Self-Supervised Learning of Discriminative Spatial–Spectral Features for Hyperspectral Images Clustering

Zhiming Mei and Zengshan Yin

***Abstract*— Deep subspace clustering methods have demonstrated its outstanding capability for hyperspectral image (HSI) clustering. However, there is a lack of explicit supervision to ensure that low-dimensional spatial–spectral features learned from high-dimensional HSI cubes have good subspace structures. In this letter, we propose a cascade residual capsule network (CRCN) for extracting deep spatial–spectral features and introduce the coding rate reduction (CRR) to measure the compactness of learned spatial–spectral features. We exploit the difference between the coding rate of all features and the sum of that of features of each category as loss function, which provides explicit supervision for learning invariant spatial–spectral features to HSI cube flips and rotations and facilitate the subsequent HSI clustering. To learn features that are discriminative to diverse spatial context information on land-cover objects of the same category, we add a regularization term to the loss function, which is a brink loss between the l_2-norm of active vectors of class capsules in the proposed CRCN and the assigned labels. Experimental results on three benchmark HSI datasets demonstrate the effectiveness of the proposed method, which achieves superiority performance over several state-of-the-art HSI clustering methods.**

***Index Terms*— Coding rate reduction (CRR), hyperspectral image (HSI), self-supervised learning, spatial–spectral features, subspace clustering.**

I. Introduction

HYPERSPECTRAL image (HSI) clustering is crucial in many remote sensing applications. It has attracted much attention in recent years. HSI clustering aims at partitioning hyperspectral pixels into clusters, which is still a very challenging task due to sophisticated spatial–spectral structures and high-dimensional spatial and spectral information in HSI data [1].

Recently, subspace clustering methods have shown their capability to process high-dimensional HSI data and their effectiveness in HSI clustering [2]–[4]. These methods can be divided into two categories. One category of them builds affinity matrix directly from the original HSI cube [5], and the other category constructs affinity matrix from the feature space of HSI cube [6]. In general, constructing affinity matrix from deep feature space can well capture the nonlinear low-dimensional structures of HSI data. Affinity matrix constructing and spectral clustering are independent processes in these methods. Deep spatial–spectral features cannot always discriminate between land-cover categories for the subsequent subspace clustering if deep model lacks explicit supervision.

In this letter, we propose a cascade residual capsule network (CRCN) to extract deep spatial–spectral features. The proposed model contains a residual module and a capsule module. The residual module is composed of four residual blocks. Each residual block consists of three convolution layers with shortcut connection [7], batch normalization (BN), and activation operation. The capsule module is developed on capsule networks [8]–[10], which transforms features into spatial–spectral and class capsules. In addition, we introduce the coding rate reduction (CRR) [11] to measure the intrinsic geometric or statistical properties of low-dimensional structures learned from high-dimensional HSI cubes for deep subspace clustering. We maximize the difference between the coding rate of all spatial–spectral features and the sum of that of spatial–spectral features of each category in training process, which provides explicit supervision for deep model to learn equivalent subspace structures of HSI cubes to HSI cube flips and rotations.

II. CRCN-CRR for HSI Clustering

In this section, we first introduce the CRCN, which is proposed for extracting deep spatial–spectral features. The CRCN consists of a residual module and a capsule module. Then, we introduce the CRR as an explicit supervision for deep model to learn equivalent subspace structures of HSI cubes, which are invariant to HSI cube flips and rotations and facilitate the subsequent HSI clustering.

A. CRCN

The network structure of the proposed CRCN consists of a residual module and a capsule module, which is shown in Fig. 1. $f(X)$ denotes the CRCN model, which extracts feature Z from input HSI cube X.

1) Residual Module: In the residual module, each residual block is composed of three BN, rectified linear unit (ReLU), and 3-D convolution layers, which is defined as

$$y = F\left(z, \left\{\mathbf{W}^{\mathbf{r}}{}_i\right\}\right) + \mathbf{W}^{\mathbf{r}}{}_s z, \quad i = 1, 2, 3 \tag{1}$$

$$F\left(z, \left\{\mathbf{W}^{\mathbf{r}}{}_i\right\}\right) = \mathbf{W}^{\mathbf{r}}{}_3 \sigma\left(\mathbf{W}^{\mathbf{r}}{}_2 \sigma\left(\mathbf{W}^{\mathbf{r}}{}_1 \sigma(z) + B_1\right) + B_2\right) + B_3 \tag{2}$$

Manuscript received January 18, 2022; revised March 9, 2022; accepted April 16, 2022. Date of publication April 18, 2022; date of current version May 3, 2022. *(Corresponding author: Zhiming Mei.)*

Zhiming Mei is with the Innovation Academy for Microsatellites, Chinese Academy of Sciences, Shanghai 201210, China, and also with the University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: meizhm@shanghaitech.edu.cn).

Zengshan Yin is with the Innovation Academy for Microsatellites, Chinese Academy of Sciences, Shanghai 201210, China.

Digital Object Identifier 10.1109/LGRS.2022.3168722

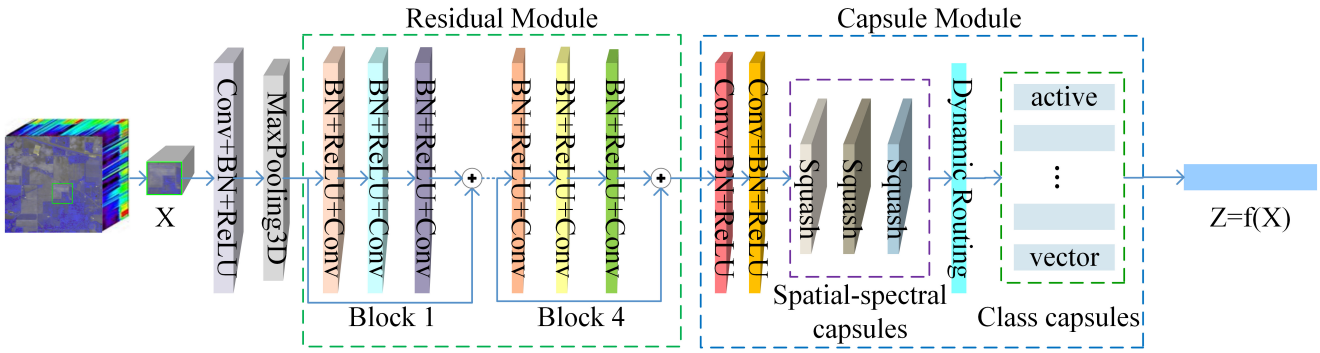

Fig. 1. Network structure of the proposed CRCN. The Indian Pines image is used as an input instance to show the network structure.

where function $F(\cdot)$ is exploited to learn the mapping of a residual block. $\mathbf{W}^{\mathbf{r}}{}_i$ and $\mathbf{W}^{\mathbf{r}}{}_s$ are weight parameter matrix of the ith 3-D convolution layer. B_i and $\sigma(\cdot)$ are the bias and the ReLU activation function, respectively. The residual module is designed to learn and integrate low-, mid-, and high-level spectral features extracted from input HSI cubes. Also, the levels of spectral features can be enriched by the number of residual blocks.

2) Capsule Module: As illustrated in Fig. 1, spatial–spectral feature maps are transformed into spatial–spectral capsules. Afterward, vector u_i of spatial–spectral capsules is linearly transformed by the transformation matrix $\hat{\mathbf{W}}_{ij}$ to obtain vector u_{ij} in the dynamic routing process, which is expressed as

$$u_{ij} = \hat{\mathbf{W}}_{ij} u_i, \quad s_j = \sum_i c_{ij} u_{ij} \tag{3}$$

where s_j is the weighted sum of all vector u_{ij}, and c_{ij} is the coupling coefficient. The initial value b_{ij} of c_{ij} represents the log prior probabilities that the ith spatial–spectral capsule is coupled to the jth class capsule. The initialization of c_{ij} is expressed as

$$c_{ij} = \frac{e^{b_{ij}}}{\sum_{j=1}^{f} e^{b_{ij}}}. \tag{4}$$

The length of vector s_j is scaled down via a nonlinear squash function, which is used as an activation function. The squash function normalizes vector s_j to unit vector with a squeeze coefficient, which is defined as

$$v_j = \frac{\|s_j\|^2}{1 + \|s_j\|^2} \frac{s_j}{\|s_j\|} \tag{5}$$

where $(\|s_j\|^2/(1 + \|s_j\|^2))$ is the squeeze coefficient. One step of the dynamic routing process is briefly illustrated in Fig. 2. The capsule module is employed to learn the intrinsic spatial relationship between the part of labeled land-cover areas in HSI cubes and the whole of that in original HSI.

B. CRR for HSI Clustering

The samples of HSI dataset are HSI cubes $\mathbf{X} = [x_1, \ldots, x_n]$. Each input HSI cube $x \in \mathbb{R}^{H \times W \times C}$ consists of $H \times W$ hyperspectral pixels.

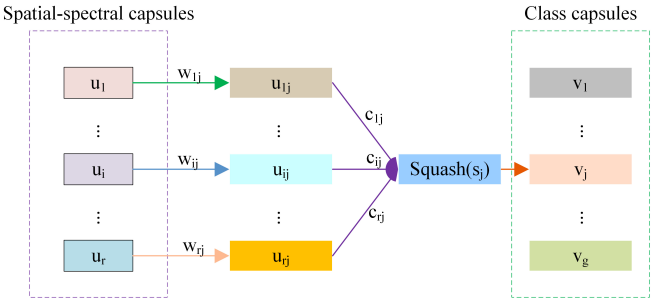

Fig. 2. All vectors of spatial–spectral capsules are transformed by transformation matrices. The transformed vectors are weighted and summed to obtain vectors of class capsules. Every vector of class capsule is activated by the squash function. Active vectors of class capsules have a big scalar product with the prediction coming from the vectors of spatial–spectral capsules.

The proposed CRCN extracts low-dimensional spatial–spectral features $z_i = f(x_i, \theta) \in \mathbb{R}^m$, where $i = 1, \ldots, n$ from high-dimensional input HSI cubes. Spatial–spectral features that are discriminative and facilitate the subsequent HSI clustering task are required to have the following properties: spatial–spectral features of different clusters should be highly uncorrelated and belong to different low-dimensional subspaces, spatial–spectral features of the same cluster should be relatively correlated and belong to the same low-dimensional subspace, and the dimension of spatial–spectral features for each cluster should be as large as possible as long as they stay uncorrelated from other categories. In addition, we require low-dimensional spatial–spectral features to be invariant to HSI cube flips and rotations.

Although the abovementioned properties are all highly desirable for spatial–spectral features, they are difficult to obtain. Lezama *et al.* [12] used a nuclear norm-based geometric loss to enforce orthogonality between categories, but it does not promote diversity in the learned features. To measure the compactness of spatial–spectral features learned by deep model in terms of all these properties, we introduce the concept of coding rate [13] to encode spatial–spectral features $\mathbf{Z} = [z_1, \ldots, z_n]$ of the whole HSI dataset. The coding rate is the average coding length for each learned spatial–spectral feature, which is defined as

$$R(\mathbf{Z}, \epsilon) \doteq \frac{n+m}{2n} \log\det\left(I + \frac{m}{n\epsilon^2}\mathbf{Z}\mathbf{Z}^T\right) \tag{6}$$

where ϵ is the upper bound on the decoding error $E[\|\mathbf{Z} - \hat{\mathbf{Z}}\|_2] \leq \epsilon$ for the decoded $\hat{\mathbf{Z}}$.

Spatial–spectral features $\mathbf{Z}$ contain representations learned from HSI cubes of all k categories. Representations of the same category may belong to one low-dimensional subspace. To estimate the coding rate of each low-dimensional feature subspace, we divide Z and the assigned label of each HSI cube into k subsets according to the category, respectively. The assigned label is the index of the maximal length of active vectors in class capsules, which is a property of the learned feature vectors themselves. The partitions are denoted as $\mathbf{Z} = \mathbf{Z}_1 \cup \cdots \cup \mathbf{Z}_k$, $\mathbf{\Lambda} = \{\mathbf{\Lambda} \in \mathbb{R}^{n\times n} | \mathbf{\Lambda}_j \geq 0, \mathbf{\Lambda}_1 + \cdots + \mathbf{\Lambda}_k = I\}$, $\mathbf{\Lambda}_j$ is a diagonal matrix, and the diagonal entry $\mathbf{\Lambda}_j(i,i)$ of $\mathbf{\Lambda}_j$ indicates the probability that feature i belongs to category j. Then, the coding rate of spatial–spectral features of jth category is

$$R(\mathbf{Z}_j, \epsilon | \mathbf{\Lambda}) \doteq \frac{\mathrm{tr}(\mathbf{\Lambda}_j) + m}{2\mathrm{tr}(\mathbf{\Lambda}_j)} \log\det\left(I + \frac{m}{\mathrm{tr}(\mathbf{\Lambda}_j)\epsilon^2}\mathbf{Z}\mathbf{\Lambda}_j\mathbf{Z}^T\right) \tag{7}$$

where $\mathrm{tr}(\mathbf{\Lambda}_j)$ is the trace of matrix $\mathbf{\Lambda}_j$.

As discussed earlier, the spatial–spectral features of different clusters are preferred to be maximally uncorrelated to each other. So the dimension of all spatial–spectral features space should be as large as possible, and the corresponding coding rate of $\mathbf{Z}$ should be maximized. Meanwhile, the spatial–spectral features of the same cluster are preferred to be highly correlated. Hence, the dimension of spatial–spectral features space of each cluster should be very small, and the corresponding coding rate of each subset should be minimized. Therefore, the compactness of representations $\mathbf{Z}$ learned from $\mathbf{X}$ can be measured by the difference between the coding rate of $\mathbf{Z}$ and the sum of that of all its subsets with a partition $\mathbf{\Lambda}$, which is expressed as

$$\Delta R(\mathbf{Z}, \mathbf{\Lambda}, \epsilon) \doteq R(\mathbf{Z}, \epsilon) - \sum_{j=1}^{k} \frac{\mathrm{tr}(\mathbf{\Lambda}_j) R(\mathbf{Z}_j, \epsilon | \mathbf{\Lambda})}{n}. \tag{8}$$

The loss function of the proposed CRCN is defined as

$$L = -\Delta R(\mathbf{Z}, \mathbf{\Lambda}, \epsilon) + L_B \tag{9}$$

where L_B is the brink loss added to L as a regularization item to learn features that are discriminative to diverse spatial context information on land-cover objects of the same category. The brink loss is defined as

$$L_B = \sum_{i=1}^{n}\sum_{j=1}^{k}(1 - \mathbf{L}_{ij})\sigma(\|\mathbf{V}_{ij}\|_2 - \alpha) + \mathbf{L}_{ij}\sigma(\beta - \|\mathbf{V}_{ij}\|_2) \tag{10}$$

where $\mathbf{L}_{ij} \in \mathbb{R}^{n\times k}$ and $\mathbf{V} \in \mathbb{R}^{n\times k\times o}$ are the assigned label matrix and the output representations of the proposed CRCN, respectively. α and β are the range parameters of l_2-norm of vector $\mathbf{V}_{ij}$. To make the amount of reduction comparable between different spatial–spectral features, each learned spatial–spectral feature is normalized to unit vector. All network parameters of the proposed CRCN are tuned automatically by the backpropagation and stochastic gradient descent (SGD) algorithms [14]–[16], which are exploited to minimize the loss function L.

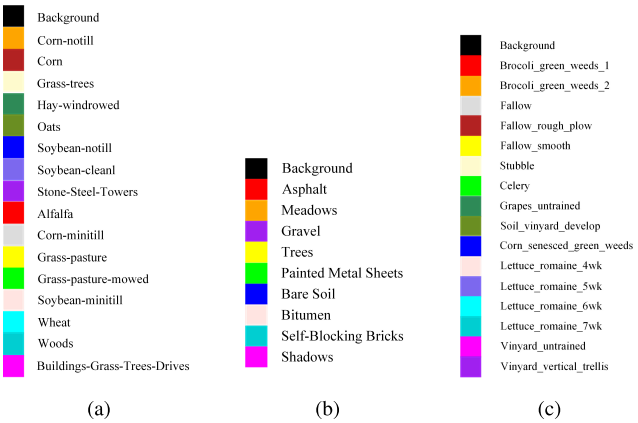

Fig. 3. (a) Color code of the Indian Pines image. (b) Color code of the University of Pavia image. (c) Color code of the Salinas image.

III. Experimental Results

We conduct several experiments to evaluate the performance of the proposed method on three benchmark HSI datasets: the Indian Pines, University of Pavia, and Salinas images. Also, the following clustering methods were selected as benchmarks: S^4C [2], S^2CSC [17], DSC [18], SDSC [19], and LSSD [6]. Not all five methods achieve the state-of-the-art performance on the three datasets. The proposed method is compared with the corresponding state-of-the-art methods on the three datasets.

A. HSI Datasets

To learn representations that are invariant to HSI cube flips and rotations, we randomly select 10%, 2%, and 0.8% pixels of each land-cover category on the Indian Pines, University of Pavia, and Salinas images for flipping and rotating, respectively. Each selected HSI cube is horizontally and vertically flipped and rotated 90°, 180°, and 270°. Low-dimensional spatial–spectral features are learned from these flipped and rotated HSI cubes. The color code of the three datasets is shown in Fig. 3.

B. Evaluation Measures

We use the normalized mutual information (NMI), adjusted rand index (ARI), overall accuracy (OA), average accuracy (AA), and kappa coefficient (Kappa) as evaluation measures to quantify the clustering performance of all compared methods. The NMI is a normalization of the MI score between two clustering. It is a metric that measure the agreement of two clustering on the same dataset. The NMI between ground truth Y and prediction partition C is defined as

$$\mathrm{NMI} = \frac{\sum_{i=1}^{k}\sum_{j=1}^{q}|Y_i \cap C_j| \log\left(\frac{n|Y_i \cap C_j|}{|Y_i||C_j|}\right)}{\sqrt{\left(\sum_{i=1}^{k}|Y_i|\log\left(\frac{|Y_i|}{n}\right)\right)\left(\sum_{j=1}^{p}|C_j|\log\left(\frac{|C_j|}{n}\right)\right)}} \tag{11}$$

where Y_i is the ith cluster in Y, C_j is the jth cluster in C, and n is the total number of samples. The ARI is a metric

TABLE I
QUANTITATIVE EVALUATIONS OF ALL COMPARED CLUSTERING METHODS ON THE INDIAN PINES DATASET

	S^4C	DSC	S^2CSC	SDSC	CRCN-CRR
NMI	0.7357	0.7982	0.8080	0.8262	0.8702
ARI	0.4749	0.6174	0.6727	0.6809	0.7100
OA(%)	63.46	74.00	75.03	75.68	76.94
AA(%)	50.19	52.80	53.28	53.79	61.15
Kappa(%)	59.87	71.09	72.10	72.85	74.66
Runtime(s)	1982.53	167.62	578.39	582.81	655.97

that measure the similarity between two clustering. The ARI is defined as

$$\mathrm{ARI} = \frac{\sum_{ij}\binom{t_{ij}}{2} - \left(\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}\right)/\binom{t}{2}}{\frac{1}{2}\left(\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}\right) - \left(\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}\right)/\binom{t}{2}} \tag{12}$$

where $t_{ij} = |Y_i \cap C_j|$, $a_i = \sum_j t_{ij}$, and $b_j = \sum_i t_{ij}$. Assuming that $M \in \mathbb{R}^{k\times k}$ denotes the error matrix of clustering results. Then, the formulas of OA, AA, and Kappa are defined as

$$\mathrm{OA} = \frac{\sum_{i=1}^{k} M_{ii}}{\sum_{i=1}^{k}\sum_{j=1}^{k} M_{ij}} \tag{13}$$

$$\mathrm{AA} = \frac{1}{k}\sum_{i=1}^{k}\frac{M_{ii}}{\sum_{j=1}^{k} M_{ij}} \tag{14}$$

$$\mathrm{P} = \frac{\sum_{h=1}^{k}\sum_{i=1}^{k}\sum_{j=1}^{k} M_{ih}M_{hj}}{\left(\sum_{i=1}^{k}\sum_{j=1}^{k} M_{ij}\right)^2} \tag{15}$$

$$\mathrm{Kappa} = \frac{\mathrm{OA} - \mathrm{P}}{1 - \mathrm{P}}. \tag{16}$$

C. Experimental Settings

For a fair comparison, all methods in our experiments use the same experimental settings, including data processing, data augmentation, and parameter settings. In addition, all the parameters of the compared clustering methods were manually tuned to the optimum. Both the batch size and the epochs are set to 300. The spatial size of input HSI cube is set to 11×11. The range parameters α and β are set to 0.1 and 0.9, respectively. The iteration times r of dynamic routing is set to 2. The initial learning rate and weight decay of each epoch are both set to 10^{-4}. All experiments are conducted with Intel Core i7-9700K, 32-GB RAM, GeForce GTX Titan X, TensorFlow 1.15.0, cuda 10.0, cudnn 7.6.0, and python 3.7.9.

D. Clustering Results and Analysis

The quantitative evaluations and clustering maps obtained by all compared clustering methods on the Indian Pines dataset are shown in Table I and Fig. 4, respectively. According to Table I, the proposed CRCN-CRR outperforms other compared methods, which indicates that the proposed CRCN-CRR provides effective supervision for a deep model to learn intrinsic subspace structures of HSI cubes that are invariant to HSI cube flips and rotations. In Fig. 4, we can see that our method

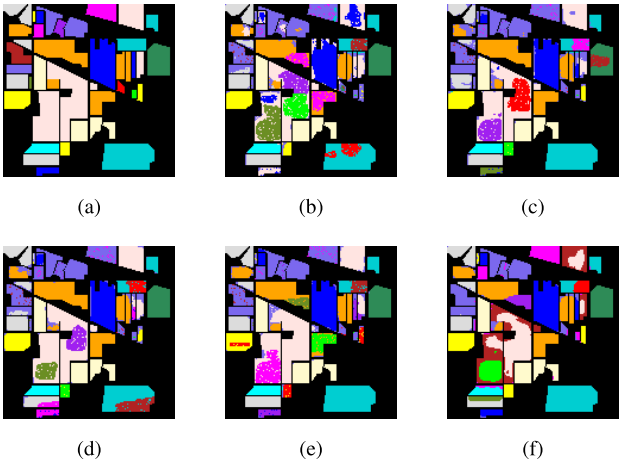

Fig. 4. Clustering maps on the Indian Pines image. (a) Ground truth. (b) S^4C. (c) DSC. (d) S^2CSC. (e) SDSC. (f) CRCN-CRR.

TABLE II
QUANTITATIVE EVALUATIONS OF ALL COMPARED CLUSTERING METHODS ON THE UNIVERSITY OF PAVIA DATASET

	LSSD	DSC	S^2CSC	SDSC	CRCN-CRR
NMI	0.8702	0.8957	0.9054	0.9162	0.9348
ARI	0.7109	0.7459	0.7568	0.7658	0.8356
OA(%)	77.79	79.54	82.83	84.94	89.18
AA(%)	62.83	66.44	77.08	80.63	85.71
Kappa(%)	68.24	74.62	78.21	79.67	83.79
Runtime(s)	2032.17	698.57	2765.36	3182.29	3647.56

TABLE III
QUANTITATIVE EVALUATIONS OF ALL COMPARED CLUSTERING METHODS ON THE SALINAS DATASET

	LSSD	DSC	S^2CSC	SDSC	CRCN-CRR
NMI	0.9001	0.8937	0.9011	0.9120	0.9218
ARI	0.7843	0.7601	0.7913	0.8227	0.8535
OA(%)	85.98	83.54	88.44	92.75	93.54
AA(%)	87.81	83.11	89.76	93.01	93.60
Kappa(%)	89.62	88.44	90.01	92.87	94.86
Runtime(s)	3487.24	867.31	3758.96	3968.57	4325.34

obtains much smoother clustering maps. The clustering results and the clustering maps obtained by all compared clustering methods on the University of Pavia dataset are shown in Table II and Fig. 5, respectively. From Table II, we can see that the proposed CRCN-CRR achieves superior performance. From Fig. 5, we can observe that the clustering map of our method produces areas that are more homogeneous and less outlier than other compared methods. The clustering results and the clustering maps obtained by all compared clustering methods on the Salinas dataset are shown in Table III and Fig. 6, respectively. From Table III, we can see that our method achieves best clustering results. As can be observed in Fig. 6, the proposed method obtains much better clustering maps than other compared methods, which indicates that the

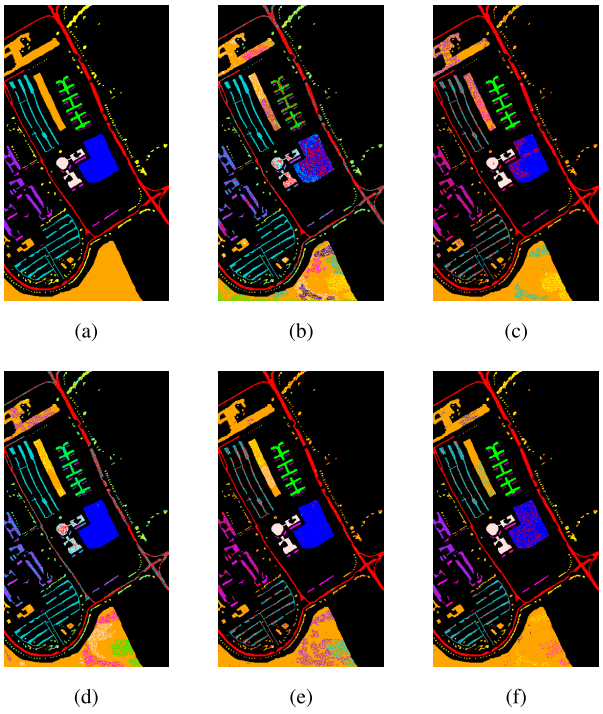

Fig. 5. Clustering maps on the University of Pavia image. (a) Ground truth. (b) LSSD. (c) DSC. (d) S^2CSC. (e) SDSC. (f) CRCN-CRR.

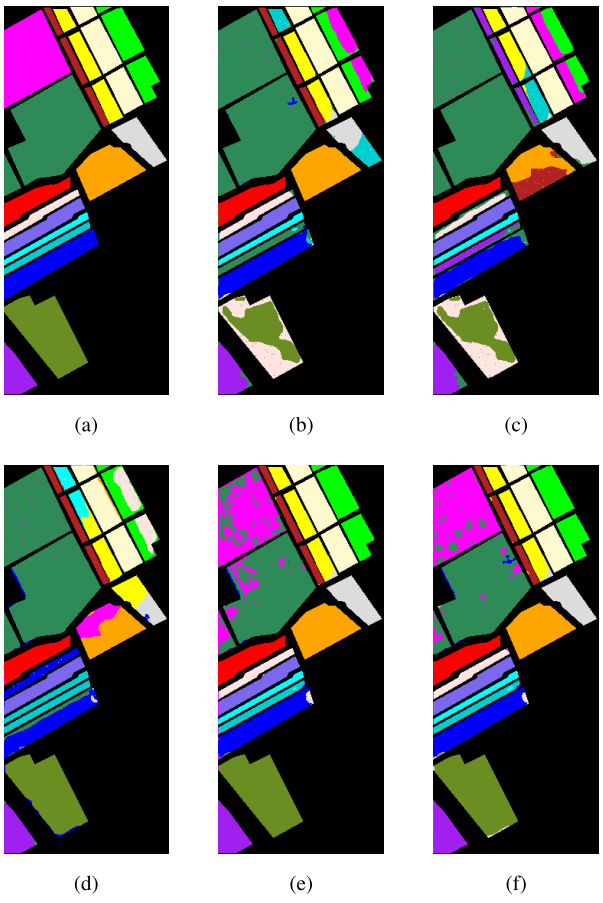

Fig. 6. Clustering maps on the Salinas image. (a) Ground truth. (b) LSSD. (c) DSC. (d) S^2CSC. (e) SDSC. (f) CRCN-CRR.

spatial–spectral features learned by our method draw closer to aforementioned subspace clustering properties.

IV. Conclusion

In this letter, we propose a novel deep model named CRCN to extract low-dimensional deep spatial–spectral features from high-dimensional HSI cubes. In addition, we introduce the CRR to provide explicit supervision for the deep model to learn equivalent subspace structures of HSI cubes, which are invariant to HSI cube flips and rotations. The proposed CRCN-CRR method can measure the compactness of spatial–spectral features learned by the deep model in terms of good clustering properties. Experimental results demonstrate the effectiveness of the proposed method.

References

[1] L. Zhang, L. Zhang, B. Du, J. You, and D. Tao, "Hyperspectral image unsupervised classification by robust manifold matrix factorization," *Inf. Sci.*, vol. 485, pp. 154–169, Jun. 2019.

[2] H. Y. Zhang, H. Zhai, L. P. Zhang, and P. X. Li, "Spectral–spatial sparse subspace clustering for hyperspectral remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 6, pp. 3672–3684, Jun. 2016.

[3] H. Zhai, H. Zhang, L. Zhang, P. Li, and A. Plaza, "A new sparse subspace clustering algorithm for hyperspectral remote sensing imagery," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 1, pp. 43–47, Jan. 2017.

[4] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2765–2781, Nov. 2013.

[5] R. Wang, F. Nie, and W. Yu, "Fast spectral clustering with anchor graph for large hyperspectral images," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 11, pp. 2003–2007, Nov. 2017.

[6] Y. Qin, L. Bruzzone, and B. Li, "Learning discriminative embedding for hyperspectral image clustering based on set-to-set and sample-to-sample distances," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 1, pp. 473–485, Jan. 2020.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[8] M. E. Paoletti *et al.*, "Capsule networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 4, pp. 2145–2160, Apr. 2018.

[9] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 3859–3869.

[10] A. Sepas-Moghaddam, A. Etemad, F. Pereira, and P. L. Correia, "CapsField: Light field-based face and expression recognition in the wild using capsule routing," *IEEE Trans. Image Process.*, vol. 30, pp. 2627–2642, 2021.

[11] Y. Yu, K. Ho Ryan Chan, C. You, C. Song, and Y. Ma, "Learning diverse and discriminative representations via the principle of maximal coding rate reduction," 2020, *arXiv:2006.08558*.

[12] J. Lezama, Q. Qiu, P. Muse, and G. Sapiro, "OLE: Orthogonal low-rank embedding, a plug and play geometric loss for deep learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8109–8118.

[13] Y. Ma, H. Derksen, W. Hong, and J. Wright, "Segmentation of multivariate mixed data via lossy data coding and compression," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 9, pp. 1546–1562, Sep. 2007.

[14] S. L. Goh and D. P. Mandic, "Stochastic gradient-adaptive complex-valued nonlinear neural adaptive filters with a gradient-adaptive step size," *IEEE Trans. Neural Netw.*, vol. 18, no. 5, pp. 1511–1516, Sep. 2007.

[15] V. J. Mathews and Z. H. Xie, "A stochastic gradient adaptive filter with gradient adaptive step size," *IEEE Trans. Signal Process.*, vol. 41, no. 6, pp. 2075–2087, Jun. 1993.

[16] S. Bonnabel, "Stochastic gradient descent on Riemannian manifolds," *IEEE Trans. Autom. Control*, vol. 58, no. 9, pp. 2217–2229, Sep. 2013.

[17] J. Zhang *et al.*, "Self-supervised convolutional subspace clustering network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5468–5477.

[18] P. Ji, T. Zhang, H. D. Li, M. Salzmann, and I. Reid, "Deep subspace clustering networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 24–33.

[19] K. Li, Y. Qin, Q. Ling, Y. Wang, Z. Lin, and W. An, "Self-supervised deep subspace clustering for hyperspectral images with adaptive self-expressive coefficient matrix initialization," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 3215–3227, 2021.