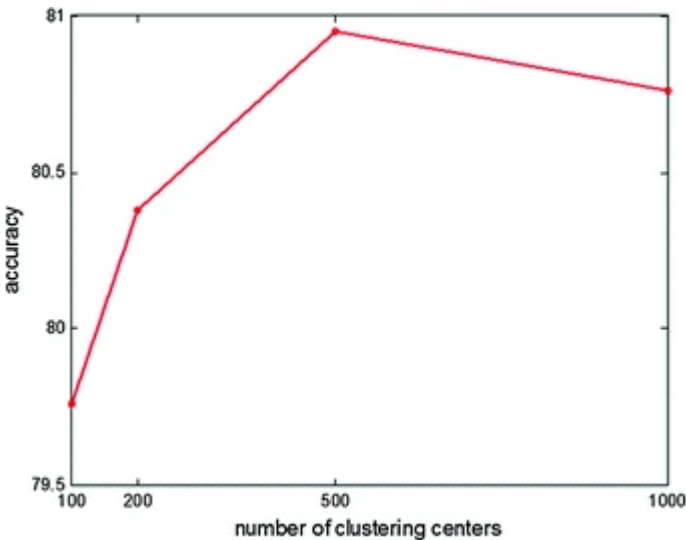


Confusion matrices of the two-layer BDDS-EPLS

Effects of the Different Number of Clustering Centers.

The experiment is designed for testing effects of the different number of clustering centers m in BDDS. We follow the experimental pipeline setup of [10]. The only difference is the way selecting training patches in which we use color LBP features of training patches for BDDS. m is set to 100, 200, 500, 1000 respectively and Fig. 6 shows that the highest classification 80.95% appears on m equal to 500. It maybe means that when the number of clustering centers equals to the number of neurons in the network, the best suppression was yielded for “neuron saturation”.

Fig. 6.



Effects of the different number of clustering centers

Effects of the Different Number of Layers in BDDS–EPLS.

The experiment is designed for testing effects of the different number of layers in BDDS and looks forward to a better performance by deeper networks. Table 1 shows the classification performance of both the EPLS and the BDDS–EPLS, for the different number of layers as configurations. As shown in the table, the BDDS–EPLS perform better than EPLS on all the single-layer, two-layer and three-layer networks, which proves the ability for better feature representation. It is noteworthy that the two-layer BDDS–EPLS network obtains the best result 85.95% that significantly improves the performance of two-layer EPLS network 83.81 by 2.14%. And only 2100 images in Ucmcerced dataset maybe the reason for the accuracy decline in the three-layer EPLS, because comparing to the huge

number of parameters in a three-layer EPLS the number of training samples (2100) is too small to fit the network.

Table 1. Accuracies of EPLS and BDDS–EPLS with the different number of layers.

4.3 Comparing with Other State of the Art Unsupervised Algorithms

Table 2 shows the classification accuracies of several state of the art algorithms on Ucmerced dataset. Our two-layer based BDDS–EPLS gets the best performance 85.95% even higher than the six-layer DCGANs and MARTA GANs (without data augmentation). EPLS that combines CNN with strong population sparsity and lifetime sparsity has the great ability of unsupervised representation learning. The BDDS method addresses the “neuron saturation” phenomenon in EPLS and release EPLS’s ability, thus achieve the best performance. It is also noteworthy that EPLS’ classification accuracy in [10] achieves 84.53% with 1000 neurons per layer, while our method’s classification accuracy reaches 85.95% with only 500 neurons per layer. It shows the power of our BDDS–EPLS method.

Table 2. Classification accuracies of some state of the art algorithms. Best result in bold.

5 Conclusions and Future Work

In this work, we apply a deep BDDS–EPLS network on Ucmerced dataset, and significantly improve the classification accuracy. In the future, we will try to increase the depth of the network and generalize BDDS–EPLS for hyperspectral imagery.

References

1. Chen, K., Ding, G., Han, J.: Attribute-based supervised deep learning model for action recognition. *Front. Comput. Sci.* **11**, 219–229 (2017)

2. Bengio, Y.: Learning Deep Architectures for AI. Foundations and Trends in Machine Learning. Now Publishers Inc, Breda (2009)

[MATH](#) [Google Scholar](#)

3. Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.A.: Extracting and composing robust features with denoising autoencoders. In: International Conference on Machine Learning, pp. 1096–1103 (2008)

[Google Scholar](#)

4. Valpola, H.: From neural PCA to deep unsupervised learning. arXiv preprint (2014)

[Google Scholar](#)

5. Smolensky, P.: Information processing in dynamical systems: foundations of harmony theory. In: David, E.R., James, L.M. (eds.) Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol. 1, pp. 194–281 (1986). Chapter 6

[Google Scholar](#)

6. Hinton, G.E., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief nets. Neural Comput. **18**, 1527–1554 (2006)

[Article](#) [MathSciNet](#) [MATH](#) [Google Scholar](#)

7. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S.: Generative adversarial nets. In: Conference on Neural Information Processing Systems, vol. 3, pp. 2672–2680 (2014)

[Google Scholar](#)

8. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. In: Computer Science (2015)

[Google Scholar](#)

9. Martin, A., Soumith, C., Léon, B.: Wasserstein GAN. arXiv preprint (2017)

[Google Scholar](#)

10. Romero, A., Gatta, C., Camps-Valls, G.: Unsupervised deep feature extraction for remote sensing image classification. *IEEE Trans. Geosci. Remote Sens.* **54**(3), 1–14 (2015)

[Google Scholar](#)

11. Yang, Y., Zhiqiang, G., Ping, Z.: Balanced data driven sparsity for unsupervised deep feature learning in remote sensing images classification. In: *IEEE International Geoscience and Remote Sensing Symposium* (2017)

[Google Scholar](#)

12. Willmore, B., Tolhurst, D.J.: Characterizing the sparseness of neural codes. *Network* **12**(12), 255–270 (2001)

[Article](#) [Google Scholar](#)

13. Ojala, T., Pietikäinen, M., Mäenpää, T.: Gray scale and rotation invariant texture classification with local binary patterns. In: *European Conference on Computer Vision*, pp. 404–420 (2000)

[Google Scholar](#)

14. Bengio, Y., Lamblin, P., Popovici, D., Larochelle, H.: Greedy layerwise training of deep networks. In: *Conference on Neural Information Processing Systems*, pp. 153–160 (2006)

15. Yang, Y., Newsam, S.: Bag-of-visual-words and spatial extensions for land-use classification. In: ACM SIGSPATIAL GIS, pp. 270–279 (2010)

[Google Scholar](#)

16. Cheriyyadat, A.: Unsupervised feature learning for aerial scene classification. IEEE Trans. Geosci. Remote Sens. 52(1), 439–451 (2014)

[Article](#) [Google Scholar](#)

17. Yang, Y., Newsam, S.: Spatial pyramid co-occurrence for image classification. In: Proceeding of IEEE International Conference on Computer Vision, pp. 1465–1472, November 2011

[Google Scholar](#)

Acknowledgement

This research was conducted with the support of the Natural Science Foundation of China under Grant 61671456 and 61271439, A Foundation for the Author of National Excellent Doctoral Dissertation of P. R. China (FANEDD) under Grant 201243, Program for New Century Excellent Talents in University under Grant NECT-13-0164.

Author information

Authors and Affiliations

ATR Laboratory, School of Electronic Science and Engineering, The National University of Defense Technology, Changsha, 410073, Hunan, China

Yang Yu, Zhiqiang Gong, Ping Zhong & Jiabin Shan

Corresponding author

Correspondence to [Ping Zhong](#).

Editor information

Editors and Affiliations

Beijing Jiaotong University, Beijing, China

Yao Zhao

Dalian University of Technology, Dalian, China

Xiangwei Kong

UNSW, Sydney, New South Wales, Australia

David Taubman

Rights and permissions

[Reprints and permissions](#)

Copyright information

© 2017 Springer International Publishing AG

About this paper

Cite this paper

Yu, Y., Gong, Z., Zhong, P., Shan, J. (2017). Unsupervised Representation Learning with Deep Convolutional Neural Network for Remote Sensing Images. In: Zhao, Y., Kong, X., Taubman, D. (eds) Image and Graphics. ICIG 2017. Lecture Notes in Computer Science(), vol 10667. Springer, Cham. https://doi.org/10.1007/978-3-319-71589-6_9

[.RIS](#) [.ENW](#) [.BIB](#)

DOI

https://doi.org/10.1007/978-3-319-71589-6_9

Published

29 December 2017

Publisher Name

Springer, Cham

Print ISBN

978-3-319-71588-9

Online ISBN

978-3-319-71589-6

eBook Packages

[Computer Science](#)[Computer Science \(R0\)](#)

Share this paper

Anyone you share the following link with will be able to read this content:

[Get shareable link](#)

Provided by the Springer Nature SharedIt content-sharing initiative

Publish with us

[Policies and ethics](#) 

Societies and partnerships



[The International Association for Pattern Recognition](#) 