A JOINT SPATIAL-SPECTRAL REPRESENTATION BASED CAPSULE NETWORK FOR HYPERSPECTRAL IMAGE CLASSIFICATION

Qun Zhou, Wenxing Bao, Xiaowu Zhang, Xuan Ma*

School of Computer Science and Engineering North Minzu University, Yinchuan 750021, China
*Corresponding author: bwx71@163.com

ABSTRACT

Recently, capsule networks have shown excellent performance in hyperspectral image (HSI) classification. However, when the HSI contains complex topographic features, the classification performance of capsule networks will deteriorate. To address this issue, we propose a novel joint spatial-spectral representation based capsule network (JSSR-CapsNet), which deeply exploits spatial-spectral information to improve classification performance. Specifically, JSSR-CapsNet utilizes an attention module to highlight the validity of sensitive pixels from redundant spatial-spectral features. Then, we built a dilated convolutional pyramid (DCP) to take aggregation of pixels with the same class into account, and suppress the interference of noisy pixels to enhance the intra-class consistency. The quantitative results conducted on two real HSIs show that JSSR-CapsNet improves the overall accuracy by 1.86% and 2.07% respectively over state-of-the-art capsule networks.

Index Terms— Hyperspectral image classification, capsule network, joint spatial-spectral representation, dilated convolutional pyramid

1. INTRODUCTION

Deep learning-based methods have achieved promising performance in hyperspectral image (HSI) classification [1], where a convolutional neural network (CNN) has attracted wide concern [2]. However, CNN has the following shortcomings: 1) CNN lacks spatial generalization ability with limited labeled samples, which means that it needs a large number of training samples, and yet the HSI normally has insufficient labeled data; 2) In the pooling layer of CNN, a lot of spatial information of HSI will be lost. In 2017, Sabour *et al.* [3] proposed a novel capsule network (CapsNet), which employs a vector output from the capsule instead of the scalar output of CNN, that improves the drawback of CNN in the limited samples scenario. In addition, CapsNet adopts a dynamic routing strategy to overcome the defects of pooling layers of CNN. Subsequently, Arun *et al.* [4] proposed a classification model for extracting spectral features using CapsNet, which addresses translational invariance. Paoletti *et al.* [5] developed a hyperspectral classification architecture based on spectral-spatial CapsNet, which has low computational complexity. Zhu *et al*. [6] designed a convolutional capsule network (Conv-CapsNet) that uses local connections and shared transformation matrices, containing less trainable parameters to alleviate the overfitting. Although the above classification model can obtain good classification results, in the neighborhood with complex pixel distribution, the vector feature representation will be affected by noise pixels, resulting in a decrease in classification accuracy.

In this paper, we propose a novel joint spatial-spectral representation based capsule network framework for hyperspectral classification, called JSSR-CapsNet. First, JSSR-CapsNet utilizes an attention module to extract abundant spatial-spectral features. Then, JSSR-CapsNet introduces a dilated convolutional pyramid (DCP) module to aggregate features from different scales, which alleviates noise interference and enhances the intra-class consistency. The paper is structured as follows. Section 2 introduces the JSSR-CapsNet model. Section 3 validates the classification performance of JSSR-CapsNet. Section 4 concludes the paper with some remarks.

2. METHOD

2.1. Joint spatial-spectral attention module

The attention mechanism has been widely used in various types of deep learning tasks such as speech recognition [7], image classification [8], and natural language processing [9], it can quickly screen high utility information in limited global receptive fields [10]. For HSI classification, it can effectively extract spatial-spectral information [11], and highlights the role of effective features in homogeneous areas, and suppresses irrelevant features.

In this paper, we develop a joint spatial-spectral attention (JSSA) module, which is constructed by a feature fusion strategy based on branch networks. As shown in **Figure 1**, a spatial feature extractor takes the data cube $\boldsymbol{F} \in R^{s\times s\times c}$ as input, where s denotes spatial size of neighborhood block and c denotes the number of spectral bands. For the input dataset, it is performed by average pooling and maximum pooling operations along spectral

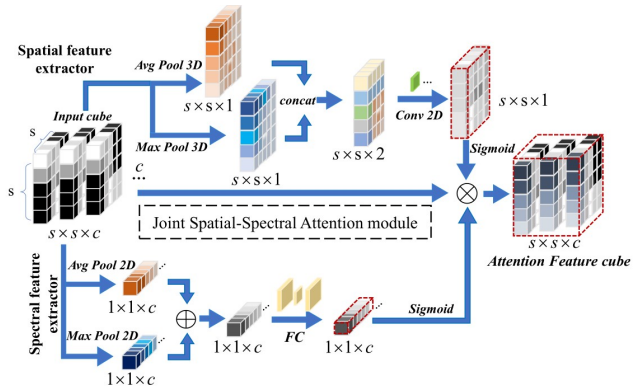

Figure 1. The JSSA module.

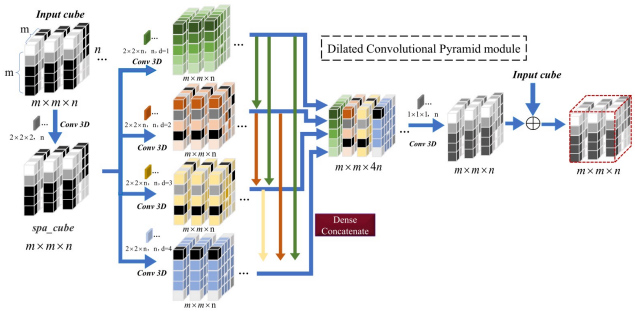

Figure 2. The DCP module.

dimension to obtain their corresponding features. Then, a 'concatenation' strategy is used to construct feature maps. Finally, it learns a set of spatial filters with *sigmoid* activation function to get spatial attention weights $\boldsymbol{P}_{spa}$. For the spectral feature extractor, it first explores spectral feature along the spatial dimension with pooling operations, and generates feature maps with 'Addition' strategy. Then, the feature maps are input into a fully connected layer with the *sigmoid* activation function to produce a weight vector of spectral attention $\boldsymbol{P}_{spc}$. The outputted cube $\boldsymbol{G} \in R^{s\times s\times c}$ is obtained by element-wise multiplication of $\boldsymbol{P}_{spa}$, $\boldsymbol{P}_{spc}$ and $\boldsymbol{F}$, which is given by:

$$\boldsymbol{G}(\boldsymbol{F}) = \boldsymbol{P}_{spa} \times \boldsymbol{P}_{spc} \times \boldsymbol{F}, \quad (1)$$

$$\boldsymbol{P}_{spa} = \sigma\left(\boldsymbol{conv}\left(\boldsymbol{concat}\left(\boldsymbol{avg}(\boldsymbol{F}); \boldsymbol{max}(\boldsymbol{F})\right)\right)\right), \quad (2)$$

$$\boldsymbol{P}_{spc} = \sigma\left(\boldsymbol{FC}\left(\boldsymbol{add}\left(\boldsymbol{avg}(\boldsymbol{F}); \boldsymbol{max}(\boldsymbol{F})\right)\right)\right). \quad (3)$$

where σ, ***conv***(.) and ***concat***(.) denote respectively the *sigmoid* activation function, the convolution operation and the concatenate operation. ***Avg***(.) and ***max***(.) represent the average pooling operation and the maximum pooling operation respectively. ***FC***(.) and ***add***(.) denote the fully connected operation and the add operation respectively.

2.2. DCP Module

The dilated convolution generates HSI spatial structure information of different scales by adding expansion factors to capture a wider range of spatial features [12, 13]. However, such operation considers discrete sampling method, resulting in insufficient utilization of spatial information [14].

In this paper, we develop a DCP module [see **Figure 2**] to extract multi-scale spatial features. The module can suppress the interference of noisy pixels and improve the effective representation of the vector features in the capsule layer, thereby enhancing the intra-class consistency. First, the module inputs a feature cube $\boldsymbol{I} \in \mathbb{R}^{m\times m\times n}$ (where m represents the spatial size of neighborhood block and n represents the number of spectral bands) to perform convolution operations to obtain a feature tensor ***spa_cube***. Then, for the feature tensor ***spa_cube***, four different dilated factors d = {1, 2, 3, 4} are used to execute convolution operations to get multi-scale spatial feature maps $\boldsymbol{T}_1$, $\boldsymbol{T}_2$, and $\boldsymbol{T}_3$, given by:

$$\boldsymbol{T}_1 = \boldsymbol{add}\left(\boldsymbol{d\_conv1}(\boldsymbol{I}); \boldsymbol{d\_conv2}(\boldsymbol{I})\right), \quad (4)$$

$$\boldsymbol{T}_2 = \boldsymbol{add}\left(\boldsymbol{T}_1; \boldsymbol{d\_conv3}(\boldsymbol{I})\right), \quad (5)$$

$$\boldsymbol{T}_3 = \boldsymbol{add}\left(\boldsymbol{T}_2; \boldsymbol{d\_conv4}(\boldsymbol{I})\right), \quad (6)$$

$$\boldsymbol{K} = \boldsymbol{concat}\left(\boldsymbol{I}; \boldsymbol{T}_1; \boldsymbol{T}_2; \boldsymbol{T}_3\right). \quad (7)$$

where ***add***(.), ***d_conv***(.) and ***concat***(.) represent the add operation, the dilated convolution operation and the concatenate operation, respectively. Finally, a hierarchical feature fusion strategy is used to merge multi-features to generate $\boldsymbol{K}$.

2.3. JSSR-CapsNet Model

Figure 3 is the classification flowchart of JSSR-CapsNet. Suppose a data cube $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\} \in R^{w\times w\times h}$ is divided into three subsets: training dataset $\boldsymbol{X}^1$, validation dataset $\boldsymbol{X}^2$ and testing dataset $\boldsymbol{X}^3$. $\boldsymbol{Y}^1$ and $\boldsymbol{Y}^2$ are the label vector sets corresponding to $\boldsymbol{X}^1$ and $\boldsymbol{X}^2$, respectively. Noting that w denotes spatial size of neighborhood block and h denotes the number of spectral bands. Finally, $\boldsymbol{X}^3$ is the input of well-trained model JSSR-CapsNet which is utilized to predict classification results $\hat{\boldsymbol{Y}}$.

Figure 4 shows the network structure of JSSR-CapsNet. Given the Indian Pines dataset, it firstly learns a set of spatial-spectral filters that put 27×27×200 cube into two 2-D convolutional layers to extract high-order features. Then the JSSA module is used to capture sensitive features of spatial-spectral information and send sensitive features into the capsule network. JSSR-CapsNet has three capsule layers, i.e., ***PrimaryCaps***, ***Conv_Caps*** and ***Class_Caps***. The DCP is embedded between the ***PrimaryCaps*** layer and the

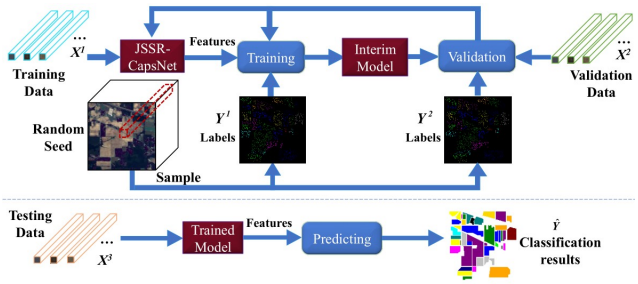

Figure 3. The classification flowchart of the JSSR-CapsNet.

Conv_Caps layer to aggregate the feature of pixels with the same class. After applying three capsule layers, JSSR-CapsNet gets 16 capsules, each of which represents a HSI land class. The length of the vector output of each capsule represents the probability that the input target pixel belongs to each class. $\|L_2\|$ is the Euclidean norm of a vector.

In the ***Conv_Caps*** layer, an iterative dynamic routing strategy is used to calculate the output of the capsule. First, the output of the capsule in the ***PrimaryCaps*** layer is multiplied by the transformation matrix $\boldsymbol{w}_{ij}^{pq}$ to obtain a prediction vector $\boldsymbol{u}_{j|i}^{(x+p)(y+q)}$. Then, the weighted sum $\boldsymbol{s}_j^{xy}$ of all prediction vectors is used as the input of the capsule in the ***Conv_Caps*** layer. Finally, a *squash* activation function is applied to the input vector to generate the output $\boldsymbol{v}_j^{xy}$ of the capsule, given as:

$$\boldsymbol{u}_{j|i}^{(x+p)(y+q)} = \boldsymbol{w}_{ij}^{pq}\boldsymbol{u}_i^{(x+p)(y+q)}, \tag{8}$$

$$\boldsymbol{s}_j^{xy} = \sum_{i=1}^{I}\sum_{p=0}^{P-1}\sum_{q=0}^{Q-1}\boldsymbol{c}_{ij}^{pq}\boldsymbol{u}_{j|i}^{(x+p)(y+q)}, \tag{9}$$

$$\boldsymbol{v}_j^{xy} = \boldsymbol{squash}(\boldsymbol{s}_j^{xy}) = \frac{\| \boldsymbol{s}_j^{xy} \|^2}{1+\| \boldsymbol{s}_j^{xy} \|^2}\frac{\boldsymbol{s}_j^{xy}}{\| \boldsymbol{s}_j^{xy} \|}, \tag{10}$$

where $\boldsymbol{u}_i^{(x+p)(y+q)}$ is the output of i-th channel's capsule in the ***PrimaryCaps*** layer at position $(x+p, y+q)$. I denote the number of capsule channels in the ***PrimaryCaps*** layer, P and Q are the kernel size. And $\boldsymbol{c}_{ij}^{pq}$ is a coupling coefficient derived from dynamic routing strategy.

In the dynamic routing strategy, $\boldsymbol{c}_{ij}^{pq}$ is obtained by applying a *softmax* activation function on $\boldsymbol{b}_{ij}^{pq}$, given as:

$$\boldsymbol{c}_{ij}^{pq} = \frac{\exp\left(\boldsymbol{b}_{ij}^{pq}\right)}{\sum_k \boldsymbol{b}_{ik}^{pq}}, \tag{11}$$

where $\boldsymbol{b}_{ij}^{pq}$ denotes the connection strength between capsules belonging to the ***PrimaryCaps*** layer and the ***Conv_Caps*** layer. The $\boldsymbol{b}_{ij}^{pq}$ is updated as:

$$\boldsymbol{b}_{ij}^{pq} \leftarrow \boldsymbol{b}_{ij}^{pq} + \boldsymbol{u}_{j|i}^{(x+p)(y+q)}\boldsymbol{v}_j^{xy}. \tag{12}$$

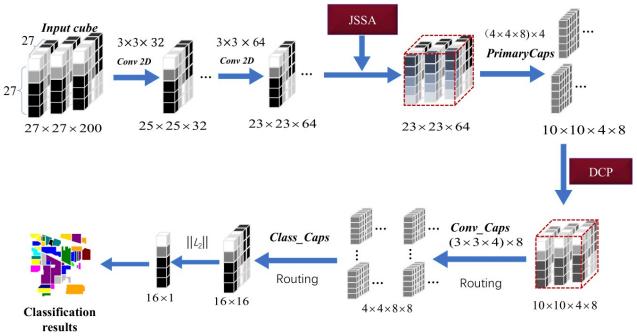

Figure 4. The network structure of JSSR-CapsNet.

3. EXPERIMENTS

3.1. Experimental data and parameter settings

The JSSR-CapsNet is evaluated on the Indian Pines (IN) and the University of Pavia (UP) datasets. The IN dataset consists of 145×145 pixels and 200 bands, including 16 classes. The UP dataset consists of 610×340 pixels and 103 bands, including 9 classes. For the JSSR-CapsNet, we extract 5% and 1% of the labeled samples from IN and UP respectively as the training data, and extract 20% of the labeled samples respectively as the validation data, finally we take the remaining labeled samples as the test data. Then, we use overall accuracy (OA), average accuracy (AA) and Kappa (K) to evaluate the experimental results. The results are compared with the results achieved by three further methods, including SVM [15], 3D-CNN [16] and Conv-CapsNet[6], to assess its performance.

3.2. Evaluations of two datasets

Table 1 The classification accuracy on the IN.

Method	SVM	3D-CNN	Conv-CapsNet	JSSR-CapsNet
OA(%)	78.16±0.21	90.68±0.45	93.75±0.82	**95.61±0.53**
AA(%)	75.12±0.35	91.15±0.73	92.46±0.89	**96.02±0.13**
K*100	75.34±0.53	89.21±0.68	93.02±0.71	**94.99±0.34**

Table 2 The classification accuracy on the UP.

Method	SVM	3D-CNN	Conv-CapsNet	JSSR-CapsNet
OA(%)	88.76±0.19	95.18±0.69	96.61±0.91	**98.68±0.26**
AA(%)	87.01±0.43	93.56±0.32	94.29±0.57	**97.89±0.33**
K*100	84.21±0.77	93.27±0.55	95.69±0.87	**98.31±0.62**

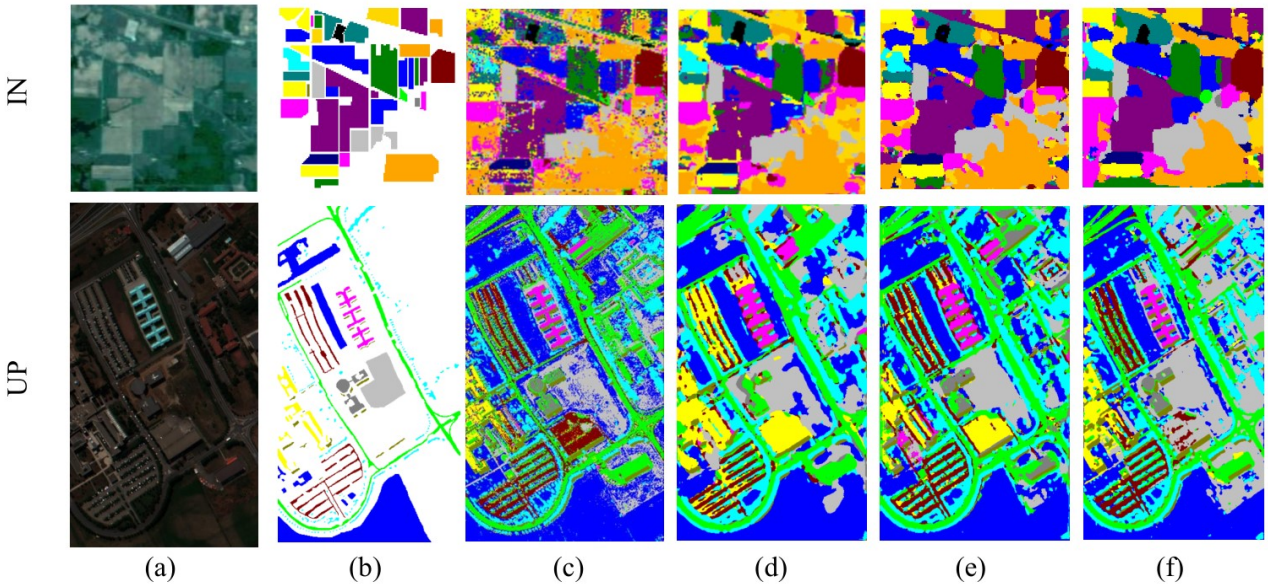

Figure 5. (a) False color image. **(b)** Ground truth labels. **(c)-(f)** denote classification maps obtained from considered algorithms on the IN and UP dataset: (**c**) SVM, (**d**) 3D-CNN, (**e**) Conv-CapsNet, (**f**) JSSR-CapsNet.

As shown in **Table 1** and **Table 2**, the evaluation results of each method were reported with the average accuracy across the 10 folds. As can be seen from two tables, JSSR-CapsNet has the best OA, AA and Kappa results over other comparison algorithms on both real hyperspectral datasets. Comparing with the Conv-CapsNet, it can be seen that JSSR-CapsNet can achieve a better performance because of its JSSA module and DCP module. **Figure 5** details the classification maps obtained from four methods. As can be seen from the classification maps, JSSR-CapsNet can produce a visually smoother map compared with other methods.

4. CONCLUSION

In this paper, we propose a novel joint spatial-spectral representation based capsule network framework (JSSR-CapsNet) for HSI classification. The proposed framework utilizes JSSA module and DCP module to explore recognizable and effective features. The quantitative experimental results indicate that our method has a certain improvement over other methods on two real HSI datasets. However, JSSR-CapsNet model requires a long time to train. In the future, we will carry out in-depth research and improvement on this issue.

5. ACKNOWLEDGEMENTS

This work is supported by National Natural Science foundation of Ningxia Province of China (Project No. 2020AAC02028). We thank the Image and Intelligence Information Processing Innovation Team of National Ethnic Affairs Commission of China for their support.

6. REFERENCES

[1] S. Li, W. Song, L. Fang, Y. Chen, P. Ghamisi and J. A. Benediktsson, "Deep Learning for Hyperspectral Image Classification: An Overview," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 9, pp. 6690-6709, 2019.

[2] P. Jia, M. Zhang, W. Yu, F. Shen and Y. Shen, "Convolutional neural network based classification for hyperspectral data," in *2016 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2016, pp. 5075-5078.

[3] S. Sabour, N. Frosst and G. E. Hinton, "Dynamic routing between capsules," in *Advances in neural information processing systems*. 2017, pp. 1-11.

[4] P. V. Arun, K. M. Buddhiraju and A. Porwal, "Analysis of capsulenets towards hyperspectral classification," in *2018 9th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing*. IEEE, 2018, pp. 1-5.

[5] M. E. Paoletti, J. M. Haut, "Capsule Networks for Hyperspectral Image Classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 4, pp. 2145-2160, 2019.

[6] K. Zhu, Y. Chen, P. Ghamisi, X. Jia, JA. Benediktsson, "Deep Convolutional Capsule Network for Hyperspectral Image Spectral and Spectral-Spatial Classification, " *Remote Sensing*, vol. 11, no. 3, pp. 223-250, 2019.

[7] J. K. Chorowski, D. Bahdanau, D. Serdyuk, "Attention-based models for speech recognition," in *Advances in neural information processing systems*. 2015, pp. 577-585.

[8] F. Wang, M. Jiang, C. Qian, "Residual attention network for image classification," in *2017 IEEE Conference on Computer Vision and Pattern Recognition*.IEEE, 2017, pp. 6450-6458.
[9] A. Galassi, M. Lippi and P. Torroni, "Attention in Natural Language Processing," in *IEEE Transactions on Neural Networks and Learning Systems*. IEEE, 2020, pp. 1-18.
[10] Y. Zeng, X. Guo, H. Wang, M. Geng and T. Lu, "Efficient Dual Attention Module for Real-Time Visual Tracking," in *2019 IEEE Visual Communications and Image Processing*. IEEE, 2019, pp. 1-4.
[11] E. Pan, Y. Ma, "Spectral-Spatial Classification of Hyperspectral Image based on a Joint Attention Network," in *IEEE International Geoscience and Remote Sensing Symposium*. 2019, pp. 413-416.
[12] X. Cui, K. Zheng, L. Gao, "Multiscale spatial-spectral convolutional network with image-based framework for hyperspectral imagery classification," *Remote Sensing*, vol. 11, no. 19, pp. 2220-2240, 2019.
[13] K. Pooja, R. R. Nidamanuri and D. Mishra, "Multi-Scale Dilated Residual Convolutional Neural Network for Hyperspectral Image Classification," in *2019 10th Workshop on Hyperspectral Imaging and Signal Processing: Evolution in Remote Sensing* . IEEE, 2019, pp. 1-5.
[14] Z. Wang, S. Ji, "Smoothed dilated convolutions for improved dense prediction," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2018, pp. 2486-2495.
[15] K. Tan, P. J. Du, "Hyperspectral remote sensing image classification based on support vector machine," *Journal of Infrared and Millimeter Waves*, vol. 27, no. 2, pp. 123-128, 2008.
[16] A. Ben Hamida, A. Benoit, P. Lambert and C. Ben Amar, "3-D Deep Learning Approach for Remote Sensing Image Classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 8, pp. 4420-4434, 2018.