Taylor & Francis
Taylor & Francis Group

Remote Sensing Letters

ISSN: 2150-704X (Print) 2150-7058 (Online) Journal homepage: http://www.tandfonline.com/loi/trsl20

Spectral-spatial classification of hyperspectral imagery using a dual-channel convolutional neural network

Haokui Zhang, Ying Li, Yuzhu Zhang & Qiang Shen

To cite this article: Haokui Zhang, Ying Li, Yuzhu Zhang & Qiang Shen (2017) Spectral-spatial classification of hyperspectral imagery using a dual-channel convolutional neural network, Remote Sensing Letters, 8:5, 438-447, DOI: 10.1080/2150704X.2017.1280200

To link to this article: https://doi.org/10.1080/2150704X.2017.1280200

Published online: 18 Jan 2017.

REMOTE SENSING LETTERS, 2017
VOL. 8, NO. 5, 438–447
http://dx.doi.org/10.1080/2150704X.2017.1280200

Taylor & Francis
Taylor & Francis Group

Spectral-spatial classification of hyperspectral imagery using a dual-channel convolutional neural network

Haokui Zhang[a], Ying Li[a], Yuzhu Zhang[a] and Qiang Shen[b]

[a]School of Computer Science, Northwestern Polytechnical University, Xi' an, China; [b]Institute of Mathematics, Physics and Computer Science, Aberystwyth University, Aberystwyth, UK

ABSTRACT

In this article, a novel dual-channel convolutional neural network (DC-CNN) framework is proposed for accurate spectral-spatial classification of hyperspectral image (HSI). In this framework, one-dimensional CNN is utilized to automatically extract the hierarchical spectral features and two-dimensional CNN is applied to extract the hierarchical space-related features, and then a softmax regression classifier is used to combine the spectral and spatial features together and predict classification results eventually. To overcome the problem of the limited available training samples in HSIs, we propose a simple data augmentation method which is efficient and effective for improving HSI classification accuracy. For comparison and validation, we test the proposed method along with three other deep-learning-based HSI classification methods on two real-world HSI data sets. Experimental results demonstrate that our DC-CNN-based method outperforms the state-of-the-art methods by a considerable margin.

ARTICLE HISTORY
Received 23 October 2016
Accepted 3 January 2017

1. Introduction

Hyperspectral remote sensors capture digital images in hundreds of continuous narrow spectral bands and produce hyperspectral imagery (HSI) which contains both spectral and spatial information. The rich spectral information of HSI has been widely used in many different applications, such as environmental management, agriculture and mineralogy. Classification of each pixel in HSI plays a crucial role in these applications.

Conventional HSI classification methods are often based on spectral information only. In order to address the curse of dimensionality, spectral features are extracted via some feature extraction methods such as principal component analysis (Licciardi et al. 2012), independent component analysis (Villa et al. 2012) and linear discriminant analysis (Bandos, Bruzzone, and Camps-Valls 2009). Then these extracted spectral features are combined with some classifiers to finish the HSI classification task. However, classification algorithms exploiting only the spectral information fail to capture the important spatial variability observed for high resolution data, generally resulting in lower performance. To improve classification performance, a natural idea is to design classifiers using the spectral-spatial information jointly, in order to incorporate the spatial structure into the pixel-level classifiers. The spatial dependence is

CONTACT Ying Li ✉ lybyp@nwpu.edu.cn School of Computer Science, Northwestern Polytechnical University, Xi' an, Shaanxi 710129, China

extracted in advance through various spatial filters such as morphological profiles (Benediktsson, Palmason, and Sveinsson 2005), attribute profiles (Dalla Mura et al. 2011) and low-rank representation (Jia, Zhang, and Li 2015). These transformed spatial features are combined with the spectral features, which in some cases undergo dimensionality reduction, to perform pixel-wise classification.

However, most conventional HSI classification methods are commonly based on handcrafted features and shallow learning models, relying highly on specific domain knowledge. The design of handcrafted features can be tedious and is typically sub-optimal. Recently, deep learning has emerged as one of the state-of-the-art machine learning techniques, with great potential in the field of image processing. Rather than using handcrafted features, deep learning aims to learn data-adaptive, hierarchical and distributed representation from raw data. Deep learning has achieved impressive performance in the field of computer vision and image processing, such as image classification (Krizhevsky, Sutskever, and Hinton 2012), object detection (Girshick et al. 2014) and depth estimation from a single image (Liu, Shen, and Lin 2015)

Very recently, deep learning methods have been introduced into hyperspectral data classification (Chen et al. 2014; Chen, Zhao, and Jia 2015; Yue, Mao, and Li 2016; Slavkovikj et al. 2015). For example, unsupervised feature learning was applied to HSI classification. Chen, Zhao, and Jia (2015) proposed the stacked autoencoder (SAE) and deep belief network (DBN)-based spectral-spatial features extraction and classification frameworks. Although SAE and DBN can extract deep features hierarchically in a layer-wise training fashion, the training samples composed of image patches should be flattened to one-dimension to meet the input requirement of the models. However, the flattened training samples do not retain the same spatial information that the original image may contain. Moreover, SAE and DBN are unsupervised and do not directly use the label information when learning the features. Slavkovikj et al. (2015) proposed to use convolutional neural network (CNN) for HSI classification where the spectral-dominated features were extracted from a small neighbourhood. Yue et al. (2015) adopted the CNN model on the first a few principal component (PC) bands of the original HSI data to extract spatial-dominated features. These two CNN-based methods did not fully exploit the spatial and spectral information simultaneously. Yue, Mao, and Li (2016) proposed a spectral-spatial deep learning framework for HSI classification where the spectral features and spatial features were extracted via SAE and CNN separately. However, due to the intertwined connection of different layers in the SAE model, it demands the training of many parameters. In addition, almost all of the deep learning methods mentioned above do not consider the problem of the limited available training samples.

In this article, we introduce a dual-channel CNN (DC-CNN) model into HSI classification where spectral features and spatial features are extracted via one-dimensional CNN (1D CNN) and two-dimensional CNN (2D CNN) respectively, and then a softmax regression classifier (Liao and Chin 2007) is used to combine the spectral and spatial features together and predict classification results eventually. Furthermore, to address the problem of over-fitting caused by limited training samples of HSIs, we propose a simple but effective data augmentation method for improving classification accuracy. We compared our DC-CNN-based method with three deep learning based methods on two real HSI data sets. Experimental results demonstrate that the proposed method significantly outperforms other methods in terms of classification accuracy.

2. Proposed method

In this section, we explain in detail the basic operations of DC-CNN based method and elaborate on how to train this network.

2.1. *1D-CNN-based spectral feature extraction*

The most significant advantage of CNN is that it offers an algorithmic means to extract features directly from the raw input imagery. It is thus simpler to use, and more applicable in generic domains. Inspired by this, we introduce a 1D CNN model for spectral feature extraction. The 1D kernels instead of the conventional 2D kernels are exploited to effectively capture intrinsic semantic content along the 1D spectral dimensions.

Typically, a CNN alternatively stacks several convolutional layers and pooling layers to form a deep architecture. In the 1D convolution operation, the input data is convolved with 1D kernels (the length of 1D kernels is the size of the receptive field), and then go through the activation function to form the output data (feature vectors). The value at position x on the jth feature vector in the lth layer is given by

$$v_{l,j}^{x} = f\left(\sum_{m}\sum_{h=0}^{H_l-1} k_{l,j,m}^{h} v_{(l-1),m}^{(x+h)} + b_{l,j}\right) \tag{1}$$

where l is the layer number, j is the feature vector number in the lth layer, $b_{l,j}$ is the bias of the jth feature vector in the lth layer and $f()$ is the activation function. m indexes over the set of feature vectors in the $(l-1)$th layer connected to the current feature vector. Finally, $k_{l,j,m}^{h}$ is the hth value of the kernel connected to the mth feature vector in the $(l-1)$th layer, H_l is the length of the kernel. There are different kinds of activation functions available to apply. In our CNN implementation, the rectified linear unit (ReLU) (Krizhevsky, Sutskever, and Hinton 2012) is utilized as the non-linear activation function in the convolutional layers. Its formula is given as follows:

$$f(x) = \max(0, x) \tag{2}$$

Pooling can be used to reduce the dimensions of feature vectors, offer invariance and increase the receptive field. A pooling layer follows a convolutional layer. The neuron in the pooling layer combines a small 2 × 1 strip of the convolutional layer. The most common pooling operation is max-pooling (Krizhevsky, Sutskever, and Hinton 2012), which is used in our CNN models.

The full flow chart of 1D-CNN-based spectral feature extraction is shown in Figure 1. For one pixel of hyperspectral data which is to be dealt with, a $3 \times 3 \times L$-sized cube is extracted from its eight neighbourhoods as its original input data. The size of spatial neighbourhood is empirically determined through experimental investigation. We have studied the effect of different spatial sizes including 1×1, 3×3 and 5×5, and subsequently chosen 3×3 as the input to obtain a high classification accuracy for the 1D CNN. To meet the input requirement of 1D CNN, the original data is rearranged into nine pixel vectors, and the length of each pixel vector is L. The rearranged features are fed into the 1D CNN to extract the high-level abstract spectral features. Then the extracted features are stretched into a 1D vector $\boldsymbol{F}_{1D}^{1}$ to be fed into a softmax regression classifier to yield the prediction probability vector $\boldsymbol{F}_{1D}^{2}$. Such a 1D CNN

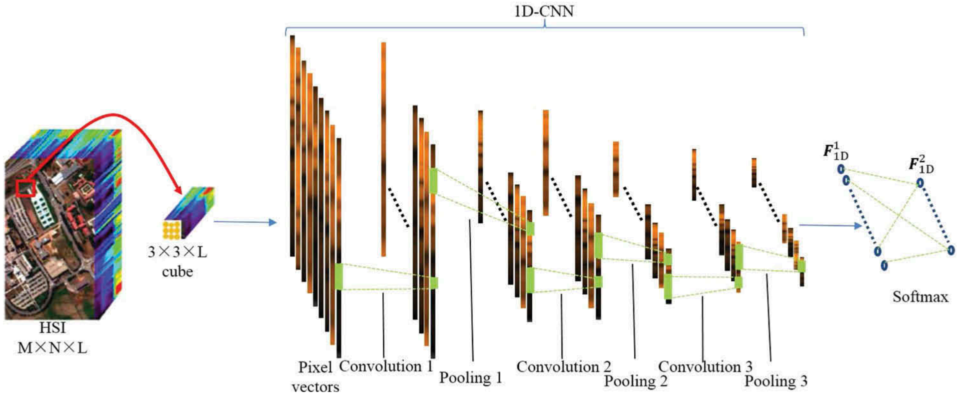

Figure 1. Illustration of 1D-CNN-based spectral feature extraction framework.

system is trained via stochastic gradient descent with momentum and weight decay strategies (Krizhevsky, Sutskever, and Hinton 2012).

2.2. *2D-CNN-based spatial feature extraction*

Similar to 1D CNN, a typical 2D CNN stacks convolutional and pooling layers. In a convolution layer, the input data is convolved with 2D kernels, and then goes through the activation function to form the output data (feature maps). This operation can be formulated as follows:

$$map^{x,y}_{l,j} = f\left(\sum_{m}\sum_{h=0}^{H_l-1}\sum_{w=0}^{W_l-1} k^{h,w}_{l,j,m} map^{(x+h),(y+w)}_{(l-1),m} + b_{l,j}\right) \tag{3}$$

where $k^{h,w}_{l,j,m}$ is the value at the position (h,w) of the kernel connected to the mth feature map in the $(l-1)$th layer, H_l and W_l are the height and width of the kernel, respectively, and $b_{l,j}$ is the bias of the *j*th feature map in the *l*th layer. Pooling is carried out in the similar way to the 1D CNN. The neuron in the pooling layer combines a small 2×2 patch of the convolutional layer.

For a certain pixel in the original HSI, it is natural to consider its neighbouring pixels to extract the spatial features. However, due to the hundreds of channels along the spectral dimension of hyperspectral data, the region-stacked feature vector will result in too large an input dimension. As a result, PCA is commonly executed in the first step to map the data to an acceptable scale with a low information loss and then 2D CNN can be applied for feature extraction and classification.

The full structure of 2D-CNN-based spatial feature extraction framework is shown in Figure 2. Firstly, we reduce the spectral dimension of the original HSI to three (which is empirically chosen as a trade-off between accuracy and computational complexity). Then, for each pixel which is to be dealt with, we choose a relatively large image patch from its neighbouring region as the input of the 2D CNN model. After that, we build deep CNN to extract the high-level spatial features. Finally, the extracted spatial features are vectorized into a 1D vector $\boldsymbol{F}^1_{2D}$ to be fed into a softmax regression classifier to yield the prediction probability vector $\boldsymbol{F}^2_{2D}$, which is similar to the spectral information extraction scheme. In addition, the 2D CNN model contains a large number of parameters which can be prone

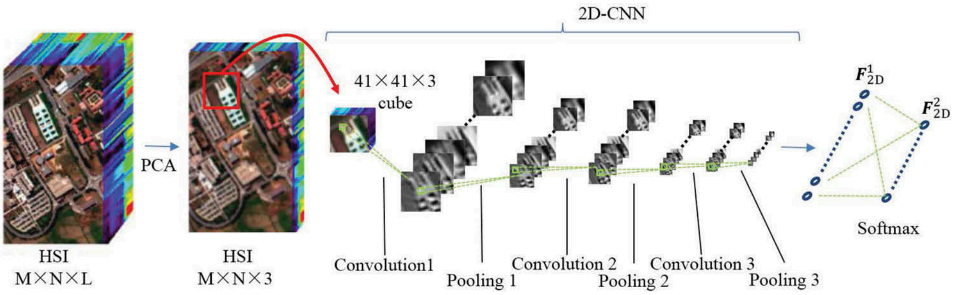

Figure 2. Illustration of 2D-CNN-based spatial feature extraction framework.

to over-fitting. We adopt a normal dropout rate of 50% in the 2D CNN model to handle over-fitting during the training process.

2.3. *DC-CNN-based HSI classification*

HSIs contain both spectral and spatial information. Therefore, the spectral-spatial-feature-based classification methods are more effective than single-feature-based classification methods. In this section, a DC-CNN framework is proposed for spectral-spatial HSI classification, as shown in Figure 3.

For illustration purposes, we divide the DC-CNN-based HSI classification framework into the following three parts. We also represent the pixel to be dealt with as P for concise expression.

(1) **Channel one: 1D-CNN-based deep spectral feature extraction**. In channel one, we extract a data cube with size $3 \times 3 \times L$ from the neighbourhood of P (marked by yellow box in Figure 3) as the input of 1D CNN, and then put this cube though the trained 1D CNN and softmax regression sequentially to obtain the hierarchical spectral features $\boldsymbol{F}^1_{1D}$ and $\boldsymbol{F}^2_{1D}$.

(2) **Channel two: 2D-CNN-based deep spatial feature extraction**. In channel two, we choose a data cube with size $41 \times 41 \times 3$ from the neighbourhood of *P*

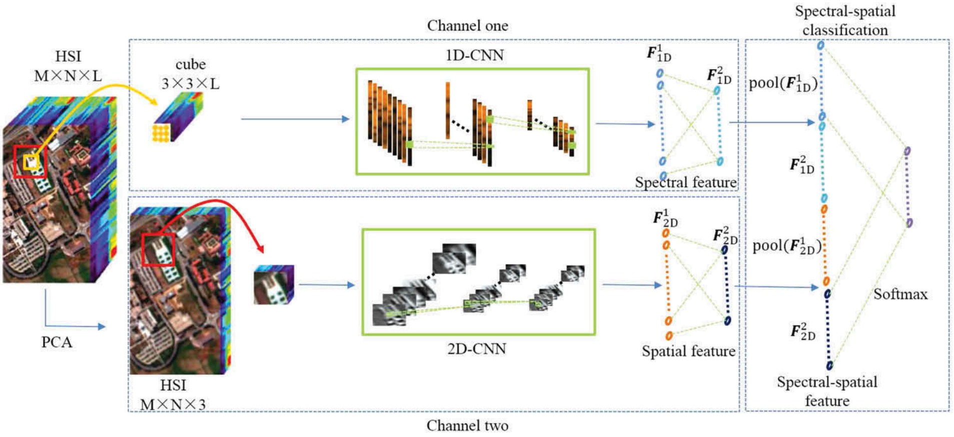

Figure 3. Illustration of DC-CNN-based classification framework.

(marked by red box in Figure 3) as the input of 2D CNN, and then put this cube though trained 2D CNN and softmax regression sequentially to obtain the hierarchical spatial features $\boldsymbol{F}_{2D}^{1}$ and $\boldsymbol{F}_{2D}^{2}$.

(3) **Spectral-spatial classification**. When the spectral and spatial features have been extracted respectively, the final classification step is carried out. Firstly, the spectral and spatial features are concatenated together as follows:

$$F = [\mathrm{pool}(\boldsymbol{F}_{1\mathrm{D}}^{1}), \boldsymbol{F}_{1\mathrm{D}}^{2}, \mathrm{pool}(\boldsymbol{F}_{2\mathrm{D}}^{1}), \boldsymbol{F}_{2\mathrm{D}}^{2}] \tag{4}$$

where $\boldsymbol{F}$ is the concatenated spectral-spatial features, $pool()$ is the max pooling operation which is used to reduce the dimensions of $\boldsymbol{F}_{1D}^{1}$ and $\boldsymbol{F}_{2D}^{1}$. In this way, we can combine the features of the different hierarchies and avoid involving too many parameters in the classification model. Then the softmax regression classifier is applied to predict the class label.

2.4. *Data augmentation*

CNN usually requires a lot of training data to learn deep structure and its related parameters. However, in HSI classification, only the limited labelled samples are available, which may lead to over-fitting. To address this issue, we adopt a simple but effective data augmentation method to generate additional data without introducing extra labelling costs. For each training sample, two forms of data augmentation are used. The first one is rotation: we do this by rotating the original samples by 90°, 180° and 270° respectively. The second form is to flip the data where the original samples are flipped along the horizontal and vertical directions, respectively. After the augmentation operation, the number of training samples can be increased by a factor of six.

3. Experiments

In order to evaluate the efficacy of the proposed DC-CNN method, we compare the proposed method with three deep learning HSI classification approaches: SSDCNN (Yue et al. 2015), SSDL (Yue, Mao, and Li 2016) and CNN (Slavkovikj et al. 2015). Overall accuracy (OA), average accuracy (AA) and kappa statistic (κ) are adopted in this article to measure the classification results of each model. The OA is calculated by the ratio between the correctly classified test samples and the total number of test samples, and the AA is the mean accuracy of each class. The last measurement metric κ is calculated by weighting the measured accuracies. It incorporates both of the diagonal and off-diagonal entries of the confusion matrix and is a robust measure of the degree of agreement. To obtain a more convincing estimate of the capabilities of these methods, for each test data set, we run the experiments 10 times. Each time we randomly choose 5% and 10% from each class of the ground truth data as the training samples for Pavia University and Indian Pines, respectively, and take the remainders as testing samples. All of the methods are implemented based on MatConvNet, a MATLAB toolbox for training CNN models. All of the experiments are implemented on the same desktop with 8 gigabytes of memory, using an Intel Core I5-3470 central processing unit.

For the two data sets used in this article, the DC-CNN structure was set using the same parameters. For the spectral feature extraction part, 1D CNN contains three 1D convolutional layers, each of which is followed by one pooling layer with pooling size [2 1] and stride [2 1], and each convolutional layer contains 36 1D kernels. Three kernels of different size are used: from the first to the third convolutional layers, the 1D kernel sizes are set to 3–7–5. For the varying size setting, we experimented with four possible structures with 1D kernel sizes 3–5–7, 7–5–3, 5–7–3 and 3–7–5; we found 3–7–5 to be the best. For the spatial feature extraction part, three principle components are extracted from the original HSI and then 41 × 41 × 3-sized image patches are extracted as the input data of 2D CNN. 2D CNN contains three convolutional layers, kernel sizes of which are set to 3–7–5. Each 2D convolutional layer contains 36 2D kernels, and is followed by one pooling layer with pooling size [2 2] and stride [2 2].

3.1. *Pavia University scene*

The Pavia University scene was acquired by ROSIS sensor in 2001, ranging from 0.43 to 0.86 um, and has a spatial resolution of 1.3 m per pixel. The corrected data has 103 bands after the 12 noisiest bands are removed and is 610 × 340 pixels in size. The image is differentiated into 9 ground truth classes. Of the 42,776 labelled samples (ground truth) contained within the corrected Pavia University scene, we randomly choose 2144 (5%) labelled samples as training data, and the rest 40,632 (95%) labelled samples are used for testing. As mentioned above, DC-CNN contains three parts, 1D-CNN-based spectral feature extraction, 2D-CNN-based spatial feature extraction and combined-feature-based classification; each part is trained via stochastic gradient descent. On this data set, 1D CNN is trained over 240 epochs (160 epochs with learning rate 0.01, 80 epochs with learning rate 0.001), 2D CNN over 60 epochs (40 epochs with learning rate 0.01, 20 epochs with learning rate 0.001) and the classification part over 15 epochs (10 epochs with learning rate 0.01, 5 epochs with learning rate 0.001). Each iteration randomly takes 40 samples, where weight decay and momentum are set to 0.0005 and 0.9, respectively. The results (the percentages of testing samples that are correctly classified) are listed in Table 1. The results represent the average over 10 runs, and the standard deviation of the average and the training time are also reported. The visual classification results and detailed mappings with different methods are shown in Figure 4.

DC-CNN+Augmentation means DC-CNN which adopted data augmentation method and the best results among the compared methods are shown in bold.

As shown in Table 1, the classification results obtained by DC-CNN provide better accuracies than the other three methods; moreover, data augmentation improved the classification accuracy by a further 0.13%. In Figure 4(e), the reduced misclassification error rate is clearly visible, there being fewer misclassified purple pixels (class 7, Bitumen).

3.2. *Indian Pines scene*

The Indian Pines scene was acquired by AVIRIS sensor in 1992. It consists of 145 × 145 pixels with moderate spatial resolution of 20 m. The number of bands of the corrected

Table 1. Classification results (%) of Pavia University scene using the different methods shown.

Class	SSDCNN	SSDL	CNN	DC-CNN	DC-CNN +Augmentation
OA	93.94 ± 0.0218	94.88 ± 0.0082	96.57 ± 0.0369	99.55 ± 0.0095	**99.68 ± 0.0008**
AA	92.20 ± 0.0419	93.29 ± 0.0112	95.73 ± 0.0403	98.86 ± 0.0293	**99.50 ± 0.0121**
K	91.95 ± 0.0342	93.21 ± 0.0179	95.43 ± 0.0589	99.40 ± 0.0166	**99.58 ± 0.0013**
Training time (s)	266.11	333.90	177.39	572.87	3421.25

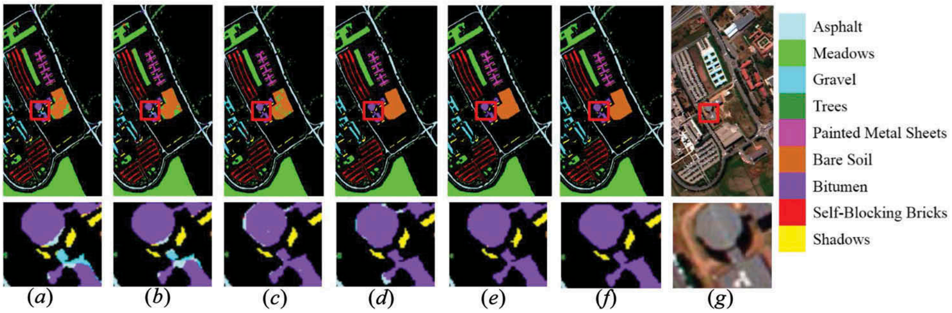

Figure 4. Classification results of Pavia University scene. (a) SSDCNN, OA = 93.94%. (b) SSDL, OA = 94.88%. (c) CNN, OA = 96.57%. (d) DC-CNN, OA = 99.55%. (e) DC-CNN+ Augmentation, OA = 99.68%. (f) Ground truth. (g) False-colour composite.

data has been reduced from 224 to 200 by removing bands covering the region of water absorption. The image contains 16 different categories. The Indian Pines scene contains 10,249 labelled samples, which were randomly separated into two parts. About 1031 samples (10%) are used as training data, and the rest 9218 samples (90%) are used for testing. Both 1D CNN and 2D CNN trained over 240 epochs (160 epochs with learning rate 0.01, 80 epochs with learning rate 0.001). At each iteration, 40 samples are randomly selected from the training set. Weight decay, momentum and dropout rate are set to 0.0005, 0.9 and 0.5, respectively. The classification part trained over 15 epochs (10 epochs with learning rate 0.01, 5 epochs with learning rate 0.001), with each iteration also randomly taking 40 samples. The results are listed in Table 2 and shown in Figure 5.

Table 2. Classification results (%) of Indian Pines scene using the different methods shown.

Class	SSDCNN	SSDL	CNN	DC-CNN	DC-CNN +Augmentation
OA	90.76 ± 0.0301	91.60 ± 0.0625	92.82 ± 0.1867	96.88 ± 0.0099	**98.76 ± 0.0077**
AA	85.52 ± 0.1365	93.96 ± 0.1404	92.97 ± 0.3901	95.38 ± 2.4834	**98.50 ± 0.3150**
K	89.44 ± 0.0369	90.43 ± 0.0872	91.82 ± 0.2424	96.44 ± 0.0128	**98.58 ± 0.0092**
Training time (s)	255.32	328.18	557.89	813.01	4860.57

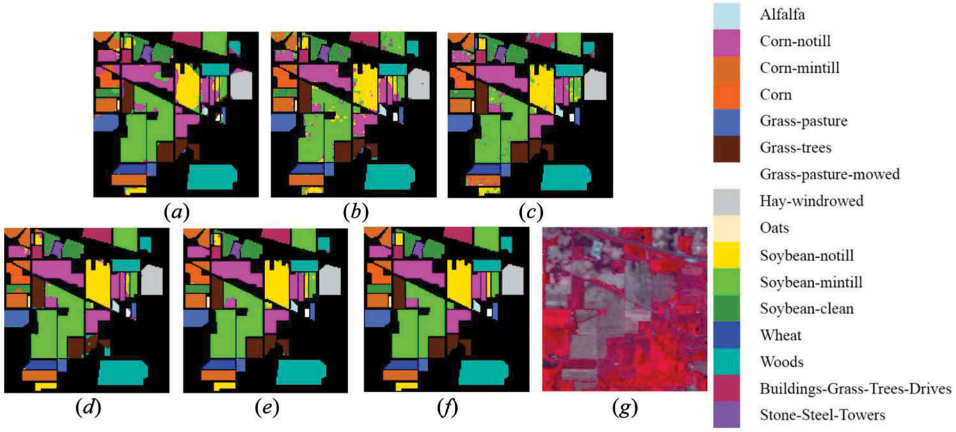

Figure 5. Classification results of Indian Pines scene. (a) SSDCNN, OA = 90.76%. (b) SSDL, OA = 91.60%. (c) CNN, OA = 92.82%. (d) MD-CNN, OA = 96.88%. (e) MD-CNN+ Augmentation, OA = 98.76%. (f) Ground truth. (g) False-colour composite.

DC-CNN+Augmentation means DC-CNN which adopted data augmentation method and the best results among the compared methods are shown in bold.

According to Table 2, DC-CNN again outperforms the state-of-the-art alternative methods by a substantial margin. Compared with the previous data set, data augmentation offers a more significant improvement upon the classification accuracy in this data set, improving the overall classification accuracy by 1.88% compared to DC-CNN. We argue that this visible improvement is likely caused by the initial small training sample size of this image set, since the Indian Pines scene contains fewer labelled samples (10,249, compared to 42,776 for Pavia). We suggest that data augmentation is particularly effective if the training sample size is relatively small and when accuracy may be prioritized over running time.

4. Discussion and conclusion

In this article, for the purpose of improving HSI classification performance, a novel DC-CNN-based spectral-spatial classification framework has been proposed. Due to the fact that HSI contains both spectral information and spatial information, we have adopted 1D CNN and 2D CNN to extract the hierarchical spectral features and the hierarchical spatial features respectively, thereby combining them together to complete the HSI classification task. Although the spatial information changes slowly (given the use of the spatial window of 4141 pixels), for a certain target pixel, its label is the main guiding information. As such, the context information is adopted as the auxiliary information during the training process. To address the problem of overfitting caused by the limited available training samples, we have adopted specialized training choices such as dropout and weight decay. This work is further supported by a simple and efficient data augmentation method. We have compared our DC-CNN method against three other state-of-the-art deep-learning-based HSI classification methods, on two public HSI benchmark data sets.

Our experiments show that the proposed DC-CNN-based HSI classification method achieved the best overall accuracy. However, in the field of HSI classification, there exists many issues to be addressed. For instance, compared with labelled samples, typically, unlabelled samples are much more frequently obtained. Supervised classification methods fail to make full use of these unlabelled samples, whilst unsupervised and semi-supervised classification methods are much more desirable to help deal with such problems. Finally, the present work applies dropout for regularization, further research will investigate the use of alternative Gaussian noise regularization.

Acknowledgement

The authors are grateful to the Editor and reviewers for their constructive comments that have helped improve this work significantly.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by National Key Research and Development Program of China [Grant No. 2016YFB0502502].

References

Bandos, T. V., L. Bruzzone, and G. Camps-Valls. 2009. "Classification of Hyperspectral Images with Regularized Linear Discriminant Analysis." *IEEE Transactions on Geoscience and Remote Sensing* 47 (3): 862–873. doi:10.1109/TGRS.2008.2005729.

Benediktsson, J. A., J. A. Palmason, and J. R. Sveinsson. 2005. "Classification of Hyperspectral Data from Urban Areas Based on Extended Morphological Profiles." *IEEE Transactions on Geoscience and Remote Sensing* 43 (3): 480–491. doi:10.1109/TGRS.2004.842478.

Chen, Y., Z. Lin, X. Zhao, and G. Wang. 2014. "Deep Learning-Based Classification of Hyperspectral Data." *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 7 (6): 2094–2107. doi:10.1109/JSTARS.2014.2329330.

Chen, Y., X. Zhao, and X. Jia. 2015. "spectral–Spatial Classification of Hyperspectral Data Based on Deep Belief Network." *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 8 (6): 2381–2392. doi:10.1109/JSTARS.2015.2388577.

Dalla Mura, M., A. Villa, J. A. Benediktsson, J. Chanussot, and L. Bruzzone. 2011. "Classification of Hyperspectral Images by Using Extended Morphological Attribute Profiles and Independent Component Analysis." *IEEE Geoscience and Remote Sensing Letters* 8 (3): 542–546. doi:10.1109/LGRS.2010.2091253.

Girshick, R., J. Donahue, T. Darell, and J. Malik. 2014. "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Columbus, June 24–27.

Jia, S., X. Zhang, and Q. Li. 2015. "Spectralspatial Hyperspectral Image Classification Using Regularized Low-Rank Representation and Sparse Representation-Based Graph Cuts." *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 8 (6): 2473–2484. doi:10.1109/JSTARS.2015.2423278.

Krizhevsky, A., I. Sutskever, and G. E. Hinton. 2012. "Imagenet Classification with Deep Convolutional Neural Networks." *Conference and Workshop on Neural Information Processing Systems (NIPS)* Lake Tahoe, December 3–8.

Liao, J., and K. Chin. 2007. "Logistic Regression for Disease Classification Using Microarray Data." *Bioinformatics* 23 (15): 1945–1951. doi:10.1093/bioinformatics/btm287.

Licciardi, G., P. R. Marpu, J. Chanussot, and J. A. Benediktsson. 2012. "Linear versus Nonlinear PCA for the Classification of Hyperspectral Data Based on the Extended Morphological Profiles." *IEEE Geoscience and Remote Sensing Letters* 9 (3): 447–451. doi:10.1109/LGRS.2011.2172185.

Liu, F., C. Shen, and G. Lin. 2015. "Deep Convolutional Neural Fields for Depth Estimation from a Single Image." *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Boston, June 7–12.

Slavkovikj, V., S. Verstockt, W. D. Neve, S. V. Hoecke, and R. V. Walle. 2015. "Hyperspectral Image Classification with Convolutional Neural Networks." *ACM International Conference on Multimedia (ACMMM)*. Brisbane, October 26–30.

Villa, A., J. A. Benediktsson, J. Chanussot, and C. Jutten. 2012. "Hyperspectral Image Classification With Independent Component Discriminant Analysis." *IEEE Transactions on Geoscience and Remote Sensing* 49 (12): 4865–4876. doi:10.1109/TGRS.2011.2153861.

Yue, J., S. Mao, and M. Li. 2016. "A Deep Learning Framework for Hyperspectral Image Classification Using Spatial Pyramid Pooling." *Remote Sensing Letters* 7 (9): 875–884. doi:10.1080/2150704X.2016.1193793.

Yue, J., W. Zhao, S. Mao, and H. Liu. 2015. "Spectralspatial Classification of Hyperspectral Images Using Deep Convolutional Neural Networks." *Remote Sensing Letters* 6 (6): 468–477. doi:10.1080/2150704X.2015.1047045.