

SPECTRAL-SPATIAL MULTISCALE RESIDUAL NETWORK FOR HYPERSPECTRAL IMAGE CLASSIFICATION

Shi He¹, Haitao. Jing^{1*}, Huazhu Xue¹

¹ School of Surveying and Land Information Engineering, Henan Polytechnic University, 454003 Jiaozuo, China -
(heshi@hpu.edu.cn, jht_6153@163.com, xhz@hpu.edu.cn)

KEY WORDS: Hyperspectral Image (HSI), Classification, Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Multi-Task Learning, Residual Networks.

ABSTRACT:

In recent years, deep neural networks (DNN) are commonly adopted for hyperspectral image (HSI) classification. As the most representative supervised DNN model, convolutional neural networks (CNNs) have outperformed most algorithms. But the main problem of CNN-based methods lies in the over-smoothing phenomenon. Meanwhile, mainstream methods usually require a large number of samples and a large amount of computation. A multi-task learning spectral-spatial multiscale residual network (SSMRN) is proposed to learn features of objects effectively. In the implementation of the SSMRN, a multiscale residual convolutional neural network (MRCNN) is proposed as spatial feature extractors and a band grouping-based bi-directional gated recurrent unit (Bi-GRU) is utilized as spectral feature extractors. To evaluate the effectiveness of the SSMRN, extensive experiments are conducted on public benchmark data sets. The proposed method can retain the detailed boundary of different objects better and yield a competitive performance compared with two state-of-the-art methods especially when the training samples are inadequate.

1. INTRODUCTION

With the rapid development of remote sensing imaging spectroscopy technology, hyperspectral images (HSIs) have become increasingly important in Earth observation due to their rich spectral and spatial information. HSI classification is the task of identifying the category for each pixel with a proper land-cover label (Sun et al., 2019), which is more challenging because of the large dimensionality, spectral heterogeneity, and complex spatial distribution of the objects (Xu, Zhang, et al., 2018).

To alleviate these problems, traditional HSI classification methods follow a two-step approach: 1) Feature selection and extraction (Lefei Zhang et al., 2016), such as subspace projection (Harsanyi & Chang, 1994), random feature selection (Waske et al., 2010) and principal component analysis (PCA) (Rodarmel & Shan, 2002). 2) Classifier training, such as the k-nearest-neighbour (KNN) (Ma et al., 2010), Gaussian mixture model classifier (GMMC) (Li et al., 2013), support vector machine (SVM) (Melgani & Bruzzone, 2004), and random forest (RF) (Ham et al., 2005). But the traditional HSI classification methods are a separated process and mainly utilize hand-crafted features, which are not robust for different input data.

Deep neural networks (DNNs) can learn very complicated relationships between their inputs and outputs with multiple non-linear hidden layers (Liu et al., 2017). A large number of DNN-based methods are proposed for end-to-end modelling, which can integrate a spectral module and a spatial module (Liangpei Zhang et al., 2016). For example, Yang et al. designed a two-convolutional neural network (CNN) model to learn the spectral features and spatial features jointly (Yang et al., 2017). Spatial-spectral unified network (SSUN) combined a spectral dimensional band grouping-based long short-term memory (LSTM) model with 2D CNN for spatial features and integrated the spectral feature extraction (FE), spatial FE, and classifier training into a unified neural network (Xu, Zhang, et al., 2018). Zhong et al. raised an end-to-end 3D residual CNN architecture

for spectral-spatial feature learning and classification (Zhong et al., 2017). Motivated by the attention mechanism of the human visual system, a spectral-spatial attention network (SSAN) (Mei et al., 2019) and a residual spectral-spatial attention network (RSSAN) (Zhu et al., 2020) were proposed for hyperspectral image classification. To reduce computations, fully convolutional networks were proposed for HSI classification (Xu et al., 2019). For correctly discovering the contextual relations among pixels, the graph convolutional network (GCN) was adopted for dealing with the HSI classification, which was originally designed for arbitrarily structured non-Euclidean data (Wan et al., 2020). DNNs have demonstrated excellent performance in image classification (Druzhkov & Kustikova, 2016).

However, most existing deep learning methods are time-consuming in the training period. A lot of parameters need to be determined, and a large number of training samples are usually required for deep learning methods (Cheng et al., 2021; Xu, Du, et al., 2018). DNNs usually yield overfitting methods (Zhang et al., 2018), and are sensitive to perturbations (Xu et al., 2020), especially when the training samples are inadequate. Exploring the proper depth of a DNN model for a given data set is still an open topic to be researched (Liangpei Zhang et al., 2016).

2. PRELIMINARY

2.1 CNN

A CNN (Shin et al., 2016) is a class of deep neural networks, most commonly applied to analysing visual imagery. Compared with other neural networks, CNNs are easier to train with their fewer parameters and connections because of parameter sharing and local connectivity. But CNN extracts the spatial structure of the current pixel neighbouring region, instead of directly for the current pixel. So, the main problem of CNN-based methods lies in the over-smoothing phenomenon. One approach to solve this

* Corresponding author

problem is to utilize superpixels segmentation, but the segmentation algorithm affects the classification results. Another approach is to use an attention mechanism. Attentional mechanisms can counteract the effects of parameter sharing, but increase the amount of computation.

2.2 RNN

To utilize the features of the current hyperspectral pixel itself, the RNN network structure can be used. An RNN extends conventional feedforward neural networks with loops in connections that can use their internal memory to exhibit temporal dynamic behaviour. RNN makes them applicable to challenging tasks involving sequential data such as speech recognition and language modelling(Cho et al., 2014). A natural idea is to consider each band as a time step, but hyperspectral data usually have hundreds of bands, which consumes high computing and storage resources. Meanwhile, a large number of spectral channels and limited training samples restrict the performance of hyperspectral image classification(Cheng et al., 2021).

3. PROPOSED FRAMEWORK

Three subsections are playing crucial roles in our methodology: a multiscale residual CNN-based spatial feature learner, a bi-directional GRU-based spectral feature learner, and a multi-task learning model.

3.1 Multiscale Residual CNN for Spatial Classification

Considering the complex environment of the HSI, we propose to extract robust and multiscale spatial features. The proposed multiscale residual CNN(MRCNN) architecture is shown in Figure 1.

Let $X \in \mathbb{R}^{r \times c \times b}$ be the original HSI data, where r , c and b are the row number, column number, and band number, respectively.

First of all, to suppress noise and reduce the computational costs, the PCA is applied to the original HSI data and only the first p principle components are reserved. Denote the dimension-reduced data by $X_p \in \mathbb{R}^{r \times c \times p}$. Around each pixel, a neighbour region is extracted with the size of $k \times k \times p$ as the input of the spatial branch.

Considering the complex environment of the HSI, where different objects tend to have different scales, we propose to extract both shallow and deep features by applying a convolution layer with ReLU activation and two residual blocks in the classification. The max pooling layer is adopted in residual blocks. We add a flatten layer with dropout technology and a fully-connected(FC) layer after the convolution layer with ReLU activation and two residual blocks. Then, these fully-connected layers are merged into a new fully-connected layer. Let $h^j = f(W^j x^j + b^j)$, $j = 1, 2, 3$ denotes the j th fully-connected layer, where x^j is the flattened features in the j th flatten layer, W^j and b^j are the corresponding weight matrix and bias term, respectively. The fourth fully-connected layer h^4 can be calculated as $h^4 = \text{concat}[h^1, h^2, h^3]$. In this way, features in different layers are taken into consideration during the classification stage, and the network will possess the property of multiscale.

The loss function for cross-entropy of MRCNN can be expressed by

$$\mathcal{L} = -\frac{1}{M} \sum_{m=1}^M \sum_{n=1}^N y_m^n \log(\hat{y}_m^n), \quad (1)$$

where y_m^n and \hat{y}_m^n denote the truth and predicted labels, respectively. M is the number of training samples and N is the number of classes.

All convolutional layers have 32 filters. The kernel size of the first left convolutional layer is 1×1 , and the other kernel sizes are 3×3 . The size of the max pooling layers is 2×2 . The first three fully-connected layers each own 128 units, and the fourth fully-connected owns 384 units.

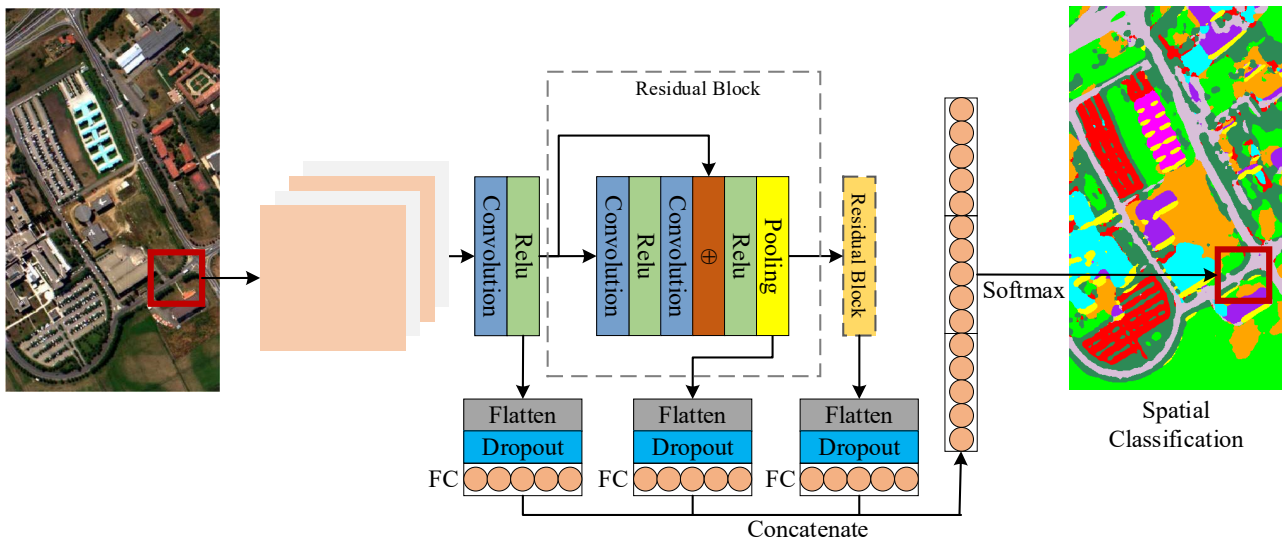


Figure 1. Architecture of the proposed MRCNN.

3.2 Bi-GRU for Spectral Classification

The GRU has fewer parameters than LSTM for modelling various sequential problems, and bi-directional RNN can make full use of both latter and previous information. Therefore, we

utilize Bi-GRU for spectral classification. The complete spectral classification framework is illustrated in Figure 2.

A natural idea is to consider each band as a time step in Bi-GRU, but hyperspectral data usually have hundreds of bands, which consumes high computing and storage resources. Thus, a suitable grouping strategy(Xu, Zhang, et al., 2018) is used in this paper. For each pixel in the HSI, let $x = [\lambda_1, \lambda_2, \dots, \lambda_j, \dots, \lambda_b]$ be the

spectral vector, where λ_j is the reflectance of the j th band and b is the number of bands. Let $r (\ll b)$ be the number of time steps (e.g., number of groups). The transformed sequences can be denoted by $c = [c_1, c_2, \dots, c_t, \dots, c_r]$, where c_t is the sequence at the t th time step. Specifically, grouping strategy is

$$\begin{aligned} c_1 &= [\lambda_1, \lambda_{1+r}, \dots, \lambda_{1+(m-1)r}] \\ c_2 &= [\lambda_2, \lambda_{2+r}, \dots, \lambda_{2+(m-1)r}] \\ &\dots \\ c_t &= [\lambda_t, \lambda_{t+r}, \dots, \lambda_{t+(m-1)r}] \\ &\dots \\ c_r &= [\lambda_r, \lambda_{r+r}, \dots, \lambda_{r+(m-1)r}], \end{aligned} \quad (2)$$

where $m = \text{floor}(b/r)$ is the sequence length of each time step and $\text{floor}(\cdot)$ function rounds numbers down. After grouping, spectral vector x is transformed into sequences $[c_1, c_2, \dots, c_t, \dots, c_r]$.

The input to our model is the sequential vector $c = [c_1, c_2, \dots, c_t, \dots, c_r]$, and the bi-directional hidden vector is calculated as:

Forward hidden state:

$$\vec{h}_t = f(\vec{W}c_t + \vec{V}h_{t-1} + \vec{b}), \quad (3)$$

Backward hidden state:

$$\overleftarrow{h}_t = f(\overleftarrow{W}c_t + \overleftarrow{V}h_{t+1} + \overleftarrow{b}), \quad (4)$$

where the coefficient matrices \vec{W} and \overleftarrow{W} are from the input at the present step, \vec{V} is from the hidden state h_{t-1} at the previous step, \overleftarrow{V} is from h_{t+1} at the succeeding step, f is the nonlinear activation of the hidden layer, and the memory of the input as the output of this encoder is g_t :

$$g_t = \text{concat}[\vec{h}_t, \overleftarrow{h}_t], \quad (5)$$

where $\text{concat}(\cdot)$ is a function of concatenation between the forward hidden state and backward hidden state.

Bi-GRU allows the sequential vector to be fed into the architecture one by one to learn continuous features with forward and backward directions. So, we can compute the predicted label y_i of pixel x_i as follows:

$$y_i = U([g_1, \dots, g_t, \dots, g_r]), \quad (6)$$

where $U(\cdot)$ is a flatten function with dropout technology. A fully-connected layer and a softmax activation function are added to accomplish the image classification.

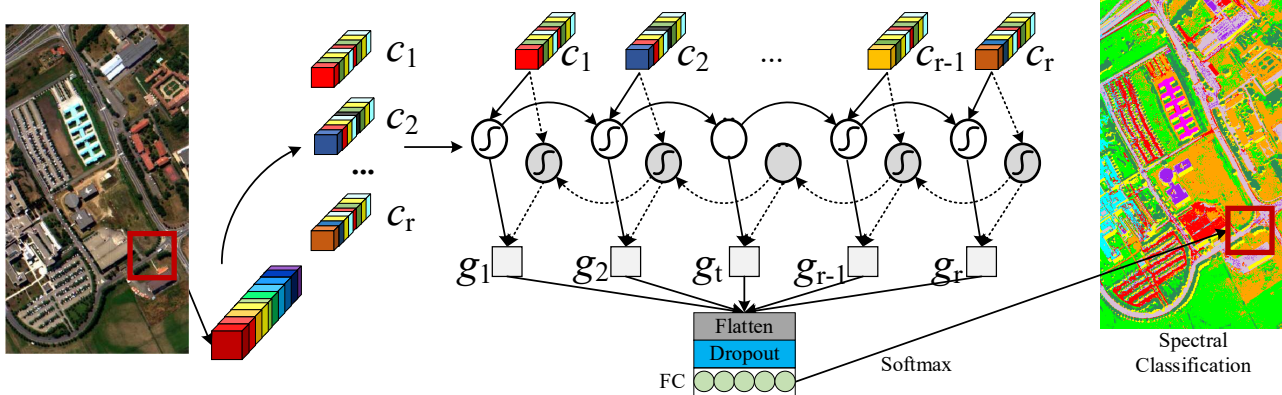


Figure 2. Architecture of the proposed band grouping-based Bi-GRU.

3.3 SSMRN

The proposed SSMRN framework is shown in Figure 3, which integrates the spectral FE, spatial FE, and classifier training into a unified neural network. The network adopts the proposed MRCNN to extract spatial features and the band grouping-based Bi-GRU algorithm to extract spectral features. The last fully-connected layer in Bi-GRU and the last one in MRCNN are concatenated to form a new fully-connected layer for the spectral-spatial classification. All parameters in the framework are trained at the same time.

To better train the whole network, two auxiliary tasks are added in the framework (Xu, Zhang, et al., 2018). So the proposed SSMRN is a triple-task framework, including one main task (classification based on spectral-spatial information) and two auxiliary tasks (classification based on spectral information and classification based on spatial information). The complete loss function for cross-entropy of the SSMRN is defined as

$$\begin{aligned} \mathcal{L} &= \mathcal{L}^{joint} + \mathcal{L}^{spectral} + \mathcal{L}^{spatial} = \\ &= -\frac{1}{M} \sum_{m=1}^M \sum_{n=1}^N y_m^n \log(\hat{y}_m^{joint}) - \\ &= \frac{1}{M} \sum_{m=1}^M \sum_{n=1}^N y_m^n \log(\hat{y}_m^{spectral}) - \\ &= \frac{1}{M} \sum_{m=1}^M \sum_{n=1}^N y_m^n \log(\hat{y}_m^{spatial}), \end{aligned} \quad (8)$$

where \mathcal{L}^{joint} is the main loss function, $\mathcal{L}^{spectral}$ and $\mathcal{L}^{spatial}$ are two auxiliary loss functions, \hat{y}_m^{joint} , $\hat{y}_m^{spectral}$, and $\hat{y}_m^{spatial}$ are the corresponding predicted labels, y_m^n is the true label. M is the number of training samples and N is the number of classes. The whole network is trained in an end-to-end manner, where all the parameters are optimized by the mini-batch or batch stochastic gradient descent algorithm at the same time. In this way, the complete loss function will balance the convergences of both the whole network and the subnetworks.

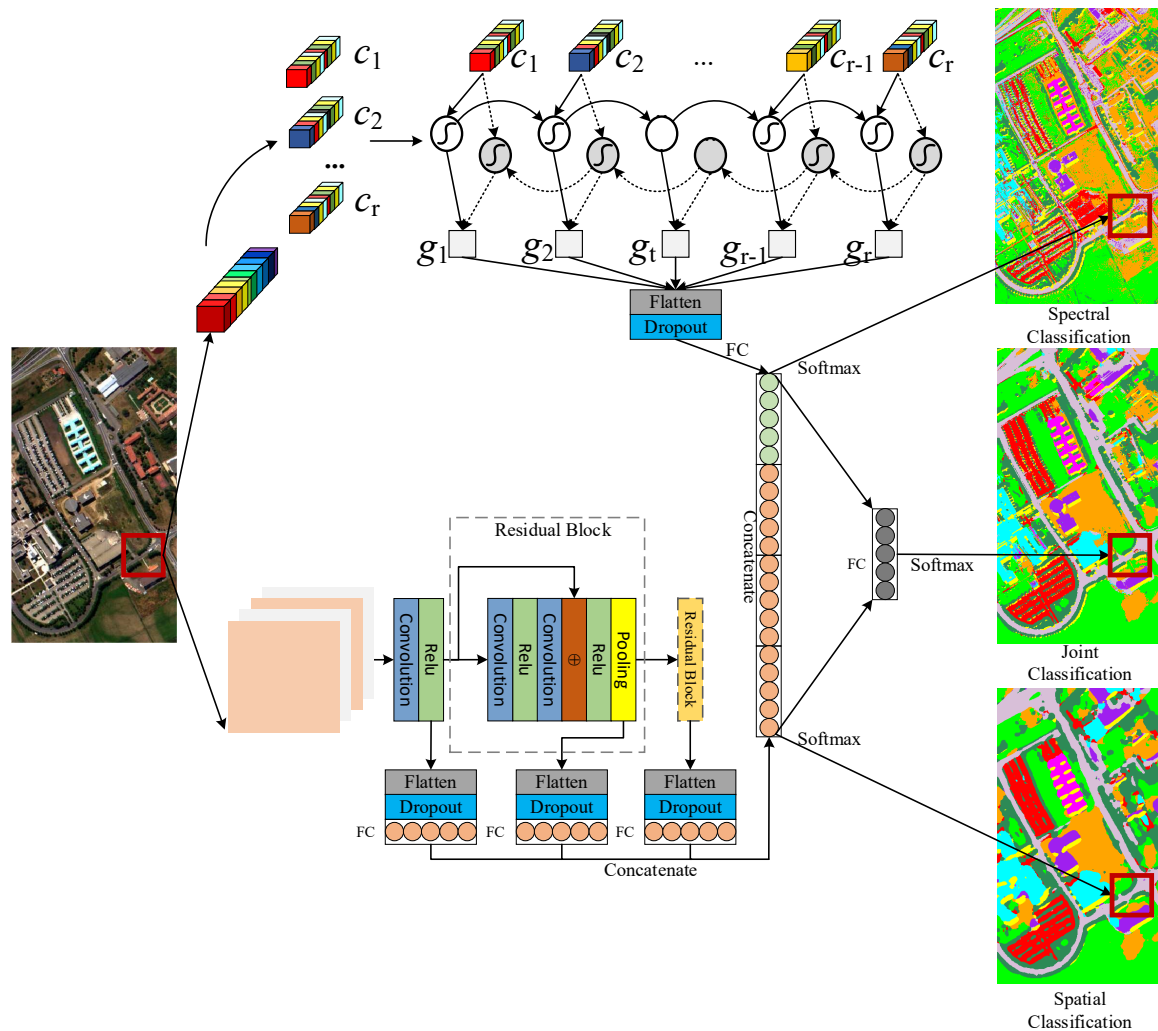


Figure 3. Architecture of the proposed SSMRN.

4. EXPERIMENTAL RESULTS

In this section, we introduce the public data sets used in our experiment and the configuration of the proposed SSMRN. In addition, classification performance based on the proposed method and other comparative methods is presented. All the experiments are implemented with an Intel(R) Xeon(R) Silver 4210 CPU @ 2.20-GHz with 64 GB of RAM and an NVIDIA RTX2080 graphic card, TensorFlow 2.3.1, and Keras 2.4.3 with python 3.7.6.

4.1 Experimental Data

Pavia University hyperspectral data set is utilized to evaluate the performance of the proposed method in the experiment. The data set is acquired by the Reflective Optics Systems Imaging Spectrometer (ROSIS) sensor during a flight campaign over Pavia, northern Italy. This image consists of 103 spectral bands with 610×340 pixels and it has a spectral coverage from 0.43 to $0.86 \cdot 10^{-6}$ meters. The spatial resolution is 1.3 m/pixel. Image ground truths differentiate 9 classes each.

All the experiments in this paper are randomly repeated 30 times with random training and test data. In each repetition, we first randomly generate the training set from the whole data set with the same number of the labelled class. Then, the remaining samples make up the test set. Details are listed in Table 1.

4.2 Parameter Setting

Let 3 be the number of time steps in the spectral branch. The input of the spatial branch is a $24 \times 24 \times 4$ patch, where 4 is the number of reserved principal components. The dropout rate is set as 60%. The number of neurons of the fully-connected layer in the spectral branch and the fully-connected layer corresponding to different scales in the spatial branch is set as 128, so the number of neurons in the joint fully-connected layer is 512. We use the Adam optimizer to train the networks with a learning rate of 0.001. The training epochs are set as 1000 with batch size 1024.

4.3 Classification Results

To demonstrate the superiority and effectiveness of the proposed SSMRN model, it is compared with advanced machine-learning methods such as SSAN(Mei et al., 2019), and RSSAN(Zhu et al., 2020). Limited by our computer configuration, we cannot run RSSAN properly with the original input, so the input of RSSAN is a $24 \times 24 \times 8$ patch, where 8 is the number of reserved principal components instead of the number of spectral bands. All algorithms are executed 30 times. The average results which add the standard deviation obtained from the 30 runs are reported to reduce random selection effects. Class-specific accuracy, overall accuracy (OA), and kappa coefficient are used as the evaluation measurements for the compared methods.

4.3.1 Qualitative Evaluation: The classification maps of different methods are shown in Figure 4. These methods make full use of the continuity of the ground object and yield a cleaner classification map. The main problem of CNN-based methods lies in the over-smoothing phenomenon, which can be observed in the results of the SSRN, RSSAN. Meanwhile, SSMRN can better retain the detailed boundary of different objects and acquire more clear results.

4.3.2 Quantitative Evaluation: The classification results of the different methods are shown in Table 2. All results in the table are obtained by the average value of running 30 times. The SSMRN surpasses other methods and owns the highest accuracy in all classes. SSMRN also owns the highest class-specific accuracy, OA, kappa, and lowest standard deviation. As the number of labelled training pixels can affect the OA of classification results, it is worth exploring how the performance behaves with settings on the different number of labelled pixels. To be specific, different training sample numbers of each class are varied 30,50,100,200 for Pavia University. As shown in Figure 5, it can be found that the proposed SSMRN generally outperforms other methods. Our method shows more robustness even when the training samples are inadequate.

Class Name	Training	Test
Asphalt	200	6431
Meadows	200	18449
Gravel	200	1899
Trees	200	2864
Painted metal sheets	200	1145
Bare Soil	200	4829
Bitumen	200	1130
Self-Blocking Bricks	200	3482
Shadows	200	747

Table 1. Number of training and test samples used in the Pavia University data set

Class Name	SSAN	RSSAN	SSMRN
Asphalt	96.86	98.68	99.48
Meadows	96.53	99.52	99.61
Gravel	98.64	99.31	99.69
Trees	98.28	98.63	99.80
Painted metal sheets	99.97	99.82	99.99
Bare Soil	99.63	99.87	99.89
Bitumen	99.77	99.74	99.96
Self-Blocking Bricks	98.61	98.47	99.69
Shadows	99.92	99.38	99.99
OA(%)	97.59	99.28	99.67
	±1.20	±0.37	±0.24
Kappa	96.79	99.03	99.56
×100	±1.59	±0.50	±0.32
Runtime(s)	321.25	335.54	207.05
	±2.00	±3.94	±2.59

Table 2. Classification results of different methods for the Pavia University data set. Bold indicates the best result.

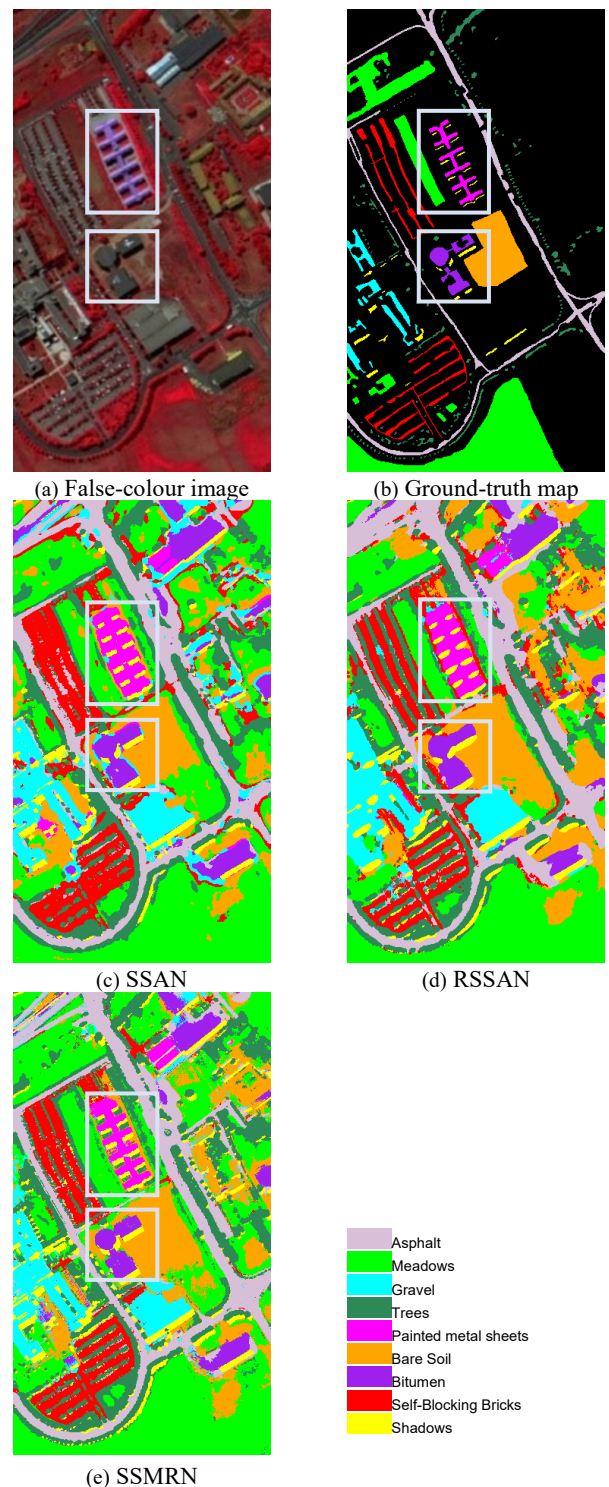


Figure 4. Classification maps using different methods.

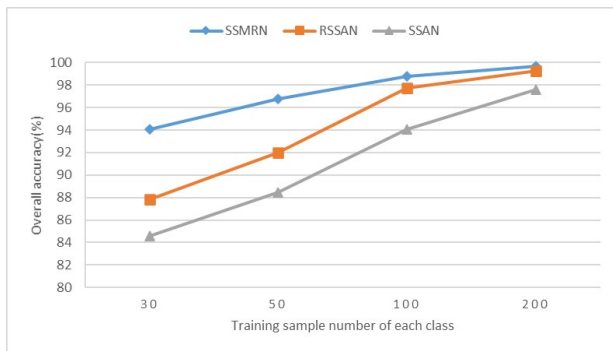


Figure 5. Overall accuracies of different methods under different training sample numbers of each class on the Pavia University data set.

5. CONCLUSIONS

To effectively learn features of objects and significantly reduce networks complexity, a multi-task learning spectral-spatial multiscale residual network (SSMRN) has been proposed to extract spectral-spatial features. The end-to-end networks based on MRCNN and Bi-GRU can learn higher-level spectral-spatial joint features. The experimental results demonstrate that the method not only has a better performance compared with the other methods in terms of class-specific accuracy, OA, and kappa, but also mitigates the over-smoothing phenomenon. Our method shows more robustness even when the training samples are inadequate.

Although we utilize the proposed band MRCNN and Bi-GRU as the spatial and spectral feature extractors in the implementation of the proposed SSMRN, other deep networks can also be introduced into our model, especially for spectral extractors. It deserves to be investigated in future work.

6. ACKNOWLEDGEMENTS

This work was supported by the Doctoral Foundation of Henan Polytechnic University under Grant B2017-09, the Major Science and Technology Program of Henan Province (No. 222102320341), the Key Scientific Research Project in Colleges and Universities of Henan Province (No. 22B420004), and the Fundamental Research Funds for the Universities of Henan Province (No. NSFRF210401).

REFERENCES

Cheng, C., Li, H., Peng, J., Cui, W., & Zhang, L. 2021. Hyperspectral Image Classification Via Spectral-Spatial Random Patches Network. *IEEE journal of selected topics in applied earth observations and remote sensing*, 14, 4753-4764.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.

Druzhkov, P., & Kustikova, V. 2016. A survey of deep learning methods and software tools for image classification and object detection. *Pattern Recognition and Image Analysis*, 26(1), 9-15.

Ham, J., Chen, Y., Crawford, M. M., & Ghosh, J. 2005. Investigation of the random forest framework for classification of hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, 43(3), 492-501.

Harsanyi, J. C., & Chang, C.-I. 1994. Hyperspectral image classification and dimensionality reduction: An orthogonal subspace projection approach. *IEEE Transactions on Geoscience and Remote Sensing*, 32(4), 779-785.

Li, W., Prasad, S., & Fowler, J. E. 2013. Hyperspectral image classification using Gaussian mixture models and Markov random fields. *IEEE Geoscience and Remote Sensing Letters*, 11(1), 153-157.

Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., & Alsaadi, F. E. 2017. A survey of deep neural network architectures and their applications. *Neurocomputing*, 234, 11-26.

Ma, L., Crawford, M. M., & Tian, J. 2010. Local manifold learning-based k -nearest-neighbor for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 48(11), 4099-4109.

Mei, X., Pan, E., Ma, Y., Dai, X., Huang, J., Fan, F., . . . Ma, J. 2019. Spectral-spatial attention networks for hyperspectral image classification. *Remote Sensing*, 11(8), 963.

Melgani, F., & Bruzzone, L. 2004. Classification of hyperspectral remote sensing images with support vector machines. *IEEE Transactions on Geoscience and Remote Sensing*, 42(8), 1778-1790.

Rodarmel, C., & Shan, J. 2002. Principal component analysis for hyperspectral image classification. *Surveying and Land Information Science*, 62(2), 115-122.

Shin, H.-C., Roth, H. R., Gao, M., Lu, L., Xu, Z., Nogues, I., . . . Summers, R. M. 2016. Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. *IEEE transactions on medical imaging*, 35(5), 1285-1298.

Sun, H., Zheng, X., Lu, X., & Wu, S. 2019. Spectral-spatial attention network for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 58(5), 3232-3245.

Wan, S., Gong, C., Zhong, P., Pan, S., Li, G., & Yang, J. 2020. Hyperspectral image classification with context-aware dynamic graph convolutional network. *IEEE Transactions on Geoscience and Remote Sensing*, 59(1), 597-612.

Waske, B., van der Linden, S., Benediktsson, J. A., Rabe, A., & Hostert, P. 2010. Sensitivity of support vector machines to random feature selection in classification of hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, 48(7), 2880-2889.

Xu, Y., Du, B., Zhang, F., & Zhang, L. 2018. Hyperspectral image classification via a random patches network. *ISPRS journal of photogrammetry and remote sensing*, 142, 344-357.

Xu, Y., Du, B., & Zhang, L. 2019. Beyond the patchwise classification: Spectral-spatial fully convolutional networks for hyperspectral image classification. *IEEE Transactions on Big Data*, 6(3), 492-506.

Xu, Y., Du, B., & Zhang, L. 2020. Assessing the threat of adversarial examples on deep neural networks for remote sensing scene classification: Attacks and defenses. *IEEE Transactions on Geoscience and Remote Sensing*, 59(2), 1604-1617.

Xu, Y., Zhang, L., Du, B., & Zhang, F. 2018. Spectral–spatial unified networks for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 56(10), 5893-5909.

Yang, J., Zhao, Y.-Q., & Chan, J. C.-W. 2017. Learning and transferring deep joint spectral–spatial features for hyperspectral classification. *IEEE Transactions on Geoscience and Remote Sensing*, 55(8), 4729-4742.

Zhang, C., Vinyals, O., Munos, R., & Bengio, S. 2018. A study on overfitting in deep reinforcement learning. arXiv preprint arXiv:1804.06893.

Zhang, L., Zhang, L., & Du, B. 2016. Deep learning for remote sensing data: A technical tutorial on the state of the art. *IEEE Geoscience and Remote Sensing Magazine*, 4(2), 22-40.

Zhang, L., Zhang, Q., Du, B., Huang, X., Tang, Y. Y., & Tao, D. 2016. Simultaneous spectral-spatial feature selection and extraction for hyperspectral images. *IEEE Transactions on Cybernetics*, 48(1), 16-28.

Zhong, Z., Li, J., Luo, Z., & Chapman, M. 2017. Spectral–spatial residual network for hyperspectral image classification: A 3-D deep learning framework. *IEEE Transactions on Geoscience and Remote Sensing*, 56(2), 847-858.

Zhu, M., Jiao, L., Liu, F., Yang, S., & Wang, J. 2020. Residual spectral–spatial attention network for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 59(1), 449-462.