Neurocomputing 219 (2017) 88–98

Contents lists available at ScienceDirect

Neurocomputing

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

Convolutional neural networks for hyperspectral image classification

Shiqi Yu [a], Sen Jia [a,*], Chunyan Xu [b]

[a] College of Computer Science and Software Engineering, Shenzhen University, China
[b] School of Computer Science and Technology, Nanjing University of Science and Technology, China

ARTICLE INFO

Article history:
Received 8 July 2016
Received in revised form
30 August 2016
Accepted 8 September 2016
Communicated by Jiwen Lu
Available online 13 September 2016

Keywords:
Hyperspectral image classification
Convolutional neural networks
Deep learning

ABSTRACT

As a powerful visual model, convolutional neural networks (CNNs) have demonstrated remarkable performance in various visual recognition problems, and attracted considerable attention in recent years. However, due to the highly correlated bands and insufficient training samples of hyperspectral image data, it still remains a challenging problem to effectively apply the CNN models on hyperspectral images. In this paper, an efficient CNN architecture has been proposed to boost its discriminative capability for hyperspectral image classification, in which the original data is used as the input and the final CNN outputs are the predicted class-related results. The proposed CNN infrastructure has several distinct advantages. Firstly, different from traditional classification methods those need hand-crafted features, the CNN model used here is designed to deal with the problem of hyperspectral image analysis in an end-to-end way. Secondly, the parameters of the CNN model are optimized from a small training set, while the over-fitting problem of the neural network has been alleviated to some extent. Finally, in order to better deal with the hyperspectral image information, 1×1 convolutional layers have been adopted, and an average pooling layer and larger dropout rates have also been employed in the whole CNN procedure. The experiments on three benchmark data sets have demonstrated that the proposed CNN architecture considerably outperforms other state-of-the-art methods.

© 2016 Elsevier B.V. All rights reserved.

1. Introduction

With the rapid development of remote sensors, the acquisition and collection of hyperspectral data has become much easier and more affordable, making hyperspectral image analysis to be one of the most promising techniques in many practical applications, including precision agriculture, environmental monitoring, military surveillance, etc. [1,2]. Hyperspectral image (HSI) data often contains hundreds of spectral bands over the same spatial area, which has provided valuable information to identify the various materials [3,4]. HSI classification is similar with image labeling in computer vision field [5]. One difference between them is the number of data bands. There are normally about 200 bands for a HSI, but only 3 or 4 ones in image labeling. Another difference is the number of labeled sample. For image labeling, it is relatively easy to label samples. In HSI classification, due to the difficulty and expense of manually labeling, the limited availability of labeled training samples is the main obstacle of hyperspectral image classification. Hence the small sample set (3S) problem has attracted increasing attention in recent years [6,7].

One reasonable way to tackle the 3S problem is dimensionality reduction [8,9], which could largely reduce the impact of Hughes phenomenon, i.e., a large amount of labeled samples is needed for the high-dimensional data to obtain reliable results [10]. Dimensionality reduction can be accomplished by transforming the original hyperspectral data into a low-dimensional space (referred as feature extraction) [11,12], such as principal component analysis (PCA) [13,14], independent component analysis (ICA) [15], manifold learning [16] or directly picking out the most representative bands from the hyperspectral data (referred as band selection) [17,18], such as the ranking-based methods [19,20] and clustering-based methods [8,21,22]. But some important information may be lost during the dimensionality reduction process. More severely, the features obtained through dimensionality reduction in the spectral domain cannot fully characterize the properties of the materials, hence more discriminative features should be extracted.

Fortunately, spatial information, which reflects the fact that the adjacent pixels in the spatial domain belong to the same class with a high possibility, is a valuable complement to the spectral signatures, and has been extensively studied for hyperspectral image classification [23]. Specifically, the mathematical morphology method applies the opening and closing morphological transforms on several principal components to obtain features containing spatial structure information [24]. Other spatial filters [25,26] are

* Corresponding author.
E-mail addresses: shiqi.yu@szu.edu.cn (S. Yu), senjia@szu.edu.cn (S. Jia), xuchunyan01@gmail.com (C. Xu).

http://dx.doi.org/10.1016/j.neucom.2016.09.010
0925-2312/© 2016 Elsevier B.V. All rights reserved.

also used to exploit the spatial regularity of materials. Moreover, the contextual information can be used to refine the classification results through a regularization process in the postprocessing stage [27]. However, a large number of training samples are generally required to adequately characterize the large variability of the objects, which is difficult to meet in practice. Alternatively, in order to extract the joint spatial-spectral features, three-dimensional wavelet-based methods, especially the Gabor filters, have been proposed to simultaneously fuse the spectral and spatial information, which has shown competitive classification performance for the 3S problem [28–30]. Recent advances have revealed that Gabor filters with different predefined orientations and scales are a kind of convolutional filters, whereas the popular convolutional neural networks (CNNs) can learn convolutional filters automatically [31]. These encouraging results have motivated us to apply the CNN model for hyperspectral image classification.

A convolutional neural network is composed of alternatively stacked convolutional layers and spatial pooling layers. The convolutional layer is to extract feature maps by linear convolutional filters followed by nonlinear activation functions (e.g., rectifier, sigmoid, tanh, etc.). Spatial pooling is to group the local features together from spatially adjacent pixels, which is typically done to improve the robustness to slight deformations of objects. CNNs have been long studied and applied in the field of computer vision. More than a decade ago, LeCun et al. [31] trained multilayer neural networks with the back-propagation algorithm and the gradient learning technique, and then demonstrated its effectiveness on the handwritten digit recognition task. The deep CNNs have exhibited good generalization power in image-related applications. Recently, Krizhevsky et al. [32] achieved a breakthrough, outperforming the existing handcrafted features on ILSVRC 2012 which contains 1000 object classes. Since 2012, CNNs have draw a resurgence of attention in various visual recognition tasks such as image classification [32,33], semantic segmentation [34,35], object recognition [36], video analysis [37], etc. Recently the networks are going deeper, such as GoogLeNet [33] which won 2014 ILSVRC classification challenge [38] by employing 22 layers. In [39] the number of layers of the proposed residual nets reaches to 152 and achieves better performance.

There are some deep learning related works on HSI classification in the literature. Such as in [40], deep stacked autoencoders are employed to extract features. The autoencoder is a kind of unsupervised method. The proposed method in [40] combines principle component analysis (PCA), autoencoders and logistic regression, and it is not an end-to-end deep method. An end-to-end deep CNN method is proposed in [41]. The method takes the raw data as the input and outputs the predicted class labels. The number of training samples of each class is 200, and it is a relative large number.

We propose a novel CNN structure for hyperspectral image analysis, where a pixel and its neighbors in a hyperspectral image are taken as inputs of the CNN, and the final CNN output is the predicted class labels. Our designed CNN structure can be illustrated in Fig. 1. The major contributions of this work can be summarized as follows:

- Different from common visual information (e.g., RGB images), hyperspectral images can collect and process visual signals across different electromagnetic spectra. Most traditional HSI classification methods employ hand-crafted features. We design a novel CNN structure to deal with the hyperspectral image analysis problem in an end-to-end way, and the network can learn features automatically.
- Under the limitation of small training samples, we employ some network strategies (e.g., data augmentation, lager dropout rates, etc.) to alleviate the over-fitting problem in the process of learning network parameters.
- For better coping with the hyperspectral image information, we adopt the 1×1 convolutional layers to analyze the hyperspectral information, and use an average pooling layer in the whole CNNs.
- Compared with several popular features, i.e., raw spectral features, morphological features, and 3D Gabor features with traditional classifiers, the state-of-the-art classification results on three popular data sets verify the effectiveness of the proposed CNN framework. And the corresponding CNN model will be released and serve as the benchmark for the problem of hyperspectral image analysis in the research community.

The remaining part of the paper is organized as follows. The proposed network structure and design principles are presented in Section 2. The data sets used in the experiments and the evaluation methods are introduced in Section 3. Section 4 is the experimental results and analysis. The last section, Section 5, concludes the paper.

2. The CNN structure design

Since the training samples are limited, the main principle in our designed CNN structure is to alleviate the overfitting problem and gains a good generalization. We employed 1×1 convolutional kernels, improved dropout rates, discarded full connection layers, etc. The CNN structure and parameters designing are described in the following parts in detail.

2.1. Network structure

The network structure is illustrated in Fig. 1.[1] There are 3 convolutional layers in our network structure. The first two convolutional layers are followed by normalization layers and dropout layers. The input data is sent to the first convolutional layer, and the data size is $5 \times 5 \times N$ where N is the number of channels for hyperspectral images. In the convolutional layers, 1×1 sized kernels are employed as suggested in [42]. In the first convolutional layer, there are 128 filters. So the output of the first convolutional layer is $5 \times 5 \times 128$. After the convolution step, a 2×2 normalization is operated on each channel, and the next step is a dropout operation. After that the data is sent to the second convolutional layer, and also followed by a normalization layer and a dropout layer as the first convolutional layer. The output of the third convolutional layer is $5 \times 5 \times C$ where C is the number of classes. The global average pooling (GAP) is following the third convolutional layer. The input to GAP is the feature map with the size of $5 \times 5 \times C$. The GAP computes the average values on different channels, and there are C channels in this situation. So the final output of the network is a $1 \times C$ vector. If the ith element in the vector has the maximal value, then i is the predicted label for the input sample. The detailed design principles of the network are described in the following subsection.

2.2. Parameter learning for the network

Data augmentation: Because the training samples are limited, the learning model tends to overfitting. To reduce overfitting in the training stage, one of the most common methods is to transform the training samples to many different ones. The transform is named as data augmentation. In our experiments, each pixel and

[1] All the source codes and model files of the proposed CNN framework will be released to the community.

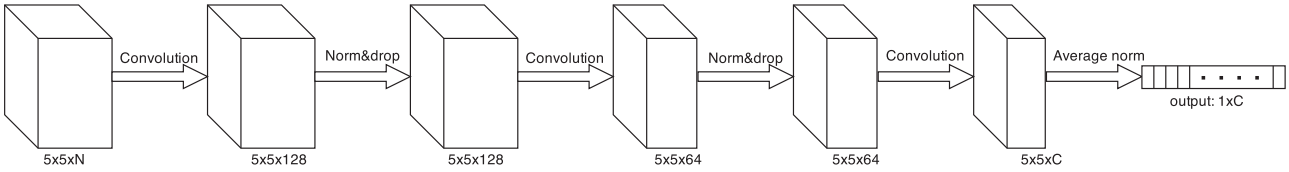


Fig. 1. An illustration of the proposed CNN network. The whole architecture is composed of convolutional layers, normalization layers, dropout layers and a average pooling layer. Specifically, the first 2 convolutional layers are followed by normalization layers and dropout layers, and the last convolutional layers are followed by a global average pooling layer. *N* is the number of input data channels and *C* is the number of classes.

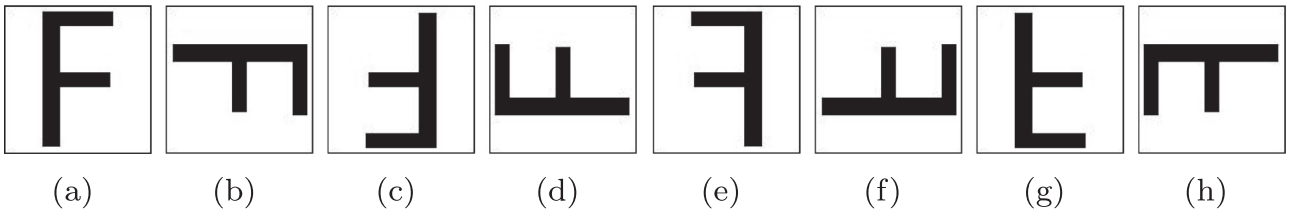


Fig. 2. An illustration of data augmentation. The 1st one is the original sample, and the 2nd, 3rd and 4th ones are rotated 90°, 180° and 270° from the original one, respectively. The last 4 samples (from e to h) row are flipped from the first 4 ones (from a to d).

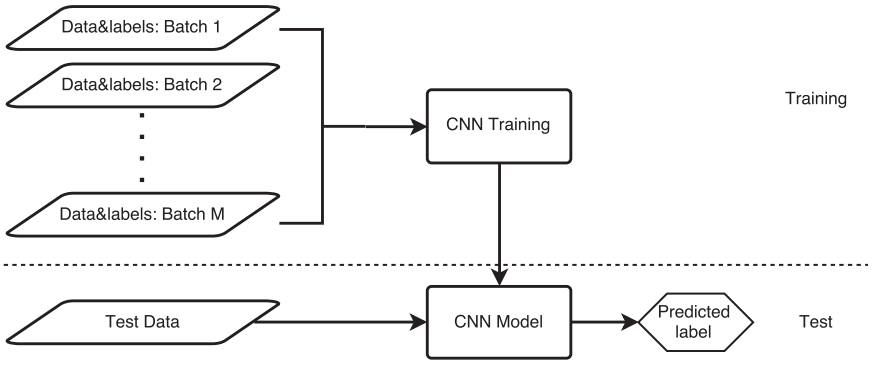


Fig. 3. Illustration of the parameter learning process and application of hyperspectral analysis.

its neighbors in a 5×5 block are taken as one sample. So the data size of each sample is $5\times 5\times N$. We rotated and flipped the training samples to enlarge the training set. For each training sample, there are 7 combinations as illustrated in Fig. 2. The sample is rotated with 90°, 180° and 270°, respectively, and then flip the 4 samples (the original one and its 3 variations). The training set is enlarged 8 times with these transformations.

Convolutional kernel: In general the size of convolutional filters in CNNs is $W\times W\times N$ where W can be 3, 5 or even greater. The filters shift horizontally and vertically on the image samples. In [42], a new structure named as Network in Network (NIN) is proposed. In NIN, Cascaded Cross Channel Parametric Pooling (CCCP) is employed, and the CCCP layer is equivalent to a convolution layer with 1×1 convolutional kernels. The 1×1 convolutional kernels can only extract features among different bands, and not in spatial domain. But there are two normalization layers and one global average pooling (GAP) layer in the proposed model. These two kinds of layers can extract features in spatial domain. We also found that convolutions on spatial domain tend to cause overfitting, and the 1×1 convolutional kernels can gain better generalization capability with better discrimination.

As suggested in [42], fully connected layers are not employed in our network structure. One global average pooling (GAP) layer is used to replace the traditional fully connected layers. There are much more parameters needed to be trained in fully connected layers than in convolutional layers. Using a GAP layer can save the parameters in fully connected layers. We also found that a network with CCCP+GAP has less global overfitting and faster convergence as mentioned in [42].

Dropout rate: Dropout can improve neural networks by reducing the overfitting problem [32,43,33], and it has been widely used in many deep learning applications. The main idea is to reduce the co-adaptation of hidden units. The operation of dropout is to set the output of each neuron to zero with a probability. Using the dropout strategy, the network is forced to learn more robust features and reduce the effect of noise. There are two dropout layers in our network structure. The drop rates are all set to 0.6 and are greater than the most commonly used value 0.5.

2.3. Application for hyperspectral image analysis

In the training process, firstly the training samples are divided into some batches randomly, and each batch contains equal number of samples. The batch size is 16 in our experiments. CNNs are trained using the stochastic gradient descent, and for each iteration only one batch is sent into the network for training. The training process will not stop until it reaches the maximal number of iterations.

In test process, the test sample is sent into the trained network, and the predicted label can be obtained by finding the maximal value in the output vector. The training and test processes are illustrated in Fig. 3.

3. Data sets and experimental design

To evaluate the proposed CNN model, three popular real-world data sets are used in the experiments. Since we mainly focus on the 3S problem, the designed experiments only take 3–15 training samples per class.

3.1. Data sets

Three real-world remote sensing hyperspectral data sets with different spatial resolution used for the experiments. They are Indian, Salinas and PaviaU data sets. The first two data sets were collected over wild areas, and the third one was over an urban area.

The first airborne data set is the commonly used Indian Pines data set acquired by the AVIRIS instrument over the agricultural area of Northwestern Indiana in 1992, which has spatial dimension of 145 × 145 and 224 spectral bands. The spatial resolution of the data is 20 m per pixel. After discarding four zero bands and 20 lower signal-to-noise ratio (SNR) bands affected by atmospheric absorption, 200 channels are preserved. The data set contains 10,366 labeled pixels and 16 ground-truth classes, most of which are different types of crops.

The second one was collected by AVIRIS sensor over Salinas Valley, California. The area covered comprises 512 lines by 217 samples and the spatial resolution of the data is 3.7 m per pixel. After 20 water absorption bands are removed, 204 out of the 224 bands are kept. The ground truth is composed of 54,129 pixels and 16 land-cover classes, including vegetables, bare soils, and vineyard fields.

The third one, PaviaU data set, is different from the previous two. It was collected in an urban area, and the previous two were collected in wild areas. PaviaU data set was acquired by the ROSIS sensor over Pavia University, Italy. The number of spectral bands is 103 after discarding the 13 noisiest bands. The hyperspectral image is of size 610 × 340. The geometric resolution is 1.3 m. There are 9 classes included in the ground truth, and 42,776 labeled samples.

All the pieces of information about the ground truth are listed in Tables 1–3 and Figs. 4–6.

3.2. Experimental design

The goal of the experiments is to evaluate the effectiveness of the proposed CNN architecture for hyperspectral image classification. For this reason, three different kinds of features are considered. The first one is the raw spectral features as a baseline (RAW), and the second one is the morphological features (MOR) [24], which are obtained by applying the opening and closing morphological transforms on the most significant principal components of the hyperspectral imagery (here two principal components are used, and four openings and four closings by a circular structuring element with a step size increment of 2 are applied on each component, resulting in a total of 18 morphological features for each hyperspectral data). Moreover, the 3D Gabor features [28] are also concerned. After the Gabor magnitude features have been extracted by convolving a set of predefined Gabor wavelets with the original hyperspectral data, a feature selection and fusion process has been developed to reduce the redundancy among Gabor features and make the fused feature more discriminative. With respect to the classification algorithms, *K*-nearest neighbors

Table 1
Land cover classes with number of samples for the Indian Pines data.

Class	Land cover type	No. of samples
*C*1	Stone-steel-towers	95
*C*2	Hay-windrowed	489
*C*3	Corn-min Till	834
*C*4	Soybean-no Till	968
*C*5	Alfalfa	54
*C*6	Soybean-clean Till	614
*C*7	Grass/Pasture	497
*C*8	Woods	1294
*C*9	Bldg-Grass-Tree-Drives	380
*C*10	Grass/Pasture-mowed	26
*C*11	Corn	234
*C*12	Oats	20
*C*13	Corn-no Till	1434
*C*14	Soybean-min Till	2468
*C*15	Grass/Trees	747
*C*16	Wheat	212
	Total	10,366

Table 2
Land cover classes with number of samples for the Salinas data.

Class	Land cover type	No. of samples
*C*1	Brocoli-green-weeds-1	2009
*C*2	Brocoli-green-weeds-2	3726
*C*3	Fallow	1976
*C*4	Fallow-rough-plow	1394
*C*5	Fallow-smooth	2678
*C*6	Stubble	3959
*C*7	Celery	3579
*C*8	Grapes-untrained	11,271
*C*9	Soil-vineyard-develop	6203
*C*10	Corn-senesced-green-weeds	3278
*C*11	Lettuce-romaine-4wk	1068
*C*12	Lettuce-romaine-5wk	1927
*C*13	Lettuce-romaine-6wk	916
*C*14	Lettuce-romaine-7wk	1070
*C*15	Vineyard-untrained	7268
*C*16	Vineyard-vertical-trellis	1807
	Total	54,129

Table 3
Land cover classes with number of samples for the PaviaU data.

Class	Land cover type	No. of samples
*C*1	Asphalt	6631
*C*2	Meadows	18,649
*C*3	Gravel	2099
*C*4	Trees	3064
*C*5	Painted metal sheets	1345
*C*6	Bare soil	5029
*C*7	Bitumen	1330
*C*8	Self-blocking bricks	3682
*C*9	Shadows	947
	Total	42,776

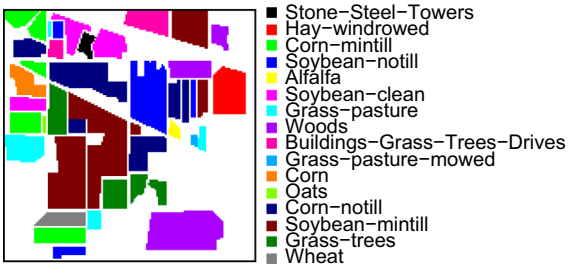


Fig. 4. Ground-truth map of the Indian Pines data set (sixteen land cover classes).

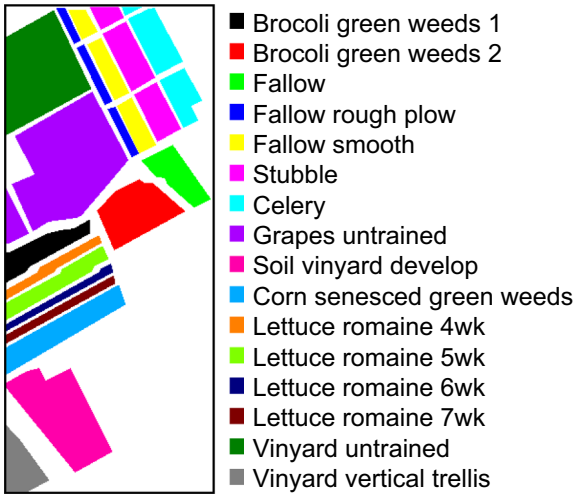


Fig. 5. Ground-truth map of the Salinas data set (sixteen land cover classes).

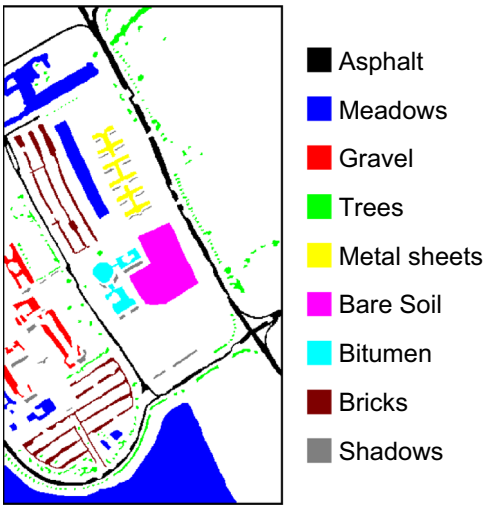


Fig. 6. Ground-truth map of the PaviaU data set (nine land cover classes).

(KNN) and support vector machine (SVM) have been applied to the three different kinds of features. In our experiments, the parameter of the number of neighbors in KNN is set to be 3, while radial basis function kernel and one-against-all scheme in SVM are used for multi-class classification. Besides, the parameters (kernel parameter and the regularization parameter) are estimated by cross validation. For the proposed CNN framework, we use Caffe software [44,45] to run the training and test procedures. All the source code and model files will been shared at http://github.com/ShiqiYu/caffe after the paper is accepted. They can be downloaded and used freely for research. In order to efficiently run our experiments, we used a computer with an Intel i7-4770 3.40 GHz CPU and a Nvidia Titan GPU to run the experiments. The training procedures all ran on the GPU to gain a fast speed. Since we train our networks with small training sets, the training can complete in several hours, not in days or weeks as some other applications [32]. It is relatively fast for CNNs training.

Since we mainly focus on small sets training with CNNs, only limited samples are chosen for training. In the most extreme case, we choose only 3 samples per class. Besides, other numbers of sample per class, from 4 to 15, are also investigated in the experiments. The rest samples those are not in the training set are taken into the test set. Due to the small training set, the performance can be heavily affected by the selected training samples. That is, training samples with good representation will bring good performance and vice versa, thus the classification accuracy may vary according to the selected training samples. Each experiment is repeated ten times with different training sets to reduce the influence of random effects, and the average results are reported. With respect to the evaluation metrics, overall accuracy (OA) and kappa coefficient (κ) are used as measures of accuracy. OA is the sum of the correctly classified samples divided by the total number of test samples, which is defined as:

$$\mathrm{OA} = \frac{\text{number of correctly classified samples}}{\text{number of test samples}}, \tag{1}$$

where κ is a statistical measure of the degree of agreement, which is computed by accounting for all elements in the confusion matrix [46]. Clearly, the higher the two metrics, the better the results.

4. Experimental results and analysis

In this section, we firstly evaluate the impact of parameters used in the proposed CNN model, including the convolutional kernel size and the adoption of dropout layers. Then the experimental results and analysis on the three real hyperspectral data sets have been reported.

4.1. The impact of convolutional kernel size

As described in Section 2, the convolutional kernel size is 1×1 in the proposed CNNs. Here we evaluate the networks with a different kernel size. The softmax loss function is employed in the training process, which is minimized after training. It is widely accepted that a well trained network should have not only small training loss, but also small test loss. If the training loss is small and the test loss is large, it means that the trained network overfits to the training samples, and the network does not have a good generalization capability.

To investigate the performance of different convolutional kernels, we also trained a network with 3×3 kernels. That means the 1×1 convolutional kernels in the 3 convolutional layers of the proposed network are all replaced with 3×3 kernels. The training and test losses of the two different networks are plotted in Fig. 7. Then we can find that the training losses of the two networks are all converging to 0 with the increasing iteration. But the test loss of the network with 3×3 keeps increasing while that of the

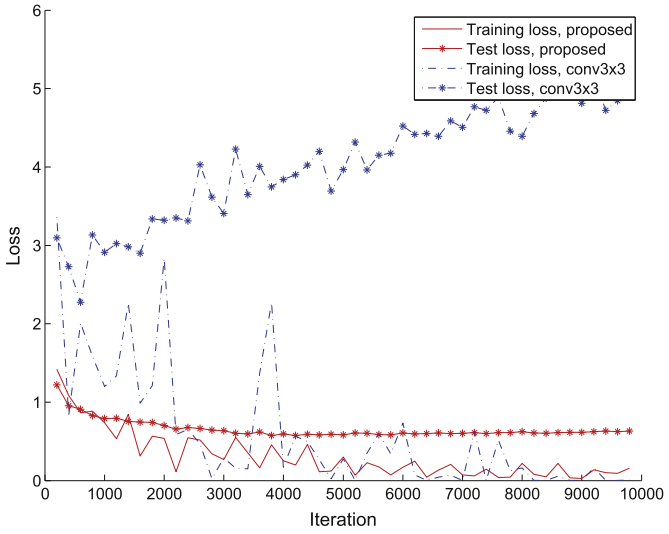


Fig. 7. The training and test losses of the two different networks. *Proposed* network is with 1×1 convolutional kernels, and *conv3* $\times$ *3* is with 3×3 ones.

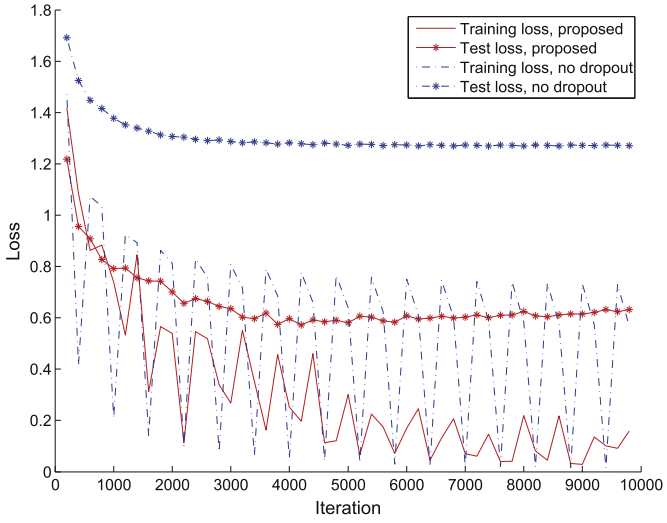


Fig. 8. The training and test losses of the two different networks. *Proposed* network contains two dropout layers, and *no dropout* has no dropout layers.

proposed one keeps decreasing. That means the network with 3×3 kernels encounters with serious overfitting problem, and the performance is getting worse. So our conclusion is that convolutions on spatial domain tend to cause overfitting with limited training samples, and the 1×1 convolutional kernels can gain better a generalization capability.

4.2. Dropout evaluation

In the dropout layer, the output of each neuron in the previous layer is set to zero with a probability. Dropout can reduce the co-adaptation of hidden units and reduce overfitting. We evaluate the effectiveness of dropout in our experiments. Two networks have been trained with the same training set. One network contains two dropout layers with a 0.6 probability as illustrated in Fig. 1, and another one has no dropout layers. Their training losses and test losses are plotted in Fig. 8. Then we can easily find that the training loss and the test loss from the proposed network are all lower than those from the network without a dropout layer. That proved that the dropout strategy can improve the performance of neural networks.

4.3. Experimental results on Indian Pines data set

Fig. 9 displays the overall accuracy (OA) and kappa coefficient measure as functions of the number of training samples per class (3, 4, 5, …, 15). From the figure we can see that, as expected, the increase of the training size has a favorable effect in the performance of all the seven techniques. The two classifiers on the raw hyperspectral data (RAW-KNN and RAW-SVM) deliver the worst results in most cases for both the OA and the kappa measures. The performance on the Gabor features (Gabor-KNN and Gabor-SVM) is better than that on the morphological features (MOR-KNN and MOR-SVM), which is due to the discriminative power of the Gabor features. Further, our CNN method consistently provides the best results. Especially, when the training size is very small, the differences between the algorithms are apparent.

Concerning the small sample size problem, when the training size is only 3 labeled samples per class, the OA and the kappa measure for each class using different approaches are shown in Table 4. From the table, it can be seen that, in most cases, the results obtained by our CNN approach are better than those yielded by the other methods. Fig. 10 displays the training set, test set

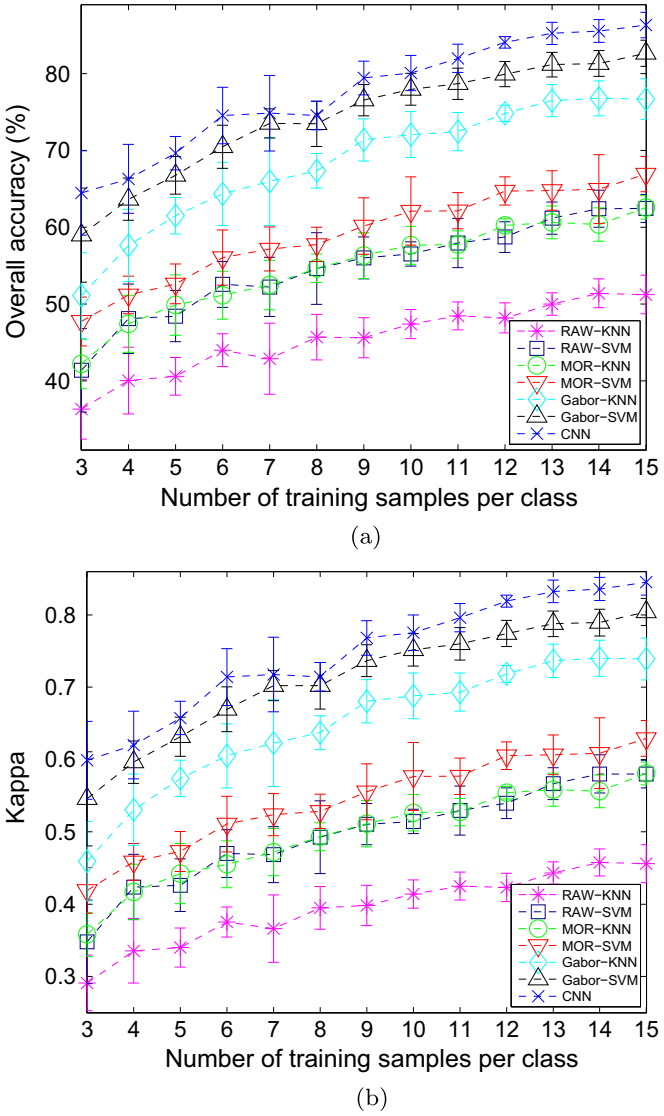


Fig. 9. Indian Pines hyperspectral data: (a) overall accuracy and (b) kappa measure versus the different number of training samples per class.

Table 4
Classification accuracy (%) and kappa measure for the Indian Pines data on the test set with 3 labeled samples per class as training set.

Class	RAW-KNN	RAW-SVM	MOR-KNN	MOR-SVM	Gabor-KNN	Gabor-SVM	CNN
C1	76.08	75.88	85.37	84.09	96.73	94.56	**99.56**
C2	26.73	42.42	61.20	69.99	87.24	**96.29**	82.57
C3	21.48	26.43	25.23	35.03	35.61	**58.84**	47.18
C4	31.54	39.65	31.70	33.32	44.87	66.32	**71.43**
C5	53.83	54.72	65.77	67.55	**100.00**	**100.00**	96.28
C6	18.44	21.29	27.83	33.98	23.56	19.31	**47.36**
C7	37.19	56.88	36.38	52.48	61.13	61.13	**72.85**
C8	52.32	53.69	59.36	65.31	67.38	69.17	**75.45**
C9	16.78	20.88	48.63	55.46	34.74	38.19	**75.35**
C10	79.05	77.86	84.18	84.98	**100.00**	**100.00**	**100.00**
C11	25.69	35.34	47.50	60.21	**91.34**	55.41	80.47
C12	70.58	83.95	87.16	89.30	**100.00**	82.35	**100.00**
C13	21.39	18.93	31.45	31.16	34.17	43.60	**48.53**
C14	30.70	32.30	28.36	31.64	54.03	**57.93**	53.90
C15	44.77	58.65	35.99	42.64	39.11	62.76	**86.55**
C16	83.21	82.81	82.34	84.16	70.81	67.46	**99.06**
Overall	36.30	41.36	42.19	47.72	50.89	58.31	**64.19**
Kappa	0.291	0.348	0.358	0.419	0.458	0.536	**0.599**

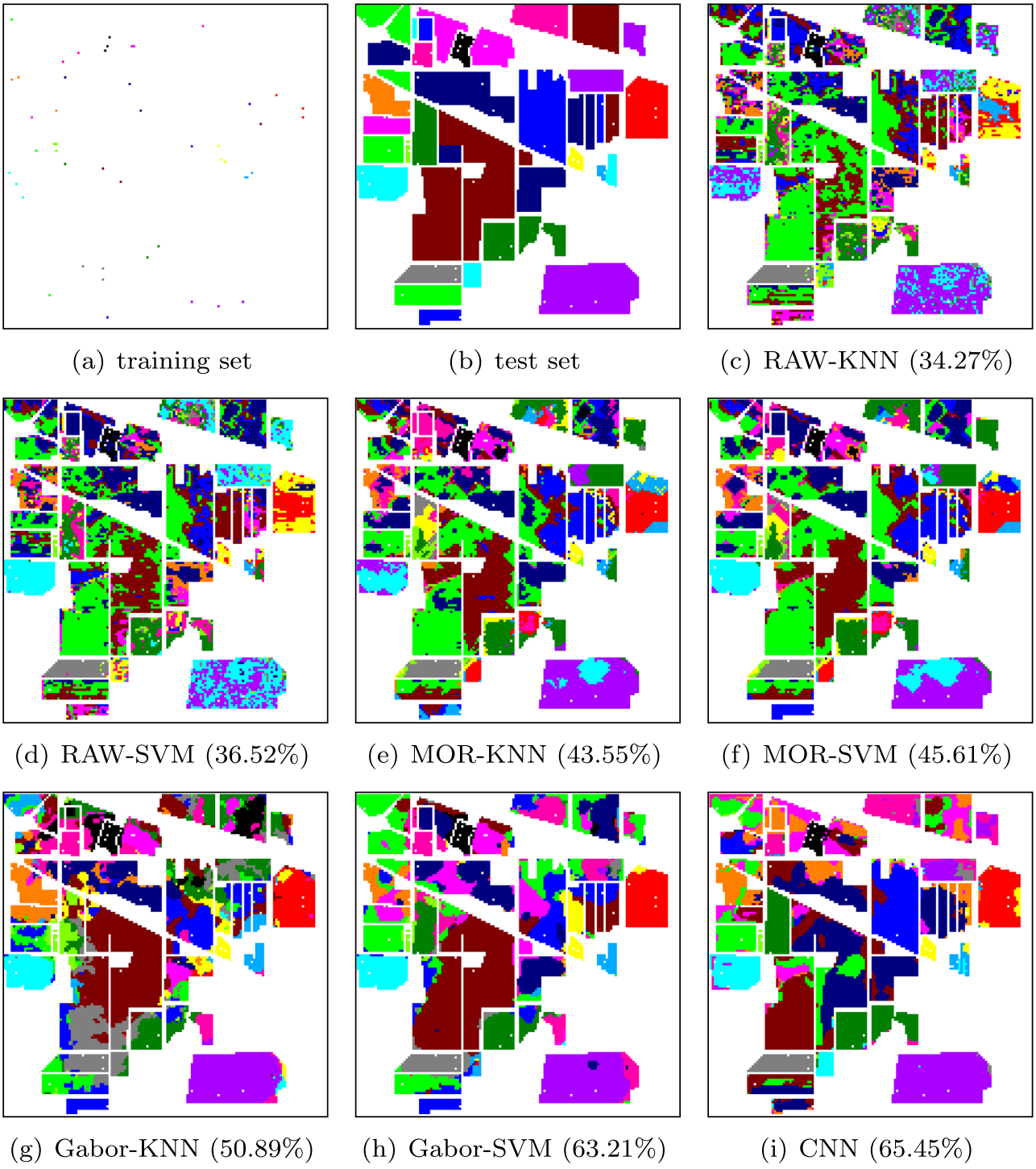


Fig. 10. Indian Pines image: (a) training set (3 labeled samples per class) and (b) test set (the rest labeled samples). Classification maps obtained by (c) RAW-KNN, (d) RAW-SVM, (e) MOR-KNN, (f) MOR-SVM, (g) Gabor-KNN, (h) Gabor-SVM and (i) CNN (the percentage in the brackets is the corresponding accuracy).

and the classification maps on the test set obtained from the various techniques in a single experiment. It can be visually seen that the map of CNN (Fig. 10(i)) is in better accordance with the real map (Fig. 10(b)) than the others, demonstrating the superiority of the proposed CNN method.

4.4. Experimental results on Salinas data set

The classification results of the seven methods with various number of labeled samples are reported in Fig. 11. It can be observed from Fig. 5 that the spatial distribution of land covers is regular, while the spatial resolution of the data is 3.7 m per pixel, so the classification accuracies are generally higher than the above Indian Pines data set. Likewise, the classifiers on the spectral features give the worst results in most cases. Meanwhile, when the number of training samples per class is small, the performance of the Gabor features is lower than that of the morphological features, which is even worse than the spectral features. This is mainly due to that the most representative Gabor features cannot be properly selected with small training set. Similarly, our CNN model consistently achieves the best results.

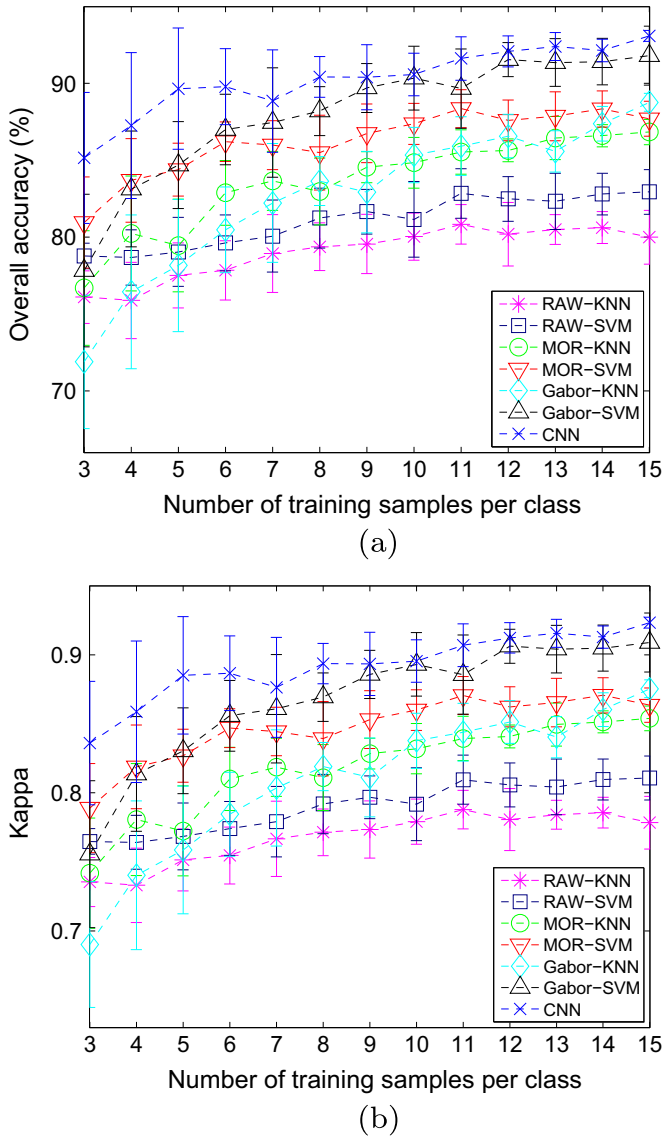


Fig. 11. Salinas hyperspectral data: (a) overall accuracy and (b) kappa measure versus the different number of training samples per class.

When only 3 labeled samples per class are chosen as the training set, i.e., 48 for sixteen classes in total, the detailed classification results with different methods are listed in Table 5. It can be seen that most of the accuracies achieved by our proposed CNN model are higher than the other methods. Specifically, the OA and the kappa measure of the proposed CNN are 85.24% and 0.836, respectively, in comparison to the OA and kappa of 76.10% and 0.735, 78.77% and 0.765, 76.70% and 0.742, 80.95% and 0.789, 71.89% and 0.690, 77.83% and 0.756 for RAW-KNN, RAW-SVM, MOR-KNN, MOR-SVM, Gabor-KNN and Gabor-SVM, respectively. Moreover, the training set, test set and the classification maps of the seven methods are displayed in Fig. 12. Similarly, the map of CNN (Fig. 12(i)) is in better accordance with the real map (Fig. 12 (b)) than the others, verifying the effectiveness of the proposed CNN architecture.

4.5. Experimental results on PaviaU data set

The PaviaU data set was collected in an urban area, and is different from the previous two. The classification results in Fig. 13

Table 5
Classification accuracy (%) and kappa measure for the Salinas data on the test set with 3 labeled samples per class as training set.

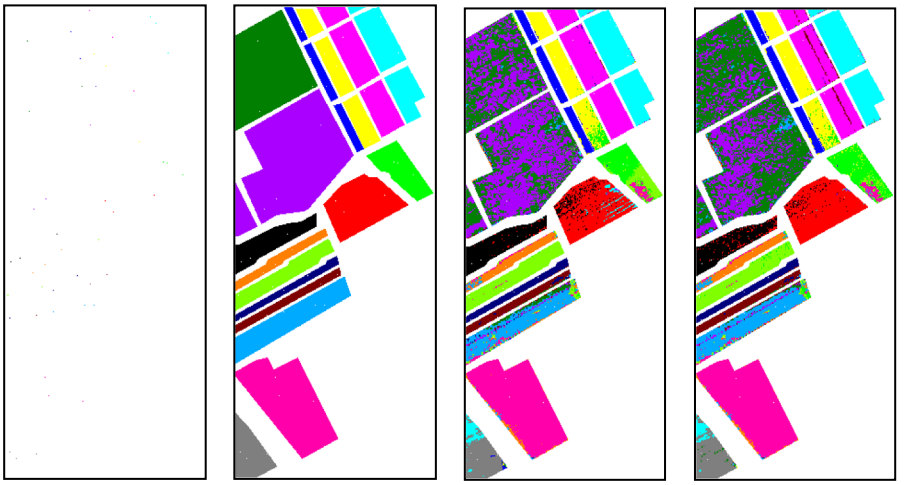
Class	RAW-KNN	RAW-SVM	MOR-KNN	MOR-SVM	Gabor-KNN	Gabor-SVM	CNN
*C*1	86.46	87.82	90.50	90.44	62.12	82.02	**98.36**
*C*2	80.85	85.65	89.11	88.46	64.35	74.21	**98.43**
*C*3	56.37	66.35	39.41	50.99	67.50	68.11	**92.97**
*C*4	88.70	88.54	62.43	80.36	86.45	81.55	**99.46**
*C*5	83.55	83.49	73.98	81.03	51.36	71.51	**91.38**
*C*6	89.13	88.47	82.78	84.12	74.84	85.02	**99.83**
*C*7	90.15	90.10	88.73	88.14	77.63	78.36	**99.68**
*C*8	45.95	47.29	53.62	56.31	47.79	47.46	**68.94**
*C*9	87.45	87.69	84.63	83.10	87.71	87.01	**98.45**
*C*10	46.14	65.12	53.34	70.86	51.56	71.56	**73.31**
*C*11	77.26	76.21	86.52	85.28	76.48	79.66	**90.85**
*C*12	86.68	88.54	66.53	78.33	73.75	80.11	**98.31**
*C*13	89.58	89.23	88.57	88.09	78.01	80.37	**97.43**
*C*14	79.87	80.65	87.68	88.32	78.51	79.85	**94.76**
*C*15	52.85	52.88	63.12	66.09	60.14	**66.61**	63.75
*C*16	68.03	74.90	51.83	63.18	77.31	77.63	**89.83**
Overall	76.10	78.77	76.70	80.95	71.89	77.83	**85.24**
Kappa	0.735	0.765	0.742	0.789	0.690	0.756	**0.836**

also show that the proposed method outperforms other methods obviously. When the number of training samples is very small, only 3 samples for each class are involved as in the previous experiments. The proposed CNN can achieve highest OA at 67.85% while the second highest OA is achieved by MOR-SVM at 61.24%. There are 10.8% improvement. Meanwhile, the proposed CNN all achieves best kappa measure compared with other methods (Table 6).

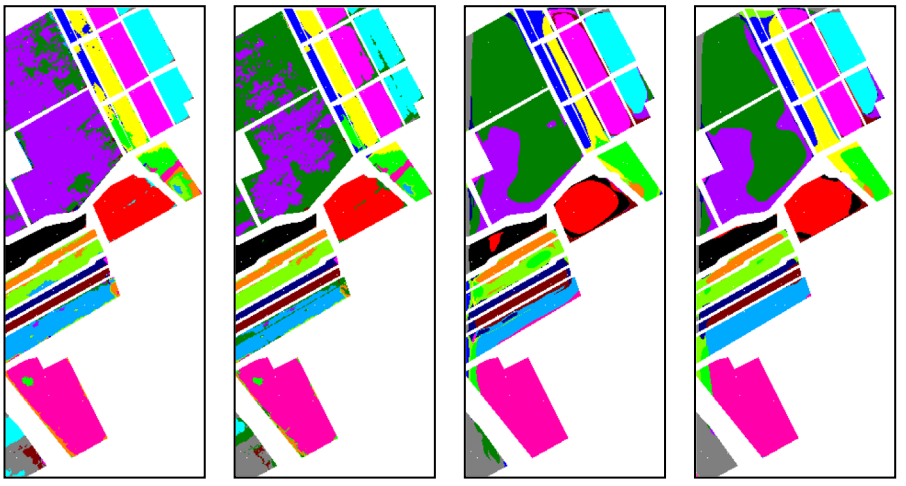
5. Conclusions and discussions

In most deep learning applications, CNNs are used for huge amount of training samples to learn powerful classifiers. In this paper, a well designed CNN structure has been proposed, which can handle limited training samples. The designing principles include data augmentation, appropriate convolutional kernel size, larger drop rates in the dropout layers, discard the most commonly used max pooling layers and full connection layers, etc. The CNNs have been tested in HSI classification on three popular data sets and achieved outstanding performance even with only very few training samples per class. Our work shows that the deep learning model CNNs can handle classification problems with few training samples without losing its generalization if the networks are well designed. Since all the source code and model files of the proposed CNN framework will be released for the community, we believe that our contribution can serve to stimulate more research of deep learning models and methods in the hyperspectral image processing area.

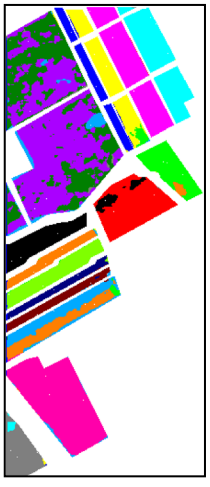
In recent years, many neural networks which are deeper were proposed. For example, *GoogLeNet* is a 22 layer deep network [33] and won object detection with additional training data in ImageNet contest 2014. Therefore, in our future work, deeper networks for hyperspectral image classification will be investigated to gain higher performance. Besides, the computational load of the proposed method will be further decreased to make the method more suitable for hyperspectral images with a large spatial coverage.



(a) training set (b) test set (c) RAW-KNN (78.18%) (d) RAW-SVM (78.72%)



(e) MOR-KNN (79.72%) (f) MOR-SVM (80.33%) (g) Gabor-KNN (73.81%) (h) Gabor-SVM (79.31%)



(i) CNN (81.75%)

Fig. 12. Salinas image: (a) training set (3 labeled samples per class) and (b) test set (the rest labeled samples). Classification maps obtained by (c) RAW-KNN, (d) RAW-SVM, (e) MOR-KNN, (f) MOR-SVM, (g) Gabor-KNN, (h) Gabor-SVM and (i) CNN (the percentage in the brackets is the corresponding accuracy).

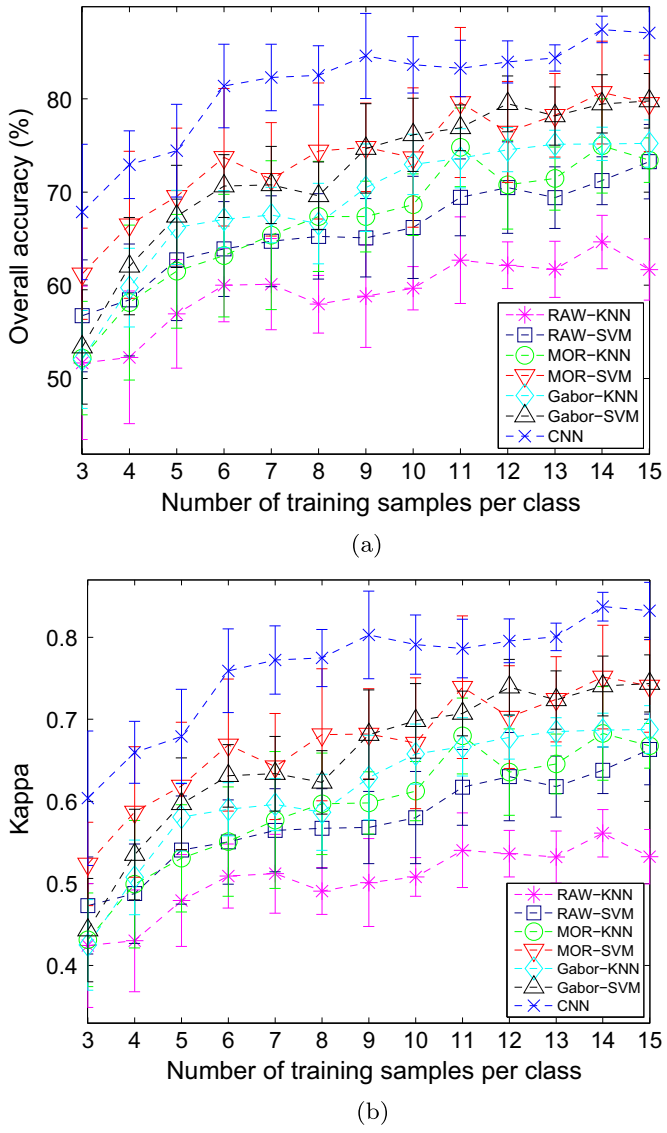


Fig. 13. PaviaU hyperspectral data: (a) overall accuracy and (b) kappa measure versus the different number of training samples per class.

Table 6
Classification accuracy (%) and kappa measure for the PaviaU data on the test set with 3 labeled samples per class as training set.

Class	RAW-KNN	RAW-SVM	MOR-KNN	MOR-SVM	Gabor-KNN	Gabor-SVM	CNN
*C*1	57.23	59.60	47.63	52.92	42.82	52.15	**69.42**
*C*2	38.66	47.82	36.91	53.81	44.69	43.50	**58.51**
*C*3	41.35	54.25	47.91	57.27	64.23	67.88	**78.86**
*C*4	83.04	75.65	81.60	85.86	37.91	42.17	**99.02**
*C*5	86.77	96.96	95.18	95.41	96.92	98.07	**100.00**
*C*6	46.98	47.47	66.96	61.56	**89.60**	87.02	63.35
*C*7	88.39	87.79	80.62	80.86	67.02	71.85	**93.82**
*C*8	55.44	58.12	63.73	**69.63**	35.35	35.56	57.54
*C*9	99.79	**99.67**	75.56	84.76	61.26	61.97	97.67
Overall	51.68	56.73	52.19	61.24	52.05	53.38	**67.85**
Kappa	0.424	0.473	0.431	0.524	0.425	0.443	**0.604**

Acknowledgment

The authors would like to thank Prof. M. Crawford for providing the AVIRIS Indian Pines hyperspectral data, along with the training and test set. This work was jointly supported by grants from National Natural Science Foundation of China (61671307, 61271022 and 61602244), Guangdong Special Support Program of Top-notch Young Professionals (2015TQ01X238), Guangdong Foundation of Outstanding Young Teachers in Higher Education Institutions (Yq2013143) and Shenzhen Scientific Research and Development Funding Program (JCYJ20140418095735628, JCYJ20160422093647889, SGLH20150206152559032 and JCYJ20150324141711699).

References

[1] D. Manolakis, D. Mardon, G.A. Shaw, Hyperspectral image processing for automatic target detection applications, Lincoln Lab. J. 14 (1) (2003) 79–115.
[2] J.M. Bioucas-Dias, A. Plaza, G. Camps-Valls, P. Scheunders, N.M. Nasrabadi, J. Chanussot, Hyperspectral remote sensing data analysis and future challenges, IEEE Geosci. Remote Sens. Mag. 1 (2) (2013) 6–36.
[3] Q. Du, L. Zhang, B. Zhang, X. Tong, P. Du, J. Chanussot, Foreword to the special issue on hyperspectral remote sensing: theory, methods, and applications, IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens. 6 (2) (2013) 459–465.
[4] N. Younan, S. Aksoy, R. King, Foreword to the special issue on pattern recognition in remote sensing, IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens. 5 (5) (2012) 1331–1334.
[5] A. Wang, J. Lu, J. Cai, G. Wang, T.J. Cham, Unsupervised joint feature learning and encoding for RGB-D scene labeling, IEEE Trans. Image Process. 24 (11) (2015) 4459–4473.
[6] A. Plaza, J.M. Bioucas-Dias, A. Simic, W.J. Blackwell, Foreword to the special issue on hyperspectral image and signal processing, IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens 5 (2) (2012) 347–353.
[7] Q. Tong, Y. Xue, L. Zhang, Progress in hyperspectral remote sensing science and technology in China over the past three decades, IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens 7 (1) (2014) 70–91.
[8] A. Martínez-usó, F. Pla, J.M. Sotoca, P. García-sevilla, Clustering-based hyperspectral band selection using information measures, IEEE Trans. Geosci. Remote Sens. 45 (12) (2007) 4158–4171.
[9] P. Zhong, P. Zhang, R. Wang, Dynamic learning of SMLR for feature selection and classification of hyperspectral data, IEEE Trans. Geosci. Remote Sens. Lett. 5 (2) (2008) 280–284, http://dx.doi.org/10.1109/LGRS.2008.915930.
[10] G.F. Hughes, On the mean accuracy of statistical pattern recognizers, IEEE Trans. Inf. Theory 14 (1) (1968) 55–63.
[11] S. Kumar, J. Ghosh, M.M. Crawford, Best-bases feature extraction algorithms for classification of hyperspectral data, IEEE Trans. Geosci. Remote Sens. 39 (7) (2001) 1368–1379.
[12] L.O. Jimenez-Rodrguez, E. Arzuaga-Cruz, M. Velez-Reyes, Unsupervised linear feature-extraction methods and their effects in the classification of high-dimensional data, IEEE Trans. Geosci. Remote Sens. 45 (2) (2007) 469–483.
[13] A. Agarwal, T. El-Ghazawi, H. El-Askary, J. Le-Moigne, Efficient hierarchical-PCA dimension reduction for hyperspectral imagery, in: IEEE International Symposium on Signal Processing and Information Technology, 2007, pp. 353–356.
[14] M. Fauvel, J. Chanussot, J.A. Benediktsson, Kernel principal component analysis for the classification of hyperspectral remote sensing data over urban areas, EURASIP J. Adv. Signal Process. 2009 (2009) 1–14.
[15] J. Wang, C.-I. Chang, Independent component analysis-based dimensionality reduction with applications in hyperspectral image analysis, IEEE Trans. Geosci. Remote Sens. 44 (6) (2006) 1586–1600, http://dx.doi.org/10.1109/TGRS.2005.863297.
[16] C. Xu, C. Lu, J. Gao, W. Zheng, T. Wang, S. Yan, Discriminative analysis for symmetric positive definite matrices on lie groups, IEEE Trans. Circuits Syst. Video Technol. 25 (10) (2015) 1576–1585.
[17] B.S. Serpico, L. Bruzzone, A new search algorithm for feature selection in hyperspectral remote sensing images, IEEE Trans. Geosci. Remote Sens. 39 (7) (2001) 1360–1367.
[18] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, J. Mach. Learn. Res. 3 (2003) 1157–1182.
[19] C.-I. Chang, S. Wang, Constrained band selection for hyperspectral imagery, IEEE Trans. Geosci. Remote Sens. 44 (6) (2006) 1575–1585.
[20] JS. Jia, G. Tang, J. Zhu, Q. Li, A novel ranking-based clustering approach for hyperspectral band selection, IEEE Trans. Geosci. Remote Sens. 54(1) (Jan. 2016), http://dx.doi.org/10.1109/TGRS.2015.2450759.
[21] Y. Qian, F. Yao, S. Jia, Band selection for hyperspectral imagery using affinity propagation, IET Comput. Vis. 3 (4) (2009) 213–222, http://dx.doi.org/10.1049/iet-cvi.2009.0034.
[22] S. Jia, Z. Ji, Y. Qian, L. Shen, Unsupervised band selection for hyperspectral imagery classification without manual band removal, IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens. 5 (2) (2012) 531–543.
[23] M. Fauvel, Y. Tarabalka, J.A. Benediktsson, J. Chanussot, J.C. Tilton, Advances in spectral-spatial classification of hyperspectral images, Proc. IEEE 101 (3) (2013) 652–675.
[24] J.A. Benediktsson, J.A. Palmason, J.R. Sveinsson, Classification of hyperspectral data from urban areas based on extended morphological profiles, IEEE Trans. Geosci. Remote Sens. 43 (3) (2005) 480–491.

[25] D. Tuia, M. Volpi, M. Dalla Mura, A. Rakotomamonjy, R. Flamary, Automatic feature learning for spatio-spectral image classification with sparse SVM, IEEE Trans. Geosci. Remote Sens. 52 (10) (2014) 6062–6074.
[26] S. Jia, X. Zhang, Q. Li, Spectral-spatial hyperspectral image classification using $\ell_{1/2}$ regularized low-rank representation and sparse representation-based graph cuts, IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens. 8 (6) (2015) 2473–2484.
[27] S. Jia, Y. Xie, G. Tang, J. Zhu, Spatial-spectral-combined sparse representation-based classification for hyperspectral imagery, Soft Comput. (2014), pp 1-10, http://dx.doi.org/10.1007/s00500-014-1505-4.
[28] L. Shen, S. Jia, Three-dimensional Gabor wavelets for pixel-based hyperspectral imagery classification, IEEE Trans. Geosci. Remote Sens. 49 (12) (2011) 5039–5046.
[29] S. Jia, Z. Zhu, L. Shen, Q. Li, A two-stage feature selection framework for hyperspectral image classification using few labeled samples, IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens. 7 (4) (2014) 1023–1035.
[30] S. Jia, L. Shen, Q. Li, Gabor feature-based collaborative representation for hyperspectral imagery classification, IEEE Trans. Geosci. Remote Sens. 53 (2) (2015) 1118–1129.
[31] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.
[32] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems (NIPS), 2012, pp. 1097–1105.
[33] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.
[34] X. Liang, C. Xu, X. Shen, J. Yang, S. Liu, J. Tang, L. Lin, S. Yan, Human parsing with contextualized convolutional neural network, in: International Conference on Computer Vision (ICCV), 2015.
[35] S. Liu, X. Liang, L. Liu, X. Shen, J. Yang, C. Xu, L. Lin, X. Cao, S. Yan, Matching-CNN meets KNN: quasi-parametric human parsing, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.
[36] C. Xu, C. Lu, X. Liang, J. Gao, W. Zheng, T. Wang, S. Yan, Multi-loss regularized deep neural network, IEEE Trans. Circuits Syst. Video Technol. http://dx.doi.org/10.1109/TCSVT.2015.2477937.
[37] C. Gan, N. Wang, Y. Yang, D.-Y. Yeung, A. G. Hauptmann, Devnet: a deep event network for multimedia event detection and evidence recounting, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.
[38] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: a large-scale hierarchical image database, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009, pp. 248–255.
[39] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.
[40] Y. Chen, Z. Lin, X. Zhao, G. Wang, Y. Gu, Deep learning-based classification of hyperspectral data, IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens. 7 (6) (2014) 2094–2107.
[41] W. Hu, Y. Huang, L. Wei, F. Zhang, H. Li, Deep convolutional neural networks for hyperspectral image classification, J. Sens. (2015), http://dx.doi.org/10.1155/2015/258619.
[42] M. Lin, Q. Chen, S. Yan, Network in network, in: International Conference on Learning Representations (ICLR), 2014.
[43] N. Srivastava, Improving neural networks with dropout (Master's thesis), University of Toronto, 2013.
[44] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: convolutional architecture for fast feature embedding, arXiv preprint arXiv:1408.5093.
[45] Caffe Website. ⟨http://caffe.berkeleyvision.org/⟩.
[46] J.A. Richards, Remote Sensing Digital Image Analysis: An Introduction, Springer, 2013, ISBN 978-3-642-30062-2/978-3-642-30061-5.

Shiqi Yu received his B.E. degree in Computer Science and Engineering from the Chu Kochen Honors College, Zhejiang University, in 2002, and Ph.D. degree in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Sciences, in 2007. He worked as an Assistant Professor and then as an Associate Professor in the Shenzhen Institutes of Advanced Technology, Chinese Academy of Science from 2007 to 2010. Currently, he is an Associate Professor in the College of Computer Science and Software Engineering, Shenzhen University, China. He especially focuses on image classification and related research topics.

Sen Jia received his B.E. and Ph.D. degrees from College of Computer Science, Zhejiang University, in 2002 and 2007, respectively. He is currently an Associate Professor with the College of Computer Science and Software Engineering, Shenzhen University, China. His research interests include hyperspectral image processing, signal and image processing, pattern recognition and machine learning.

Chunyan Xu received the Ph.D. degree from the School of Computer Science and Technology, Huazhong University of Science and Technology, 2015. From 2013 to 2015, she was a Visiting Scholar in the Department of Electrical and Computer Engineering at National University of Singapore. Now she is a Lecturer in the School of Computer Science and Engineering from Nanjing University of Science and Technology, Nanjing, China. Her research interests include computer vision, manifold learning and deep learning.