Convolutional Transformer Network for Hyperspectral Image Classification

Zhengang Zhao, Dan Hu, Hao Wang, and Xianchuan Yu, *Senior Member, IEEE*

***Abstract*— Convolutional neural networks (CNNs) have attained remarkable performance in hyperspectral image (HSI) classification. However, the existing CNNs are restricted by their limited receptive field in HSI classification. Recently, transformer networks have proved to be promising in many tasks thanks to the global receptive field, but they easily ignore some local information that is important for HSI classification. In this letter, we propose a novel method entitled convolutional transformer network (CTN) for HSI classification. In order to make full use of spectral information and spatial information, the method adopts center position encoding (CPE) to merge spectral features and pixel positions. Furthermore, the proposed method introduces convolutional transformer (CT) blocks. It effectively combines convolution and transformer structures together to capture local–global features of HSI patches, which is contributive for HSI classification. Experimental results on public datasets demonstrate the superiority of our method compared with several state-of-the-art classification methods. The codes of this work will be available at https://github.com/sky8791 to facilitate reproducibility.**

***Index Terms*— Center position encoding (CPE), convolutional neural network (CNN), hyperspectral image (HSI) classification, transformer.**

I. Introduction

HYPERSPECTRAL images (HSIs) capture spectral signals (pixels) within observed scenes at hundreds of narrow spectral bands spanning from visible spectrum to infrared spectrum. These signals carry abundant spectral information that reflects different remote sensing materials. With the advantage of rich spectral information, HSIs have been applied in many fields, such as agriculture, environment monitoring, and land scene classification [1]. In these applications and studies, HSI classification is an important step and has become an active research topic in remote sensing and earth observation [2], [3].

Manuscript received October 2, 2021; revised February 17, 2022 and March 31, 2022; accepted April 20, 2022. Date of publication April 22, 2022; date of current version May 5, 2022. This work was supported by the National Natural Science Foundation of China under Grant 42172323. *(Corresponding author: Xianchuan Yu.)*

Zhengang Zhao is with the School of Artificial Intelligence, Beijing Normal University, Beijing 100875, China, and also with the Business College, Hebei Normal University, Shijiazhuang 050024, China (e-mail: zhaozhengang1986@163.com).

Dan Hu is with the Department of Radiology and the Biomedical Research Imaging Center (BRIC), University of North Carolina at Chapel Hill, Chapel Hill, NC 27599 USA (e-mail: danhu@email.unc.edu).

Hao Wang and Xianchuan Yu are with the School of Artificial Intelligence, Beijing Normal University, Beijing 100875, China (e-mail: haowang19@mail.bnu.edu.cn; yuxianchuan@163.com).

Digital Object Identifier 10.1109/LGRS.2022.3169815

HSI classification aims to assign each pixel to a proper class label [1], [4]. In the last decade, many approaches leveraging machine learning have been proposed to solve the task, such as support vector machine [5], random forest [6], k-nearest neighbor [7], and sparse representation [8]. Due to information redundancy and high correlation among spectral bands, these traditional classifiers result in low performance. Moreover, these methods are often susceptible to noises.

Recently, convolutional neural networks (CNNs) based on deep learning have been successfully applied on various vision tasks and obtained remarkable performance in HSI classification [1], [9]. In terms of whether spatial information is used, CNN-based methods for HSI classification fall into two main categories: spectral-based methods and spectral-spatial-based methods. Generally, the spectral-based methods only use the spectral features when classifying HSIs. They treat hyperspectral data as a collection of spectral signatures and ignore the spatial structures among pixels. For instance, 1-D CNN [10] randomly selected some labeled pixels from HSI data as training set and adopted five layers to extract spectral features for HSI classification. Although spectral-based methods make full use of the spectral information, they are difficult to attain a breakthrough in classification performance. In contrast, the spectral-spatial-based methods combine both spectral and spatial information of HSIs. These methods usually take the target pixel and its neighbor pixels as a subcube sample (i.e., a patch) whose class label is that of its central pixel. In [11], the authors proposed a deep feature fusion network (DFFN) with 2-D CNN to extract spectral-spatial features for HSI classification, which introduced residual learning to increase the network depth and fused the outputs of different hierarchical layers together. In [12], the authors pre-processed HSI data first with an autoencoder and then adopted shallow CNN with 1×1 kernels to extract features. Furthermore, since HSIs are originally 3-D cube data, the researchers [13] directly adopted 3-D CNN for HSI classification, which simultaneously explored both the spatial and spectral information of HSI data. Zhong *et al.* [14] designed a spectral-spatial residual network (SSRN) which used 3-D convolutional layers to learn discriminative features from spectral signatures and spatial contexts in HSIs. Although CNNs have achieved acclaimed performance in HSI classification by virtue of their powerful feature-extraction ability, they are still hampered by their limited receptive fields [15]. In general, CNNs extract features relying on myriad convolutional kernels with narrow scope (e.g., 3×3 and 5×5), and these kernels are locally connected

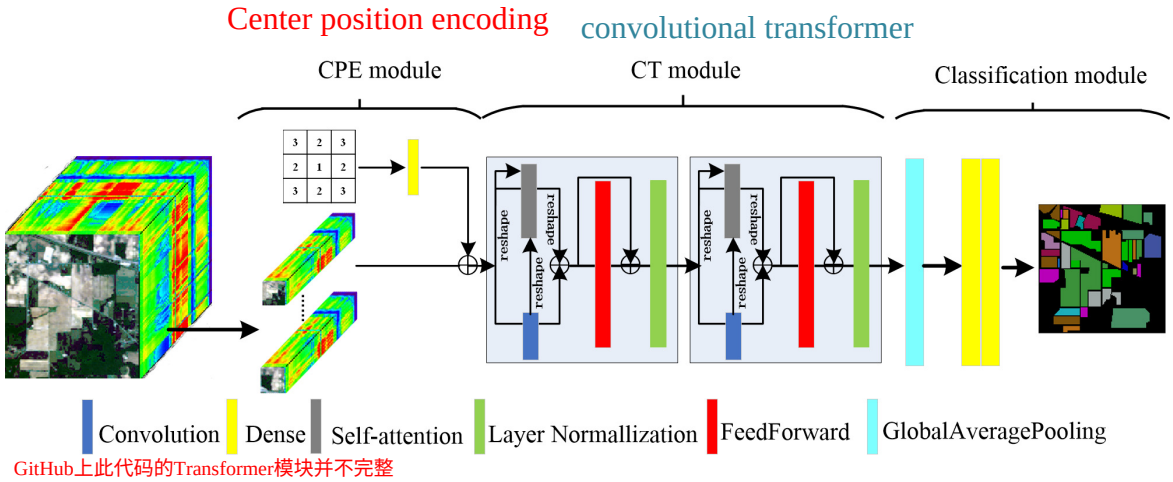

Fig. 1. Flowchart of the proposed CTN. HSI data are clipped into many patches. Each patch is processed by three modules in turn: CPE module, CT module, and classification module. And the output of the method is the category label. In this flowchart, ⊕ denotes matrix addition.

to the image. That is, the features extracted by CNNs tend to be local features. As a result, the small kernels limit the CNN receptive field.

At present, transformer networks have shown promising performance in natural language processing and visual tasks [16]–[18]. They mainly consist of many self-attention and feedforward layers which inherit global receptive field to capture global dependent features of the samples. In [17], the authors split images into fixed-size patches and linearly symbolized each of them as pixel sequences with position encoding. Wu *et al.* [18] introduced convolutions into vision transformers to improve performance and robustness. However, the operation flattening HSI patches into pixel sequences destroys the internal spatial information of the patches. And the position encoding cannot sufficiently reflect spatial information of HSI samples. Moreover, some local features are useful for HSI classification but they cannot be extracted well.

To address these issues, in this letter, we propose a novel method, convolutional transformer network (CTN), for HSI classification. CTN adopts 2-D center position encoding (CPE) instead of 1-D sequence position encoding to preserve the spatial structure of HSI samples. Meanwhile, it effectively combines local features and global features via convolutional transformer (CT) block, which is useful for improving the performance of HSI classification.

To summarize, the major contributions of this letter are listed as follows. First, we proposed CPE to obtain spatial position features for HSI patches. Second, we designed CT block that is used to extract local-global features for HSI patches. Finally, a series of experiments were conducted on two benchmark HSI datasets, and the results show that the proposed CTN achieves the best performance compared with several state-of-the-art methods.

II. Proposed Method

This section presents our CTN for HSI classification. Fig. 1 illustrates the architecture of the method. Section II-A introduces the CPE module. Section II-B shows the CT block. Detailed workflow of the proposed method is in Section II-C.

7	6	5	4	5	6	7
6	5	4	3	4	5	6
5	4	3	2	3	4	5
4	3	2	1	2	3	4
5	4	3	2	3	4	5
6	5	4	3	4	5	6
7	6	5	4	5	6	7

Fig. 2. Pixel positions in a sample when the spatial size of the sample is 7 × 7.

A. Center Position Encoding

Spatial information is useful for HSI classification. In order to better describe the relative position among pixels and keep the rotation invariance of samples, the pixel positions in one sample are defined as

$$\text{pos}(i, j) = |i - x_c| + |j - y_c| + 1 \tag{1}$$

where (x_c, y_c) represents the central position coordinates of a sample, and (i, j) represents the coordinates of any pixel in the sample. To visualize the CPE, we show an example when the spatial size of a sample is 7 × 7 (see Fig. 2).

CPE aims to preserve the spatial structures of HSI patches in the model. It transforms the pixel positions into embedded vectors. Here, the learned positional embedding is used to encode the pixel positions

$$\text{cpe}(\mathbf{x}) = \mathbf{x}\mathbf{W} \tag{2}$$

where $\mathbf{W}$ is parameter matrix, and $\mathbf{x}$ is the position matrix. After the position embedded vectors are got, they are added to the hyperspectral data and then passed to the next layer. CPE structure is shown in Fig. 1 (see CPE module).

B. Convolutional Transformer Block

CT block is the core part of the proposed method. It is used to extract local–global features. It mainly consists of 2-D convolution layer, self-attention layer, feed-forward layer, and layer-normalization layer. The architecture of the CT block is shown in Fig. 1 (see CT module).

1) Two-Dimensional Convolution: The convolution layer usually adopts many neurons (kernels) to compute feature maps. Each kernel computes the dot product between its weights and a small region of the input volume matched to it. It is formulated as follows [11]:

$$\mathrm{con}(\mathbf{V}) = \Big|\Big|_{j=1}^{J} \delta\left(\mathbf{V} * \mathbf{W}_j^{r1\times r2} + \mathbf{b}_j\right) \tag{3}$$

where $\mathbf{V}$ is the input volume, $\mathbf{W}_j^{r1\times r2}$ is the jth kernel with the size of $r1 \times r2$, and $\mathbf{b}_j$ is the jth bias. $||$ denotes concatenation operation, and $*$ represents convolution operation. $\delta(\cdot)$ is the nonlinear activation function GELU [19]. The number of kernels J is 64, and the size of kernels is 3×3.

2) Self-Attention: Self-attention is used to extract global features based on convolutional features. Its inputs are the query, key, and value (i.e., $\mathbf{Q}, \mathbf{K}, \mathbf{V}$) [15]. The values are first processed by the convolution function. The dot products are applied to compute the query with all keys. And then the softmax function [15] is used to compute the weights on the values. Our self-attention is defined as follows:

$$\mathrm{SA}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{softmax}\left(\left(\mathbf{Q}\mathbf{W}_Q\right)\left(\mathbf{K}^{\mathrm{T}}\mathbf{W}_K\right)\right)\mathrm{con}(\mathbf{V}) \tag{4}$$

where $\mathbf{W}_Q, \mathbf{W}_K$ are parameter matrices, $\{\cdot\}^{\mathrm{T}}$ is transposition, and $\mathrm{con}(\cdot)$ is the convolutional function [see (3)].

3) Feedforward: Feed-forward is applied to further transform the output vectors through fully connected layers, and it is shared across all the output vectors. It holds two linear transformations with a GELU [19] activation function in between. Its equation is defined as follows:

$$\mathrm{FF}(\mathbf{x}) = \mathrm{GELU}(\mathbf{x}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2 \tag{5}$$

where $\mathbf{W}_1, \mathbf{W}_2$ are parameter matrices, and $\mathbf{b}_1, \mathbf{b}_2$ are their biases, respectively.

4) Layer Normalization: Layer-normalization is an effective way to reduce the internal covariate shift and prevent overfitting during the training phase [20]. It is represented as

$$\mathrm{LN}(x_i) = g \cdot \frac{x_i - E(\mathbf{x})}{\sqrt{\mathrm{Var}(\mathbf{x}) + \varepsilon}} + b \tag{6}$$

where $E(\mathbf{x})$ and $\mathrm{Var}(\mathbf{x})$ represent the mean and standard deviation of $\mathbf{x}$, g, and b are the learned parameters, ε is a very small constant value.

C. Convolutional Transformer Network

As shown in Fig. 1, the proposed CTN is essentially made up of three modules: CPE module, CT module, and classification module. In this work, HSI data are first preprocessed by principal component analysis (PCA) and the top 64 components are selected. And then they are clipped into many patches of the same size.

In the first stage, when given an HSI patch with the target pixel in the center, the HSI patch is represented by CPE model. And the output contains spectral information and spatial information together. In the second stage, the output is fed into CT module which consists of two CT blocks. The CT block extracts local-global features from the spectral–spatial features for the input patch. As the final stage, classification module is composed of one global average pooling (GAP) [21] layer and two dense layers. The activation function of the first dense is GELU, and the number of its neurons is 32. The activation function of the other dense layer is softmax, and the number of its neurons is the number of categories. The categorical cross entropy is employed as the loss function, defined as

$$\mathrm{Loss} = -\sum_{i=1}^{c} y_i \log(z_i) \tag{7}$$

where c is the number of classes, z_i is the output of Classification module, y_i is the label value, and $y_i \in \{0, 1\}$ (if y_i is the ith class $y_i = 1$, else $y_i = 0$). Adam optimizer is adopted in the training process in pursuit of robustness.

TABLE I
CLASS, NAME, AND NUMBER OF SAMPLES ON UP DATASET

Class	Name	Number	Class	Name	Number
1	Asphalt	6631	6	Bare-soil	5029
2	Meadows	18649	7	Bitumen	1330
3	Gravel	2099	8	Self-blocking-bricks	3682
4	Trees	3064	9	Shadows	947
5	Painted-metal-sheets	1345			

TABLE II
CLASS, NAME, AND NUMBER OF SAMPLES ON IP DATASET

Class	Name	Number	Class	Name	Number
1	Alfalfa	46	9	Oats	20
2	Corn-notill	1428	10	Soybean-notill	972
3	Corn-mintill	830	11	Soybean-mintill	2455
4	Corn	237	12	Soybean-clean	593
5	Grass-pasture	483	13	Wheat	205
6	Grass-trees	730	14	Woods	1265
7	Grass-pasture-mowed	28	15	Buildings-grass-trees-drives	386
8	Hay-windrowed	478	16	Stone-steel-towers	93

III. EXPERIMENTS AND ANALYSIS

In order to verify the classification performance of the proposed method, we conducted a series of experiments on two widely used HSI datasets: University of Pavia (UP) and Indian pines (IP) [2], [15]. UP dataset is of the size 610×340 and contains 42 776 labeled samples of nine categories. IP dataset is of the size 145×145 and contains 10 249 labeled samples, falling into 16 categories. Tables I and II show the class, name, and number of samples on UP and IP datasets, respectively. Furthermore, four evaluation metrics are adopted to quantitatively analyze the performance of the methods, including per-class accuracy, overall accuracy (OA), average accuracy (AA), and kappa coefficient (κ) [1], [13]. OA refers to the proportion of all correctly classified samples; AA refers to the average classification accuracy of each class; kappa coefficient tells the consistency between classification results and ground truth.

A. Comparing With Other Methods

In this section, the proposed CTN was compared with several state-of-the-art classification methods, including DFFN [11], 3-D CNN [13], MANF [22], and Transformer [23]. In these experiments, HSI data were preprocessed

TABLE III

CLASSIFICATION ACCURACY (%) ACHIEVED BY DIFFERENT METHODS ON UP DATASET

	DFFN	3D-CNN	MANF	Transformer	CTN
1	93.12±3.00	88.98±4.10	**98.09±1.17**	85.74±3.73	97.34±1.65
2	99.76±0.33	99.21±0.64	99.35±0.78	99.61±0.17	**99.89±0.18**
3	88.11±4.08	80.07±6.56	77.62±11.94	78.38±6.81	**89.32±5.55**
4	92.39±3.51	87.75±3.72	**96.28±1.68**	77.98±4.62	92.14±2.07
5	99.80±0.37	**99.92±0.20**	99.86±0.16	98.83±1.64	99.79±0.23
6	98.66±1.24	89.85±6.97	97.58±2.14	97.02±2.23	**99.31±0.68**
7	98.71±1.92	93.47±10.79	93.03±6.51	85.51±6.35	**99.09±0.68**
8	83.88±7.12	69.33±9.51	91.49±5.70	71.55±5.47	**93.54±3.28**
9	87.29±8.29	62.32±16.29	**99.73±0.46**	56.28±13.22	86.38±9.41
OA	95.83±0.59	91.22±2.24	96.81±0.86	90.72±0.33	**97.48±0.44**
AA	93.53±1.23	85.66±3.61	94.78±1.76	83.43±1.36	**95.20±1.15**
κ	94.47±0.79	88.25±3.06	95.77±1.15	87.64±0.43	**96.65±0.59**

TABLE IV

CLASSIFICATION ACCURACY (%) ACHIEVED BY DIFFERENT METHODS ON IP DATASET

	DFFN	3D-CNN	MANF	Transformer	CTN
1	96.83±4.50	85.12±11.31	87.80±11.49	90.73±10.90	**99.02±1.20**
2	98.78±0.87	96.41±1.27	97.72±0.87	95.75±1.02	**99.00±0.65**
3	98.05±1.10	96.17±1.61	96.17±2.74	94.57±1.72	**98.65±0.82**
4	94.79±3.49	89.06±7.02	90.80±8.47	91.83±4.34	**96.24±3.92**
5	97.95±1.97	97.42±1.38	97.68±1.29	95.56±1.29	**98.69±1.36**
6	99.04±0.66	99.67±0.71	99.59±0.49	97.89±1.21	**99.77±0.23**
7	93.20±5.95	94.40±7.42	76.00±21.32	79.60±15.23	**99.20±1.60**
8	99.86±0.30	**99.98±0.07**	99.84±0.49	99.67±0.56	**99.98±0.07**
9	93.89±8.03	82.78±15.80	91.67±9.70	58.89±19.12	**99.44±1.67**
10	98.56±0.88	96.78±1.79	96.86±1.92	94.34±1.99	**98.86±0.79**
11	98.64±0.90	97.89±0.65	98.44±0.76	97.52±1.27	**99.32±0.36**
12	96.98±1.82	92.72±3.06	95.94±1.85	93.69±2.11	**97.96±1.09**
13	99.29±1.03	99.35±0.84	99.40±0.75	97.34±2.29	**99.62±0.65**
14	99.72±0.22	99.44±0.47	99.63±0.55	99.07±0.47	**99.90±0.14**
15	99.31±0.90	96.37±3.11	96.66±3.13	94.87±4.11	**99.42±0.63**
16	95.36±3.89	95.95±4.23	94.05±6.88	81.19±9.62	**96.07±2.06**
OA	98.58±0.19	97.19±0.48	97.73±0.89	96.20±0.49	**99.11±0.10**
AA	97.52±0.87	94.97±0.85	94.89±2.31	91.41±2.02	**98.82±0.20**
κ	98.38±0.22	96.79±0.54	97.41±1.01	95.67±0.56	**98.99±0.12**

with PCA and the top 64 components were retained. The epoch was set to 300, the batch size 32, and the patch size 15 × 15. The weight parameters were optimized by Adam, and the learning rate was set to 0.001. Other settings of the comparison methods followed the original paper. In particular, the decoder network of Transformer was replaced by classification module of CTN to achieve HSI classification.

For UP dataset, 1% of the labeled samples are randomly selected as training samples, and the rest of the labeled samples were test samples. For IP dataset, we randomly selected 10% labeled pixels from the ground truth map as training samples, while the rest are preserved for testing. We performed these methods ten times independently on each dataset, and the mean accuracy and standard deviations over the ten results were recorded. The classification results of different methods are shown in Tables III and IV. We highlight the best results in bold.

As shown in Tables III and IV, in terms of per-class accuracy, the proposed CTN yields best results in most categories. For example, five categories got the highest accuracy in UP dataset and all categories in IP dataset. From the perspective of OA, AA and κ, the classification results of CTN have a relatively stable improvement compared with other methods on UP and IP datasets. The experimental results on the two datasets demonstrate the superiority of the proposed CTN in performance.

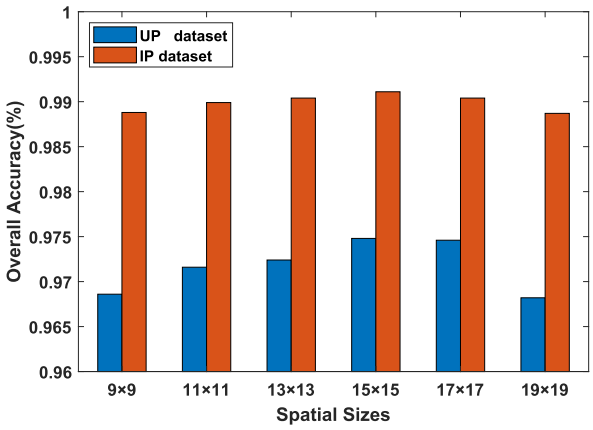

Fig. 3. OAs of the proposed CTN on UP and IP datasets with different spatial sizes.

B. Influence of the Spatial Size

The spatial size of patches has a considerable impact on HSI classification. In this section, we conducted several experiments to explore the impact of the spatial size on classification performance. For UP dataset, 1% labeled samples are randomly selected as the training samples while the other labeled samples are testing samples. For IP dataset, we randomly selected 10% labeled samples as the training samples while the other labeled samples are testing samples. The spatial sizes of patches were set to 9 × 9, 11 × 11, 13 × 13, 15 × 15, 17 × 17, and 19 × 19. Fig. 3 shows the OAs of the proposed CTN on UP and IP datasets with different spatial sizes.

From Fig. 3, we find that the spatial size of patches is important for the proposed method. On UP and IP datasets, the OAs increase first and then decrease as the spatial size expands. The reason for the initial growth in classification performance is that a larger spatial size makes the patches contain more useful pixels whose features are helpful to classify the target pixels. But the classification performance degrades if the spatial size continues to grow excessively because there will be many interfering pixels involved whose labels are different from target pixels in the neighborhoods. These interfering pixels prevent the improvement of classification performance. Therefore, a suitable spatial size is important for our method. In the proposed CTN, the spatial size is set to 15 × 15.

C. Effectiveness of CPE

CPE module plays an essential role in the proposed method. It improves the classification performance by means of fusing spectral information and position information of pixels. In order to verify the necessity of the CPE, we applied the methods either with and without the CPE on both UP and IP datasets. Under the same configurations, the OAs, AAs, and κs of the methods are shown in Table V.

According to Table V, it is evident that the CPE has boosted the classification performance on both datasets. For instance,

TABLE V
OAS, AAS, AND κS OF THE METHODS WITH AND WITHOUT THE CPE ON UP AND IP DATASETS

	model	OA(%)	AA(%)	$\kappa\times100$
UP dataset	without CPE	96.24±0.61	92.13±1.44	95.01±0.82
	with CPE	97.48±0.44	95.20±1.15	96.65±0.59
IP dataset	without CPE	98.75±0.21	98.33±0.36	98.57±0.24
	with CPE	99.11±0.10	98.82±0.20	98.99±0.12

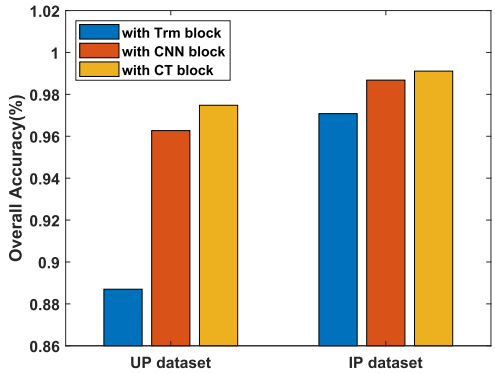

Fig. 4. OAs of the methods with Trm, CNN and CT blocks on UP and IP datasets.

the OA of the method with the CPE is improvement of +1.24 on UP dataset and +0.36 on IP dataset. Furthermore, in terms of standard deviation, the method with the CPE is more stable than that without the CPE.

D. Effectiveness of CT Block

The experiments in this section is to prove the effectiveness of the CT block. In these experiments, the method only with Transformer (Trm) block and the method only with CNN block were used as the experiment group, while other conditions remained constant. Fig. 4 shows the OAs of the methods with Trm, CNN, and CT blocks on UP and IP datasets.

From Fig. 4, it is observed that the performance of the method with CNN block is better than that of the method with Trm block. It shows that the local features extracted by convolution are more useful than the global features extracted by transformer for HSI classification in our model. Furthermore, the OAs of the method with CT block is higher than those of the method with CNN block, which indicates that the CT block in the proposed method effectively combines local and global features and improves the performance of HSI classification.

IV. CONCLUSION

In this letter, we proposed a novel method named CTN for HSI classification. The proposed method obtained spatial features via CPE and extracted local-global features via CT block, which was demonstrated to be effective for HSI classification. Compared with the state-of-the-art HSI classification methods, the proposed method performs better on classification accuracy. However, its good performance relied on many labeled samples. The future research is to further improve its performance by the means of semisupervised mode when the labeled samples are limited.

REFERENCES

[1] M. E. Paoletti, J. M. Haut, J. Plaza, and A. Plaza, "Deep learning classifiers for hyperspectral imaging: A review," *ISPRS J. Photogramm. Remote Sens.*, vol. 158, pp. 279–317, Dec. 2019.
[2] L. He, J. Li, C. Liu, and S. Li, "Recent advances on spectral-spatial hyperspectral image classification: An overview and new guidelines," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 3, pp. 1579–1597, Mar. 2017.
[3] D. Hong *et al.*, "More diverse means better: Multimodal deep learning meets remote-sensing imagery classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 5, pp. 4340–4354, Apr. 2021.
[4] A. Sellami and S. Tabbone, "Deep neural networks-based relevant latent representation learning for hyperspectral image classification," *Pattern Recognit.*, vol. 121, Jan. 2022, Art. no. 108224.
[5] F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 8, pp. 1778–1790, Aug. 2004.
[6] J. Ham, Y. Chen, M. M. Crawford, and J. Ghosh, "Investigation of the random forest framework for classification of hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 3, pp. 492–501, Mar. 2005.
[7] C. Cariou and K. Chehdi, "Unsupervised nearest neighbors clustering with application to hyperspectral images," *IEEE J. Sel. Topics Signal Process.*, vol. 9, no. 6, pp. 1105–1116, Sep. 2015.
[8] Y. Chen, N. M. Nasrabadi, and T. D. Tran, "Hyperspectral image classification using dictionary-based sparse representation," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 10, pp. 3973–3985, Oct. 2011.
[9] D. Hong, L. Gao, J. Yao, B. Zhang, A. Plaza, and J. Chanussot, "Graph convolutional networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 7, pp. 5966–5978, Jul. 2021.
[10] W. Hu, Y. Huang, L. Wei, F. Zhang, and H. Li, "Deep convolutional neural networks for hyperspectral image classification," *J. Sensors*, vol. 2015, pp. 1–12, Jan. 2015.
[11] W. Song, S. Li, L. Fang, and T. Lu, "Hyperspectral image classification with deep feature fusion network," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 6, pp. 3173–3184, Jun. 2018.
[12] H. Patel and K. P. Upla, "A shallow network for hyperspectral image classification using an autoencoder with convolutional neural network," *Multimedia Tools Appl.*, vol. 81, no. 1, pp. 695–714, 2021.
[13] M. Ahmad, A. M. Khan, M. Mazzara, S. Distefano, M. Ali, and M. S. Sarfraz, "A fast and compact 3-D CNN for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.
[14] Z. Zhong, J. Li, Z. Luo, and M. Chapman, "Spectral-spatial residual network for hyperspectral image classification: A 3D deep learning framework," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 2, pp. 847–858, Feb. 2018.
[15] J. He, L. Zhao, H. Yang, M. Zhang, and W. Li, "HSI-BERT: Hyperspectral image classification using the bidirectional encoder representation from transformers," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 1, pp. 165–178, Jan. 2020.
[16] C. Raffel *et al.*, "Exploring the limits of transfer learning with a unified text-to-text transformer," *J. Mach. Learn. Res.*, vol. 21, no. 140, pp. 1–67, 2020.
[17] A. Dosovitskiy *et al.*, "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.
[18] H. Wu *et al.*, "CvT: Introducing convolutions to vision transformers," 2021, *arXiv:2103.15808*.
[19] D. Hendrycks and K. Gimpel, "Gaussian error linear units (GELUs)," 2016, *arXiv:1606.08415*.
[20] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016, *arXiv:1607.06450*.
[21] M. Lin, Q. Chen, and S. Yan, "Network in network," 2013, *arXiv:1312.4400*.
[22] Z. Li *et al.*, "Hyperspectral image classification with multiattention fusion network," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.
[23] A. Vaswani *et al.*, "Attention is all you need," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, Red Hook, NY, USA: Curran Associates, 2017, pp. 6000–6010.