Hyperspectral image classification based on multiscale convolutional network

Yi Ju

College of computer and information, Hohai University, Nanjing, 211100, China
E-mail: juyi@hhu.edu.cn

Abstract: Hyperspectral Image (HSI) classification is an important task in the field of Hyperspectral Image processing. However, the existing classification methods unable to solve the problems caused by hyperspectral image information redundancy, insufficient image feature utilization and Hughes phenomenon. Aiming at these three problems, a hyperspectral image classification algorithm based on deep learning is proposed. The Multiscale Convolutional Neural Network (MCNN) was used to excavate deep features and realize the learning of multiscale features. Then, the features of different scales were fused and classified. The results show that the proposed algorithm has higher classification accuracy than the traditional ones. Also, it has strong generalization ability and robustness. The effectiveness and feasibility of the proposed algorithm are fully verified.

1. Introduction

Hyperspectral remote sensing is a technique for obtaining many narrow continuous spectra of image data in the visible, near-infrared, mid-infrared and thermal infrared bands[1]. Hyperspectral Image (HSI) have advantages of high resolution, high clarity, and which provide rich spectral and spatial information of the ground objects[2-4]. Currently, HSI have been widely used in military, agriculture, water conservancy and other fields. By analyzing the data of HSI, the target in disguise can be accurately distinguished; the information such as vegetation coverage, salinization degree of soil and water quality can be obtained. HSI classification is an important task of HSI information processing. How to classify images effectively is the hot topic of today's research[5].

Recently, deep learning (DL) develops rapidly along with the development of computers. Convolutional Neural Network (CNN) is an important algorithm in DL[6], which possesses the strong ability to do feature learning. The feature of its structure is very suitable for solving the problems in image area[7]. CNN extract features effectively through local connection, and reduce the number of parameters by sharing weight. The applications in image classification[8], target detection[9] and image super-resolution reconstruction[10] lay the foundation for HSI classification task. However, the high resolution and high information retention lead to the surge of data dimensions and redundant spatial information, which cause low computing efficiency. With the increase of dimension, the accuracy of classification will increase at first, but then decrease significantly—called Hughes phenomenon[11]. Therefore, how to improve classification accuracy of HSI by applying the spectral-spatial information of HSI under the premise of guaranteeing computing efficiency is an urgent problem.

Aiming at the above-mentioned problems, this article proposes a novel deep learning-based algorithm for HSI classification, which combines CNN and Multiscale Convolutional Neural Network (MCNN) to exploit the feature information. Firstly, principal component analysis (PCA) is adopted to reduce the dimensions of HSI to avoid the Hughes phenomenon and speed up the classification process.
Then, to achieve multiscale feature learning and alleviate the overfitting phenomenon, a multiscale convolutional block (MCB) is proposed. MCB extracts the features of different scales by mingle with different kernel sizes in a single convolution operation. Finally, the new images which fuse features in different scales are classified into different land-cover classes.

2. CNN

The essence of CNN is a mathematical model that maps the original matrix to a new feature expression after transformed or degraded multiple times. The structure is generally composed of input layer, convolutional layer, pooling layer, fully connected layer and output layer. Figure 1 shows the framework of CNN. Adopting convolutional layer is as feature extraction layer in order to having multi-layers of feature learning on input statistics. The pooling layer always attaches to the convolution layer, which can process the features that extracted earlier. Its function is to reduce the spatial size of the representation and to reduce the number of parameters and computation in the network, also control overfitting. As the feature mapping layer, the fully connection layer maps the feature into one-dimensional vector in order to highly summarize these features. The output layer predicts the label of pixels by exerting the processed features and calculates the cost with the marked labels. Cost is used for updating the weight and bias by backward propagation algorithm until parameters meet the conditions of stopping updating.

The 2D-CNN convolution formula used in proposed algorithm is as follows:

$$v_{ij}^x = f\left(\sum_{m} \sum_{p=0}^{K} w_{jm}^p \cdot v_{i-1,m}^{x+p} + b_{ij}\right)$$

where $v_{ij}^x$ denotes the input of the $j$th feature map position $x$ of the $i$th convolutional layer, $w_{jm}^p$ and $b_{ij}$ represent the weight and bias, $m$ is the number of feature maps that connect to this neuron, $K$ denotes the size of filter, $f(\cdot)$ generally denotes the activation function. In reality application, Batch Norm and rectified linear units are introduced before activation function in order to reduce the parameters.

3. MCNN

The parameter number of multiscale convolution is much smaller than conventional convolution. Figure 2 and figure 3 give the diagram of ordinary CNN and MCNN. Mixing up two different filters into a convolution operation could achieve multiscale feature extraction. Half of input channels convolve with one filter and the rest convolve with the other filter. Assuming that the kernel size is $3 \times 3$ in ordinary convolution, and the kernel size are $1 \times 1$ and $3 \times 3$, also these two operations have the same input channels and output channels. Neglecting the bias, the parameter number of figure 2 is $C_{in} \times C_{out} \times 3 \times 3$, and the parameter number of figure 3 can be calculated as $C_{in} / 2 \times 3 \times 3 + C_{in} / 2 \times 1 \times 1 + C_{in} \times C_{out}$. For example, $C_{in}$ and $C_{out}$ are set as 64, and then, the parameters of conventional convolution and multiscale convolution are 36864 and 4416 respectively. It shows that the parameters of multiscale convolution are less than $1/8$ of conventional convolution, which means speeding up the parameter training and optimization, lowering the computing cost.
In order to achieve multiscale feature extraction, we mix up $1 \times 1$ and $3 \times 3$ filters into a convolution operation. The total number of two filters equals to the input channels. In other word, each half of input channels convolve with one filter. Then, connecting two feature maps and convolving with another $1 \times 1$ filter—called MDSC operation as shown in figure 4. A MCB includes two MDSC operations and BN operation. Figure 5 shows the architecture of MCB.

Multiscale convolution is introduced into MCB so as to reduce computing cost and improve training efficiency. Moreover, it can alleviate overfitting to heighten transferability and generalization ability.

4. Proposed MCNN model

CNN is wildly used and get excellent performance in image filed. For HSI, MCNN can exploit the spectral-spatial information and preserve the original structure, so combining CNN with MCNN can truly improve the accuracy of HSI classification. There are some redundant bands which carry a small amount of information, but they will produce a large number of related parameters and consume resources in subsequent model training. Because of this, PCA is necessary to bring in. Using PCA to remove noises and retain useful information. Using CNN to extract features, and then, feeding feature maps into MCNN to extract features again. Increasing the convergence rate by adopting BN and Relu function after every convolution operation. To simplify the figure of proposed MCNN model, BN and Relu function are not marked in figure 6. The architecture of proposed MCNN is shown as figure 6.
The process of the algorithm is as follows. Firstly, adopting PCA to degrade the dimensions of original HSI, feeding the processed feature maps into the 2D-CNN for first extraction, after this, the characteristics of each feature map are fully expressed. Second, the data after BN operation is sent to double MCB to extract again and enhance the features. Then, using adaptive average pooling (AAP) to reshape the features for final classification. Finally, the test set is fed to trained MCNN to classify and calculate the classification accuracy.

5. Details of Experiment

5.1. Data set description
Adopting two data sets to evaluate the performance of the method in this paper, including the Pavia University (PU) data set and Salinas (SA) data set.

The PU image was captured by the ROSIS sensor over Pavia University in Italy in 2003. The PU image has 610×340 pixels and 115 bands ranging 400 to 860 nm. The spatial resolution is 1.3 m. Removing 12 bands because of existing noises, only 103 bands remained for the experiment. It contains 9 land-cover classes. Randomly selecting 2% samples in each class as train set, the rest of samples are as test set.

The SA image was captured by the AVIRIS sensor over the Salinas Valley in California in 1998. There are 512×217 pixels with the spatial resolution of 3.7 m. It originally consists of 224 bands, given 20 water absorption bands, only 204 bands are reserved for the experiment. SA image has 16 land-over classes and 2% samples in each class are selected as train set, the rest of samples are using for testing.

5.2. Experimental setup
The experimental environment is a laptop with Windows10 (x64), Intel Core i5-9300H CPU, Nvidia GeForce RTX 2060 GPU and 16GB RAM. Deep learning framework chooses Pytorch. The subsequent training and testing are based on this configuration.

The hyperparameters follow the hyperparameters performance well in pervious papers[12-14]. PCA is used for preprocessing. The number of principle components of PU image is 15, batch size is set to 120. The ratio of train set and test set is 2:98. The number of principle components of SA image is chosen 10, the others configuration is same as PU data set. According to the final classification accuracy, 0.1 is the best learning rate.

5.3. Experimental results and analysis
The overall accuracy (OA), average accuracy (AA) and kappa coefficient (Kappa) are adopted to evaluate the final classification performance. OA represents the classification ability of all classes. AA refers to the average classification accuracy of all classes, it reveals the classification ability of each class in a certain degree. Kappa is an index to measure the agreement of classification of all classes. The values of OA, AA and Kappa approach to 1 means the network performances well.

In order to find the best patch size of classification based on the network whose parameters have mentioned above, training the network by feeding the input of different sizes. Table 1 and table 2 shows the effect of patch size on the classification performance of PU and SA data set.

As the tables show, the performance (OA) increases at first and then slightly decreases as the patch size increases. When the patch size reaches 13×13, OA reaches the climax of classification accuracy of
PU data set. SA data set is classified best when patch size is $15 \times 15$. Patch size is set to $13 \times 13$ and $15 \times 15$ for the PU and SA datasets in the following experiments. Figure 7 and figure 8 are the classification maps with the proposed MCNN model of PU and SA datasets, respectively.

To fully demonstrate the performance of proposed MCNN, comparing it with the classification methods based on 2D-CNN[15], 3D-CNN[16], CD-CNN[17], DRN[18], DFFN[19]. Table 3 and table 4 are the results of comparison of PU and SA datasets.

### Table 1. Classification results of PU data set obtained by different patch sizes.

| Evaluation index | $7 \times 7$ | $9 \times 9$ | $11 \times 11$ | $13 \times 13$ | $15 \times 15$ | $17 \times 17$ | $19 \times 19$ |
|------------------|-------------|-------------|--------------|--------------|--------------|--------------|--------------|
| OA/%             | 98.52       | 98.98       | 99.17        | 99.33        | 99.10        | 99.05        | 98.52        |
| AA/%             | 97.80       | 98.48       | 98.55        | 98.83        | 98.27        | 98.21        | 97.25        |
| Kappa/%          | 98.03       | 98.65       | 98.89        | 99.12        | 98.80        | 98.75        | 98.04        |

### Table 2. Classification results of SA data set obtained by different patch sizes

| Evaluation index | $7 \times 7$ | $9 \times 9$ | $11 \times 11$ | $13 \times 13$ | $15 \times 15$ | $17 \times 17$ | $19 \times 19$ |
|------------------|-------------|-------------|--------------|--------------|--------------|--------------|--------------|

Figure 7. Classification map of PU data set. (a) Ground truth. (b) Proposed MCNN.

Figure 8. Classification map of SA data set. (a) Ground truth. (b) Proposed MCNN.
Table 3. classification results of PU data set obtained by different methods

| Evaluation index | 2D-CNN | 3D-CNN | CD-CNN | DRN | DFFN | MCNN |
|------------------|--------|--------|--------|-----|------|------|
| OA/\%            | 93.75  | 95.82  | 83.70  | 93.20| 87.64| 99.33|
| AA/\%            | 95.46  | 96.55  | 79.98  | 91.09| 82.22| 98.83|
| Kappa/\%         | 91.92  | 94.54  | 78.68  | 91.09| 83.83| 99.12|

Table 4. classification results of SA data set obtained by different methods

| Evaluation index | 2D-CNN | 3D-CNN | CD-CNN | DRN | DFFN | MCNN |
|------------------|--------|--------|--------|-----|------|------|
| OA/\%            | 91.93  | 94.55  | 87.17  | 95.40| 96.56| 99.78|
| AA/\%            | 94.59  | 96.24  | 90.61  | 97.16| 96.71| 99.57|
| Kappa/\%         | 91.07  | 93.94  | 85.76  | 94.89| 96.17| 99.75|

Compared with other methods, it can be observed that the proposed MCNN can reserve the original information of HSI quite well, extract features efficiently and classify correctly. Obtaining more than 99% accuracy in 2 data sets shows this model has generalization ability. In addition, the MCNN improves the classification accuracy of PU data set by 3.51% on the basis of traditional 3D-CNN classification accuracy, and improves the classification accuracy of SA data set to 99.78%, which verify MCNN has excellent performance in HSI classification field.

6. Conclusion

A novel HSI classification method based on MCNN is proposed to solve the problem of image features underutilization and the difficulty of HSI classification because of the Hughes phenomenon. Introducing MDSC by mixing up 1 × 1 and 3 × 3 filters to extract the spectral-spatial information of HSI. Experiment shows the proposed method could extract the spectral-spatial information efficiently, with high classification accuracy and small number of parameters. However, the method proposed in this paper still needs to be improved: it easy to overfitting because of less training samples, generalization ability remain to be improved. Subsequent studies can focus on expanding training sample capacity, reducing and optimizing network parameters, reducing computing complexity, and improving generalization ability.

Reference

[1] ZHU Shulong, ZHANG Zhanmu. The acquisition and analysis of remote sensing image[M]. Beijing: Science Press, 2000.

[2] ZHANG Xuan, HAO Minglei, XING Hongyan, er al. Research on de-speckling of SAR image. based on 3D matching block [J]. Journal Electronic Measurement and Instrumentation, 2017, 31(10):1621-1626.

[3] Zeng Z, Cumming I. Modified SPIHT Encoding for SAR Image Data[J]. IEEE Transactions on Geoence & Remote Sensing,2001, 39(3) : 546-522.

[4] FU Suining, LU Xiaoping, LU Yao. ROF Partial Differential Equations Suitable for SAR Images. De-noising[J]. Bulletin of Surveying and Mapping, 2018(11):16-19.

[5] XU Dongdong, CHENG Deqiang, CHEN Liangliang, et al. Hyperspectral image classification. based on hierarchical guidance filtering and nearest regularized subspace[J]. Acta Photonica Sinica, 2020, 49(4):197-209.

[6] XIE S, GIRSHICK R, DOLLAR P, et al. Aggregated residual transformations for deep neural. networks [C]. IEEE Conference on Computer Vision and Pattern Recognition. 2016.

[7] ZHANG Hui, DU Yu, NING Shurong. Pedestrian detection method based on Faster RCNN[J]. Transducer and Microsystem Technologies, 2019,38(2): 147-149,153.
[8] LI Hua, YANG Jianeng, LIU Feng, et al. Survey of Breast Cancer Histopathology Image Classification Based on Deep Learning [J]. Computer Engineering and Applications, 2020, 56(13):1-11.

[9] REN S., HE K., GIRSHICK R., et al. Faster R-CNN: towards real-time object detection with region proposal networks [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(6):1137-1149.

[10] NAN F., ZENG Q., XING Y., et al. Single image super-resolution reconstruction based on the ResNeXt network [J]. Multimedia Tools and Applications, 2020, 79(7): 34459-34470.

[11] HUGHES G F. On the mean accuracy of statistical pattern recognition [J]. IEEE Trans on. Inform Theory (IT), 1968, 14:55-63.

[12] ZHANG J., WEI F., FENG F., et al. Spatial-spectral feature refinement for hyperspectral image classification based on attention-dense 3D-2D-CNN [J]. Sensors (Basel), 2020, 20(18): 5191.

[13] Chen Y S, Jiang H L, Li C Y, et al. Deep feature extraction and classification of hyperspectral images based on convolutional neural networks [J]. IEEE Transactions on Geoscience & Remote Sensing, 2016, 54(10): 6232-6251.

[14] Hamida A B , Benoit A , Lambert P , et al. 3-D deep learning approach for remote sensing image classification [J]. IEEE Transactions on Geoscience & Remote Sensing, 2018, 56(8):4420-4434.

[15] LIU B, YU X, ZHANG P., et al. A semi-supervised convolutional neural network for hyperspectral image classification [J]. Remote Sensing Letters, 2017, 8(9): 839-848.

[16] LI Ying, ZHANG Haokui, SHEN Qiang. Spectral-spatial classification of hyperspectral imagery with 3D convolutional neural network[J]. Remote Sensing, 2017, 9(1):67.

[17] H. Lee, H. Kwon. Going Deeper With Contextual CNN for Hyperspectral Image Classification[J]. IEEE Transactions on Image Processing, 2017, 26(10).

[18] HE K M., ZHANG X Y., REN S Q., et al. Deep residual learning for image recognition [C]. IEEE Conference on Computer Vision and Pattern Recognition, 2016 : 770-778.

[19] W. Song, S. Li, L. Fang, and T. Lu, Hyperspectral image classification with deep feature fusion network. IEEE Transactions on Geoscience & Remote Sensing, 2018, 56(6), 3173-3184.