IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 25 February 2024, accepted 9 April 2024, date of publication 18 April 2024, date of current version 26 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3390558

RESEARCH ARTICLE

Attention-Based Deep Convolutional Capsule Network for Hyperspectral Image Classification

ZHANG XIAOXIA[1,2] **AND ZHANG XIA**[3]

[1] School of Software Engineering, Chengdu University of Information Technology, Chengdu 610225, China
[2] Sichuan Province Engineering Technology Research Center of Support Software of Informatization Application, Chengdu 610225, China
[3] Institute of Intelligent Manufacturing Technology, Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing 400700, China

Corresponding author: Zhang Xiaoxia (zhangxiaoxia158@163.com)

ABSTRACT Hyperspectral remote sensing image analysis employing deep learning (DL) models has consistently demonstrated remarkable performance, owing to their robust nonlinear modeling and end-to-end optimization capabilities. Notably, the capsule neural network (CapsNet) has attracted substantial attention for its proficient feature extraction capabilities. However, it tends to overlook the inherent spatial heterogeneity within patch features. In this paper, we introduce a spatial attention-based deep convolutional capsule network (SA-CapsNet) to enhance CapsNet's performance in hyperspectral image (HSI) classification. The incorporation of a more potent and light-weight spatial attention mechanism introduces diversity among neighboring pixels. Additionally, we enhance the stability of learned spectral-spatial features by implementing a convolutional capsule layer that extends dynamic routing with 3D convolution. Experimental results conducted on three commonly used hyperspectral datasets demonstrate that SA-CapsNet outperforms conventional and state-of-the-art DL-based HSI classification algorithms in terms of classification accuracy and computational efficiency.

INDEX TERMS Spatial attention, capsule neural network, deep learning, hyperspectral image classification.

I. INTRODUCTION

Hyperspectral images (HSIs) exhibit an exceptionally high spectral resolution, comprising hundreds of narrow continuous wavelength bands that encompass the electromagnetic spectrum. This wealth of spectral data facilitates precise discrimination among similar materials of interest. Thus, the classification of HSIs has surged in popularity within the field of remote sensing and has found application in diverse domains, including scene recognition, precision agriculture, and land monitoring, among others.

The progress in hyperspectral image (HSI) classification has greatly benefited from the application of advanced machine learning and pattern recognition techniques. Deep learning (DL), known for its remarkable capacity to represent spectral-spatial properties, can automatically generate data-adaptive high-level features. This approach has been extensively employed in HSI classification scenarios. Convolutional neural networks (CNNs) [1], [2], stacked autoencoders (SAEs) [3], [4], deep belief networks (DBNs) [5], [6], and recurrent neural networks (RNNs) [7], [8] are some examples of common deep learning models applied to HSI classification. Among these DL techniques, CNNs stand out as the most popular architecture for image recognition, classification, and detection tasks, encompassing 1D CNNs, 2D CNNs, 3D CNNs, and certain hybrid variants [9]. For instance, to accurately identify hyperspectral images (HSIs) by capturing spectral properties, the use of 1D CNNs has been suggested [10]. However, 1D CNNs flatten the spatial image into a 1D vector, overlooking the spatial distribution patterns inherent in HSIs. Consequently, 2D CNNs [11], [12] have been proposed to simultaneously extract spatial and spectral information and reduce the dimensionality of original HSI data domain, considering the abundant spectral

The associate editor coordinating the review of this manuscript and approving it for publication was Krishna Kant Singh.

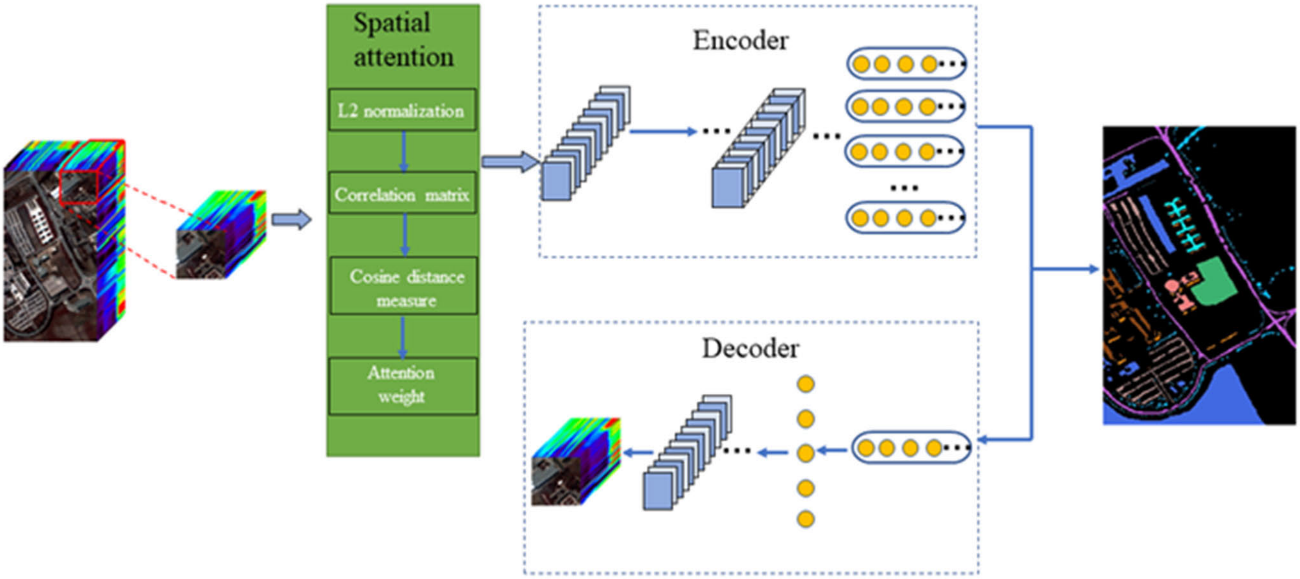

FIGURE 1. Architectural of the proposed SA-CapsNet.

information in HSIs. Nevertheless, it is worth noting that when employing 2D CNNs, the spectral-spatial integral features of hyperspectral imaging, which manifest as a cube of data, may be compromised. To fully exploit the spectral-spatial information within HSIs, 3D CNNs [13], [14], [15] have been developed. These networks simultaneously extract spectral-spatial features from HSI data, thereby enhancing classification performance and offering a promising approach for managing hyperspectral data cubes. Furthermore, various efficient methods, as documented in studies such as those by Paoletti et al. [16], Zhong et al. [17], and Tang et al. [18], have been integrated with CNNs to enhance the effectiveness of HSI classification.

Despite the impressive effectiveness of the current CNN-based HSIs classification algorithms, some limitations may still be witnessed. On the one hand, max pooling is frequently used by CNNs to lower computation costs and improve feature invariance, allowing it to collect more discriminative properties while losing the correlations be-tween the attributes of geographic objects. On the other hand, the scalar value employed in CNNs to represent features has inferior representational capacity because of the intricacy of HSIs. CapsNet, a unique deep learning model, that employs dynamic routing-by-agreement and vector-output capsules to boost the model's feature representation capability, and encapsulate the correlations between various features, has previously been suggested to enhance CNN's performance [19]. In the CapsNet, a cluster of neurons serves as a capsule in place of a neuron in the conventional neural network. The capsule is a vector that represents internal properties that can be used to acquire part-to-whole relationships between distinct objects. It can overcome the issue whereby fully connected layers in standard neural networks are unable to accurately capture the hierarchical structure to maintain the spatial information. In the previous works, a modified two-layer CapsNet with few training samples was presented by Deng et al. [20] for HSIs interpretation. To extract more discriminant information, a five-layer supervised deep CapsNet architecture was created for HSI classification [21]. Furthermore, a novel and fast D-CapsNet [22] was used to obtain the richer and more reliable features for scene classification. These researches demonstrate that CapsNets have better representations and improve classification performance.

As well, the research of HSIs classification has studied CNN with attention mechanisms extensively, including spatial attention [23], spectral attention [24], and spatial-spectral joint attention [25], [26]. To capture non-local spectral-spatial characteristics, Lei et al. [27], developed a non-local CapsNet coupled attention methodology with CapsNet. Early research has shown that the attention mechanism might enable deep models to emphasize more prominent characteristics while suppressing those that are less beneficial. Despite the great results obtained by CNN with attention mechanism, these networks are affected by a major overfitting problem due to the large number of parameters to be trained and the cost of resources spent on redundant features. Therefore, a convolutional capsule network based on a light-weight spatial attention mechanism (SA-CapsNet) is proposed to acquire the more robust and useful information for HSIs classification, improving the data representation by refining initial convolutional features will produce better features. First, to model discriminative and representative characteristics, a lighter spatial attention operator is designed before the initial convolution. Then, to improve the classification performance, a dynamic routing based on 3D convolution is implemented. Finally, with only a small number of training samples, the proposed model achieves satisfactory classification outcomes on several widely-used HSI datasets.

TABLE 1. Primary architecture of SA-CapsNet.

Spatial attention	L2 normalization	Correlation matrix	Cosine distance measure	Attention weight	—
		Kernel size	Stride	Batch normalization	Activation
Encoder	Convolution	3×3×64	(1,1)	√	Relu
	Convolution	3×3×32	(1,1)	√	Relu
	Primary capsule	3×3×16×8	(2,2)	√	Relu+Squash
	3D capsule	3×3×8×8×8	(2,2,8)	×	Squash
	Dense capsule	—	16×c	—	Squash
Decoder	Fully connected	7×7×16	—	√	Relu
	Deconvolution	3×3×64	(1,1)	×	Relu
	Deconvolution	3×3×32	(1,1)	×	Relu
	Deconvolution	3×3×16	(1,1)	×	Relu
	Deconvolution	3×3×c	(1,1)	×	Relu

The rest of this paper is organized as follows. The architecture of the SA-CapsNet is described in Section II, the findings of the experiments and discussions are presented in Part III, and the paper is concluded in Section IV.

II. RELATED WORK

Suppose $\mathbf{X} \in \mathbb{R}^{M \times B}$ be an HSI with M total pixels, B bands, and c classes. The proposed framework SA-CapsNet includes a spatial attention operator, an encoder and a decoder. The encoder is composed of two convolution operators, a primary capsule layer, a 3D convolutional capsule layer and an output layer. The decoder module consists of a full-connected layer and several deconvolution layers. The simple SA- CapsNet architecture is shown in Fig. 1. Table 1 displays the primary architecture of the SA-CapsNet for HSIs classification.

A. SPATIAL ATTENTION

In the field of HSI classification, CNNs with spatial attention mechanism have been extensively researched and proved to be effective. For our spatial attention technique, we employ a correlation matrix and a trainable cosine distance function to assign varying weights to different pixels. To elaborate, we initially construct patch features that represent spatial information using a squared moving window. To mitigate scale dependence in the input patch features, we apply L2 normalization to obtain X'. Subsequently, we utilize an attention mask based on the correlation matrix of nearby pixels.

$$\mathbf{F} = \mathbf{X}'\mathbf{X}'^{\top} \tag{1}$$

where $\mathbf{X}' \in \mathbb{R}^{M' \times B}$, M' equals $p \times p$, which represents the number of adjacent pixels, with p signifying the window size. $\mathbf{F}$ represents the correlation matrix of pairwise neighboring pixels. Subsequently, we establish a trainable cosine distance function to evaluate the similarity of each surrounding pixel to the central pixel.

$$\alpha_i = \frac{\mathbf{F}_i \Lambda \mathbf{F}_0^{\top}}{\|\mathbf{F}_i\| \left\|\mathbf{F}_0^{\top}\right\|} \tag{2}$$

where, $i \in \{1, 2, \ldots, M'\}$, the central pixel vector in the neighborhood is denoted as F_0, and F_i represents an arbitrary surrounding pixel vector. The learnable parameter α is obtained by adding the diagonal matrix $\Lambda \in \mathbb{R}^{M' \times M'}$. To ensure improved convergence, the attention weights are subsequently normalized to have a unit sum through the application of a softmax function.

$$w_i = \frac{e^{\alpha_i + b_i}}{\sum\limits_{j=1}^{M} e^{\alpha_j + b_j}} \tag{3}$$

where $\sum w_i = 1$, b represents the bias, and a diagonal weight matrix $\mathbf{W} \in \mathbb{R}^{M' \times M'}$based on w_i, can be established. This lightweight approach incorporates the variability of surrounding pixels when representing features by employing a data-adaptive and learnable weighting technique. Notably, pixels with larger weights, in contrast to those with smaller weights, carry greater importance and exert a more significant influence.

B. CAPSNET BASED ON 3D CONVOLUTION DYNAMIC ROUTING

To counteract the adverse consequences of stacking capsule layers, CapsNet employs an extended form of dynamic routing founded on 3D convolution. This modification enables the construction of a potent deep convolutional capsule network for HSI classification. Consequently, the model can generate more dependable classification outcomes even when faced with limited training data, thanks to its ability to aggregate more robust higher-level feature information. The output $\mathbf{\Phi}^k \in \mathbf{R}^{(w^k, w^k, c^k, n^k)}$ from capsule layer k was first transformed into a 3D tensor, followed by reconstruction using 3D

TABLE 2. Summary of the number of each class on the IP dataset.

Number	Color	Land cover type	Samples
1	🟦	alfalfa	46
2	🟧	corn-notill	1428
3	🟦	corn-mintill	830
4	🟦	corn	237
5	🟧	grass-pasture	483
6	🟨	grass-trees	730
7	🟨	grass-pasture-mowed	28
8	🟩	hay-windrowed	478
9	🟩	oats	20
10	🟪	soybean-notill	972
11	⬜	soybean-mintill	2455
12	🟦	soybean-clean	593
13	🟥	wheat	205
14	🟩	woods	1265
15	🟪	building-grass-trees	386
16	🟦	stone-steel-towers	98
		Total	10,249

TABLE 3. Summary of the number of each class on the KSC dataset.

Number	Color	Land cover type	Samples
1	🟦	Scrub	761
2	🟧	willow swamp	242
3	🟦	wabbage palm hammock	257
4	🟦	cabbage palm/oak hammock	252
5	🟧	slash pine	161
6	🟨	oak/broadleaf hammock	229
7	🟨	hardwood swamp	106
8	🟩	graminoid marsh	431
9	🟩	spartina marsh	520
10	🟪	cattail marsh	403
11	⬜	salt marsh	419
12	🟦	mud flats	502
13	🟥	water	928
		Total	5,211

convolution kernels. Finally, a reshape operation was applied to obtain the prediction **Y**.

$$\mathbf{Y} = \text{Reshape}\left(\text{Conv3D}\left(\text{Reshape}\left(\boldsymbol{\Phi}^{k}\right)\right)\right) \tag{4}$$

Each capsule tensor s within layer k generates a prediction denoted as $\mathbf{Y}s$. Consequently, a 3D variant of the softmax function can be employed to calculate the coupling coefficients $\mathbf{H}s$ for the predictions for all s.

$$\mathbf{H}_s = \text{soft}\max\left(\mathbf{B}_s\right) \tag{5}$$

TABLE 4. Summary of the number of each class on the PU dataset.

Number	Color	Land cover type	Samples
1	🟦	brocoli_green_weeds1	2009
2	🟧	brocoli_green_weeds2	3726
3	🟦	fallow	1976
4	🟦	fallow_rough_plow	1394
5	🟧	fallow_smooth	2678
6	🟨	stubble	3959
7	🟨	celery	3579
8	🟩	grapes_untraine	21,414
9	🟩	soil_vinyard_develop	6203
10	🟪	corn_senesced_weeds	3278
11	⬜	lettuce_romaine_4wk	1068
12	🟦	lettuce_romaine_5wk	1927
13	🟥	lettuce_romaine_6wk	916
14	🟩	lettuce_romaine_7wk	1070
15	🟪	vineyard_untrained	7268
16	🟦	vineyard_vertical_trells	1807
		Total	54,129

$$h_{pqrs} = \frac{\exp\left(b_{pqrs}\right)}{\sum_{x}\sum_{y}\sum_{z}\exp\left(b_{xyzs}\right)} \tag{6}$$

At position (p, q, r) within $\mathbf{H}s$ and $\mathbf{B}s$, h_{pqrs} and b_{pqrs} represent the coupling coefficient and logit, respectively. The total input of the capsule, S_{pqr}, in layer $(k+1)$ is computed as a weighted sum of all prediction vectors Y_{pqrs}. Subsequently, applying a squash function yields the capsule's output, V_{pqr}, in layer $(k + 1)$. The logit b_{pqr} is updated by assessing the compatibility between V_{pqr} and Y_{pqr}. The output of capsule layer $(k+1)$, denoted as $\boldsymbol{\Phi}^{k+1}$, is generated by utilizing V_{pqr}.

$$S_{pqr} = \sum_{s} h_{pqrs} \cdot Y_{pqrs} \tag{7}$$

$$V_{pqr} = \frac{\left\|S_{pqr}\right\|^2}{1 + \left\|S_{pqr}\right\|^2} \cdot \frac{S_{pqr}}{\left\|S_{pqr}\right\|} \tag{8}$$

$$b_{pqrs} \leftarrow b_{pqrs} + V_{pqr} \cdot Y_{pqrs} \tag{9}$$

Apart from capturing part-to-whole data correlations within the target geographic object, the 3D convolution-based extension of dynamic routing also extracts spectral-spatial features from the input feature maps. The loss function of the proposed framework is a combination of margin loss L_m and reconstruction loss L_r.

$$\begin{aligned} L &= L_m + \lambda L_r \qquad (10) \\ L_m &= T_c \max\left(0, m^+ - \|\mathbf{v}_c\|\right)^2 \end{aligned}$$

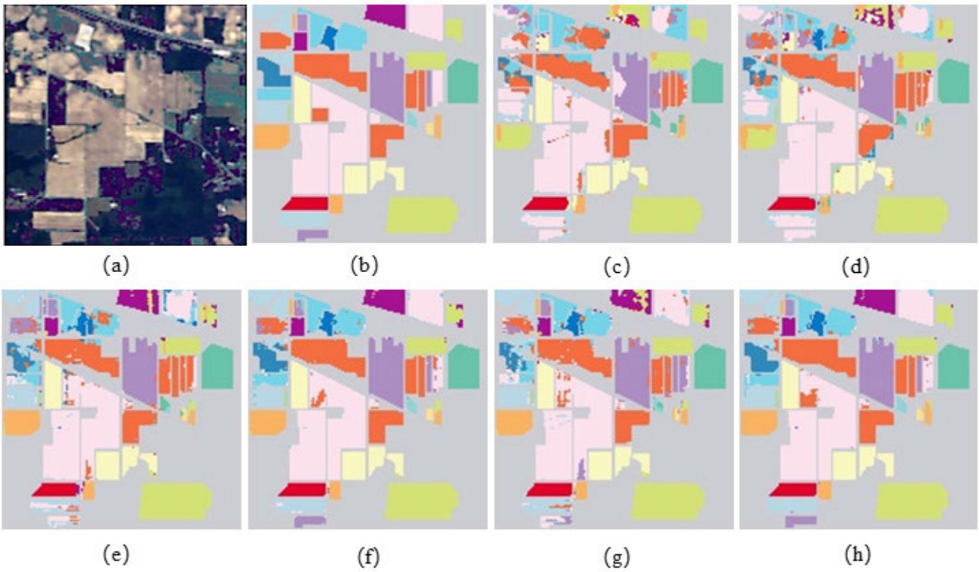

FIGURE 2. Classification maps for the IP dataset: (a) false color image, (b) ground truth map, (c)-(h) corresponding to SVM, 2D-CNN, 3D-CNN, RSSAN, 3D-CapsNet and SA-CapsNet, respectively.

$$+ \lambda_1 (1 - T_c) \max \left(0, \|\mathbf{v}_c\| - m^-\right)^2 \quad (11)$$

$$Lr = \|\mathbf{X} - \mathbf{X}_r\| \quad (12)$$

where, λ serves as a balancing coefficient, ensuring proper control over the reconstruction loss. The variable c denotes the classification category, and λ_1 can be empirically adjusted to 0.5. When class c is present in the image, T_c is assigned the value 1; otherwise, it is changed to 0. Furthermore, m^+ represents the lower threshold for correct classification, while m^- denotes the upper threshold for misclassification. When $\|v_c\| \in \left[m^+, 1\right]$, the current input image is attributed to class c. Conversely, when $\|v_c\| \in \left[0, m^-\right]$, the current input image is considered non-membership to class c. For this study, m^+ and m^- were configured at 0.9 and 0.1, respectively. Moreover, $\|\mathbf{v}_c\|$ represents the length of the activity vector.

III. EXPERIMENTAL RESULTS

A. HYPERSPECTRAL DATASETS

The first hyperspectral dataset used in our research was acquired by the Aerial Visible/Infrared Imaging Spectrometer (AVIRIS) sensor above the Indian Pines test site in northwest Indiana. This image consists of 145 by 145 pixels and comprises 220 bands within a spectral range of 0.2 to 2.4 m, offering a spatial resolution of 20 m, and it depicts an agricultural scene. After removing 20 water absorption bands, 200 bands were retained for further analysis.

The second hyperspectral dataset was obtained via the AVIRIS sensor flying over Kennedy Space Center (KSC) in Florida at an altitude of approximately 20 km, featuring a spatial resolution of 18 m. After filtering out bands with suboptimal signal-to-noise ratios and water absorption, the dataset retained 176 bands for the analysis. The dataset is associated with 13 defined classes used for the site classification.

The third hyperspectral dataset was gathered by the AVIRIS sensor above the city of Salinas Valley in California, USA. The image encompasses dimensions of 512×217 pixels and exhibits a remarkable spatial resolution of 3.7 m. It contains 224 wavelength bands, with 204 spectral bands reserved for the experiment after excluding water ab-sorption bands within the ranges of 108-112, 154-167, and 224.

B. EXPERIMENTAL SETTINGS

Regarding the hyperparameters, the batch size is set as 32. To avoid overfitting, we employ the early stopping strategy and dynamic learning rate adjustment. If the validation loss shows no improvement after 40 epochs, the training process is terminated, with a maximum limit of 200 training epochs. The initial learning rate is set at 0.001. If, after 10 epochs, the validation loss ceases to decline, the learning rate is halved successively until it reaches zero or the training process is completed. For the spatial attention, the diagonal matrix elements are initialized to one, while the bias term b is initialized to zero. In our experiments, we randomly select 4% of the training samples from the IP and KSC datasets, and 1% from the PU dataset. The number of validation samples equals the size of the training set. The remaining data are reserved for evaluating the proposed model's performance. To ensure a fair comparison, the compared methods adhere to the same ratio of samples. We consider input patches of a spatial size of 13 × 13 pixels for the proposed method.

C. COMPARISON OF CLASSIFICATION PERFORMANCE

Our first experiment aims to assess the performance of the proposed method by comparing it with well-established HSI classification approaches from the existing literature. Tables 5-7 offer a quantitative evaluation of classification accuracy by utilizing the IP, KSC, and PU datasets. The

TABLE 5. Classification results of different models on IP dataset.

method class	SVM	2D-CNN	3D-CNN	RSSAN	3D-CapsNet	SA-CapsNet
1	54.91±40.12	70.09±27.64	90.46±9.83	**100**±0	98..72±1.81	96.69±2.86
2	62.03±4.83	73.18±0.85	75.87±2.40	92.40±2.53	85.40±3.43	**95.20**±1.39
3	54.21±5.04	68.18±13.06	82.62±5.26	94.20±1.53	86.00±3.80	**98.40**±0.64
4	70.70±9.43	72.60±15.51	80.87±11.75	92.93±2.73	86.02±9.29	**96.11**±2.82
5	68.91±4.39	78.32±8.07	93.79±0.78	97.73±1.13	94.20±4.04	**97.85**±1.58
6	85.58±4.15	88.05±3.39	94.62±6.25	97.79±3.01	99.39±0.31	**99.78**±0.08
7	52.05±37.96	60.32±43.35	88.31±8.95	81.76±25.79	89.1±10.45	**97.10**±4.10
8	91.81±4.96	93.06±3.69	95.12±1.63	**99.92**±0.11	97.88±1.53	99.67±0.47
9	31.81±22.78	72.81±28.94	68.22±20.15	**94.74**±4.30	90.24±7.67	94.12±8.32
10	67.70±2.67	77.81±6.15	81.51±1.16	92.96±1.12	83.93±2.97	**96.81**±0.68
11	71.81±6.29	79.64±5.95	87.03±1.72	96.03±0.37	90.26±0.67	**98.76**±0.49
12	50.54±16.69	56.22±4.31	64.79±2.74	85.04±6.62	74.25±4.10	**92.52**±2.69
13	86.27±8.17	90.83±5.29	96.89±4.40	**99.26**±0.24	97.74±2.42	98.17±2.59
14	79.62±10.08	85.08±6.71	94.74±2.50	97.93±0.87	94.77±2.43	**98.84**±0.30
15	87.76±10.86	79.03±11.49	78.77±13.93	90.48±1.17	90.46±3.85	**97.20**±1.47
16	60.42±42.91	84.02±6.20	74.16±12.50	87.75±6.79	89.66±4.28	**89.89**±4.63
OA	71.00±5.28	78.42±3.99	84.96±1.67	94.60±0.90	89.43±0.86	**97.49**±0.36
AA	67.26±11.14	76.83±6.73	84.23±2.59	93.81±1.52	90.50±1.95	**96.69**±0.94
Kappa×100	66.50±6.22	75.19±4.64	82.83±1.89	93.84±1.03	87.93±0.97	**97.14**±0.41

TABLE 6. Classification results of different models on the KSC dataset.

method class	SVM	2D-CNN	3D-CNN	RSSAN	3D-CapsNet	SA-CapsNet
1	84.15±10.25	98.17±1.07	98.38±0.75	98.79±1.24	99.23±0.65	**99.90**±0.07
2	77.48±6.91	80.14±4.39	81.33±5.39	93.27±2.89	88.26±1.60	**95.88**±3.00
3	45.03±9.02	49.28±3.31	59.25±9.19	74.50±5.46	65.33±21.30	**91.40**±8.05
4	50±40.82	49.99±6.09	52.10±1.58	69.32±5.90	57.02±14.07	**86.23**±7.83
5	49.74±40.83	53.3±41.09	57.6±42.24	57.7±14.75	59.60±14.02	**80.77**±4.83
6	42.23±11.30	79.6±14.33	73.9±17.29	91.75±3.52	76.77±6.93	**99.36**±0.45
7	48.50±34.53	73.38±8.92	61.76±5.83	76.30±5.19	58.61±8.00	**94.19**±7.28
8	56.39±19.16	64.2±12.82	78.1±13.84	94.47±3.25	77.49±5.37	**97.42**±2.39
9	70.43±0.95	70.91±2.33	71.72±2.06	97.01±3.09	80.93±7.83	**98.55**±1.20
10	76.73±25.85	95.29±6.25	97.59±3.22	98.52±2.10	97.91±2.96	**100**±0
11	88.60±6.86	98.46±0.98	98.26±1.60	97.98±2.14	**98.49**±1.10	95.46±2.81
12	97.07±2.23	80.02±7.88	85.14±7.92	98.70±0.45	95.04±4.87	**99.53**±0.66
13	91.50±4.57	96.96±0.76	97.10±0.47	99.93±0.11	99.49±0.56	**100**±0
OA(%)	75.60±7.24	81.97±2.04	83.83±1.62	93.48±0.64	86.67±2.06	**97.10**±0.74
AA(%)	67.53±5.31	76.15±4.57	77.88±4.44	88.33±2.05	81.09±3.40	**95.28**±1.07
Kappa×100	72.65±8.22	79.89±2.29	81.97±1.81	92.74±0.71	85.16±2.29	**96.76**±0.83

assessment encompasses several classifiers, such as SVM, 2D-CNN, 3D-CNN, RSSAN, 3D-CapsNet, and our proposed approach SA-CapsNet. Rows in Tables 5-7 present classification outcomes and overall metrics, while the classes are listed in columns. Moreover, the classification maps corresponding to the tests listed in tables are shown in Figs. 2-4 for illustrative purposes. Additionally, each table displays the average and standard deviation values obtained from three iterations.

To thoroughly assess the superiority of our proposed method, we present quantitative results for IP, KSC, and PU datasets, including performance metrics for each class, Overall Accuracy (OA), Average Accuracy (AA), and Kappa×100, in Tables 5-7. It is evident that SA-CapsNet outperforms all other methods, demonstrating the highest classification accuracy across all three metrics, with the most notable improvements observed in the IP and PU datasets. Take Table 5 as an example for analysis. First, SA-CapsNet achieves a remarkable increase in OA compared to SVM, 2D-CNN, 3D-CNN, RSSAN, and 3D-CapsNet, with improvements of 26.49%, 19.07%, 12.53%, 2.89%, and 8.06%, respectively, which are unexpected gains in accuracy. Second, SA-CapsNet achieves an OA above 90% for all

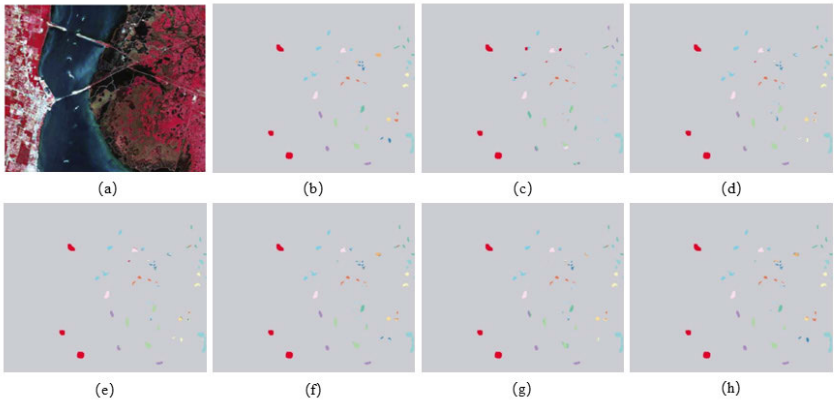

FIGURE 3. Classification maps for the KSC dataset: (a) false color image, (b) ground truth map, (c)-(h) corresponding to SVM, 2D-CNN, 3D-CNN, RSSAN, 3D-CapsNet and SA-CapsNet, respectively.

TABLE 7. Classification results of different models on the PU dataset.

method class	SVM	2D-CNN	3D-CNN	RSSAN	3D-CapsNet	SA-CapsNet
1	82.09±1.26	79.81±1.83	87.11±2.40	93.69±2.86	86.29±6.54	**97.18**±0.77
2	90.02±3.26	88.20±3.25	92.44±0.97	98.23±0.21	96.01±1.25	**99.26**±0.08
3	36.82±4.60	46.46±3.38	56.50±4.90	87.39±9.23	82.75±11.14	**96.35**±1.38
4	85.7±14.99	98.37±0.61	97.57±1.62	98.04±1.29	98.05±1.46	**98.99**±0.66
5	91.54±9.94	97.34±0.50	96.22±4.77	93.91±5.10	98.43±1.27	**99.35**±0.71
6	58.79±9.18	68.66±5.62	75.8±12.56	95.88±1.10	89.92±4.51	**98.98**±0.63
7	85.2±10.54	66.6±47.14	65.85±5.58	93.01±2.32	84.68±5.18	**98.10**±1.18
8	61.96±2.96	73.64±4.46	69.12±5.22	88.09±5.36	82.06±5.86	**93.36**±0.76
9	81.6±15.19	96.58±1.01	96.49±1.21	95.67±2.40	97.55±0.30	**98.76**±0.58
OA	77.74±3.41	82.44±1.71	85.25±1.20	95.42±0.75	91.62±1.50	**98.18**±0.08
AA	74.88±2.88	79.52±6.29	81.90±1.39	93.77±1.04	90.64±2.32	**97.82**±0.26
Kappa×100	71.00±3.99	76.34±2.41	80.44±1.42	93.91±1.01	88.82±1.99	**97.58**±0.10

11 classes in the KSC dataset, especially performing well in classes with fewer samples, such as ‘‘slash pine’’ and ‘‘hardwood swamp’’, even some classes have a classification accuracy of more than 99%. As for the IP and KSC datasets in general, it can be noticed that 2D-CNN and 3D-CNN suffer from significant accuracy deterioration. The possible reason is that these methods overly focus on the spatial features, while the spatial resolution of these datasets is relatively low. Moreover, our proposed approach consistently demonstrates superior performance in these two datasets.

When assessing the three metrics applied to the comparative approaches, it becomes evident that the two deep learning methods, 2D-CNN and 3D-CNN, exhibit higher classification accuracies in comparison to the classical SVM, primarily attributed to their utilization of spatial context characteristics. However, it’s worth noting that 2D-CNN and 3D-CNN may underperform SVM in specific classes due to an inadequate amount of data for effective network training. Notably, 3D-CNN, 3D-CapsNet, and SA-CapsNet outperform 2D-CNN, primarily because they fully exploit deep spectral-spatial features. Among these, SA-CapsNet demonstrates the most robust performance across all datasets, capitalizing on the exploration and utilization of potent 3D spectral-spatial features.

Figures 2-4 depict the classification maps generated by the algorithms across the three datasets, as well as false color images and ground truth maps. As illustrated in Figs. 2-4, SVM and 2D-CNN exhibit noticeable classification noise within the land covers. While RSSAN and 3D-CapsNet can reduce spot-like misclassification, they cannot differentiate feature boundaries and disregard some smaller features, such as ‘‘soy-bean-mintill’’ and ‘‘grass-pasture-mowed’’ in Fig. 2. As for 3D-CNN, it also exhibits significant misclassification at the edges and fails to capture numerous morphological characteristics of ground features. In contrast, our proposed SA-CapsNet minimizes internal noise while preserving intricate details of ground objects, including ‘‘cattail marsh’’ and ‘‘water’’ that are difficult to identify in Fig. 3. By examining these images, SA-CapsNet can extract class-oriented information for the image patch. This yields lower attention weights for pixels whose class labels differ from the central pixels and higher attention weights for neighboring pixels that belong to the central pixel. SA-CapsNet represents class-based discriminatory information in this circumstance,

TABLE 8. Computation efficiency for different models on the three datasets.

Dataset		SVM	2D CNN	3D-CNN	RSSAN	3D-CapsNet	SA-CapsNet
IP	Parameters	—	**378,116**	2,401,756	151,420	30,745,728	480,674
	Execution time (s)	**22.19**	77.42	124.79	200.07	288.45	163.33
KSC	Parameters	—	**377,914**	2,087,553	150,692	23,366,784	384,586
	Execution time (s)	**6.86**	45.47	76.42	117.76	116.63	82.70
PU	Parameters	—	377,409	832,349	150,008	1,436,800	**369,609**
	Execution time (s)	**13.56**	59.10	109.84	123.65	173.06	99.29

TABLE 9. Overall accuracy (%) for the contrast model on the three datasets.

	IP	KSC	PU
no spatial attention	95.34±0.86	96.18±1.25	96.27±0.20
proposed	**97.49**±0.36	**97.10**±0.74	**98.18**±0.08

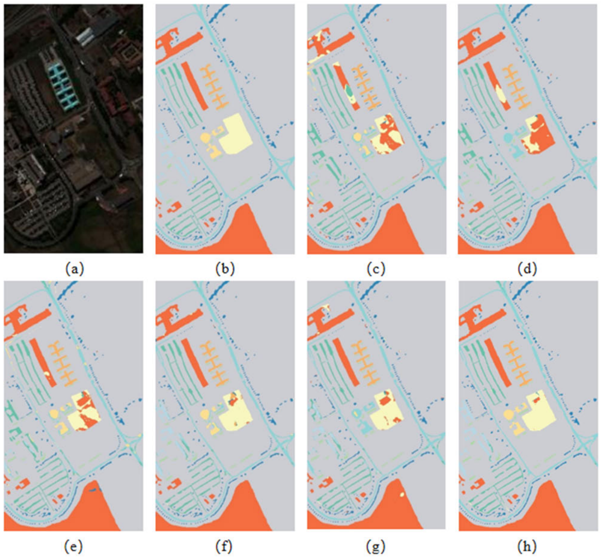

FIGURE 4. Classification maps for the PU dataset: (a) false color image, (b) ground truth map, (c)-(h) corresponding to SVM, 2D-CNN, 3D-CNN, RSSAN, 3D-CapsNet and SA-CapsNet, respectively.

thus consistently producing classification maps that closely resemble the distribution of actual ground objects.

D. COMPARISON OF COMPUTATION EFFICIENCY

We further assess the computational complexity of various methods in terms of their running time (in seconds) and parameter size. As presented in Table 8, in comparison to other deep learning approaches like RSSAN and 3D-CapsNet, SA-CapsNet exhibits a substantial enhancement in training time.

For evaluating the efficiency of the proposed method, we count the execution time and parameters for the proposed method and other comparison methods on the three datasets. It could be seen that the computation cost of SVM is far less than that of deep learning models. There are too many parameters of 3D-CapsNet, so it has the longest training time for almost all three datasets. Among all the state-of-the-art CNN-based methods, SA-CapsNet works well with fewer parameters than 3D-CNN and 2D-CNN on the PU dataset. Furthermore, the number of parameters of the proposed method is more than RSSAN. However, the execution time of SA-CapsNet is not as high as RSSAN's on the three datasets. The efficiency of SA-CapsNet outperforms 3D-CapsNet and RSSAN, owing to the highly efficient straightforward spatial attention and some normalization mechanisms, which can effectively improve the convergence level. Compared with 3D-CNN, SA-CapsNet converges fast on the PU dataset, but not on the IP and KSC dataset. It may attribute to the fact that some classes in IP and KSC images have similar property, which may cause the model to need more training time to learn the differences between these classes.

E. EFFECTIVENESS OF SPATIAL ATTENTION

In the design of the proposed architecture, the spatial attention is employed to promote the model to emphasize more relevant features but suppress less informative ones. To validate of the positive impact on classification results, we conducted a comparative experiment. Specifically, we introduced a new architecture by removing the attention mechanism from SA-CapsNet. To ensure a fair comparison, we utilized input HSI patches with a spatial size of 13 × 13, and maintained the same training ratio as in the previous experiments. The results of the comparison between adopting the proposed spatial attention and not using spatial attention are presented in Table 9.

Obviously, the proposed network SA-CapsNet significantly enhances classification accuracy, with noticeable improvements even when dealing with limited training samples. In detail, with training sets comprising only 4%, 4%, and 1% for the IP, KSC, and PU datasets, respectively, the accuracy of the proposed network with spatial attention reached 97.49%, 97.10%, and 98.18%, receptively, marking an improvement of 2.15 %, 0.92 %, and 1.91% compared to the model without spatial attention. This phenomenon is more evident regarding to the India pines dataset. Accordingly, with the attention mechanism, SA-CapsNet is superior over the model without spatial attention. This improvement stems from the network's ability to efficiently learn spectral-spatial features while taking into account the spatial pixel relationships.

IV. CONCLUSION

In this study, we introduce SA-CapsNet, a novel approach for HSI classification. The key innovation of SA-CapsNet lies in its lightweight spatial attention operator, which harnesses data-adaptive attention weights to model discriminative and representative features effectively. To enhance the robustness of CapsNet, we employ dynamic routing based on 3D convolution. Our experimental results, conducted on three HSI datasets, establish the superior performance of SA-CapsNet compared to conventional and state-of-the-art deep learning-based HSI classification methods. These experiments reveal that SA-CapsNet not only offers a smaller quantity of parameters but also achieves higher accuracy, particularly when dealing with limited training samples, outperforming its counterparts. While our experimental findings have yielded positive results, our intention is to further enhance the proposed approach by incorporating a spectral attention mechanism.

REFERENCES

[1] H. Gao, Y. Yang, S. Lei, C. Li, H. Zhou, and X. Qu, "Multi-branch fusion network for hyperspectral image classification," *Knowl.-Based Syst.*, vol. 167, pp. 11–25, Mar. 2019.

[2] S. K. Roy, G. Krishna, S. R. Dubey, and B. B. Chaudhuri, "HybridSN: Exploring 3-D–2-D CNN feature hierarchy for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 2, pp. 277–281, Feb. 2020.

[3] P. Zhou, J. Han, G. Cheng, and B. Zhang, "Learning compact and discriminative stacked autoencoder for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 7, pp. 4823–4833, Jul. 2019.

[4] Y. Chen, X. Zhao, and X. Jia, "Spectral–spatial classification of hyperspectral data based on deep belief network," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 6, pp. 2381–2392, Jun. 2015.

[5] C. Li, Y. Wang, X. Zhang, H. Gao, Y. Yang, and J. Wang, "Deep belief network for spectral–spatial classification of hyperspectral remote sensor data," *Sensors*, vol. 19, no. 1, p. 204, Jan. 2019.

[6] Z. Li, H. Huang, Z. Zhang, and G. Shi, "Manifold-based multi-deep belief network for feature extraction of hyperspectral image," *Remote Sens.*, vol. 14, no. 6, p. 1484, Mar. 2022.

[7] W. Zhou, S.-I. Kamata, Z. Luo, and H. Wang, "Multiscanning strategy-based recurrent neural network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5521018.

[8] L. Mou, P. Ghamisi, and X. X. Zhu, "Deep recurrent neural networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3639–3655, Jul. 2017.

[9] J. Liu, T. Wang, A. Skidmore, Y. Sun, P. Jia, and K. Zhang, "Integrated 1D, 2D, and 3D CNNs enable robust and efficient land cover classification from hyperspectral imagery," *Remote Sens.*, vol. 15, no. 19, p. 4797, Oct. 2023.

[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. Zhao and S. Du, "Spectral–spatial feature extraction for hyperspectral image classification: A dimension reduction and deep learning approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 8, pp. 4544–4554, Aug. 2016.

[12] A. Santara, K. Mani, P. Hatwar, A. Singh, A. Garg, K. Padia, and P. Mitra, "BASS net: Band-adaptive spectral–spatial feature learning neural network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 9, pp. 5293–5301, Sep. 2017.

[13] P. Ghamisi, Y. Chen, and X. X. Zhu, "A self-improving convolution neural network for the classification of hyperspectral data," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 10, pp. 1537–1541, Oct. 2016.

[14] Y. Li, H. Zhang, and Q. Shen, "Spectral–spatial classification of hyperspectral imagery with 3D convolutional neural network," *Remote Sens.*, vol. 9, no. 1, p. 67, Jan. 2017.

[15] S. Mei, J. Ji, Y. Geng, Z. Zhang, X. Li, and Q. Du, "Unsupervised spatial–spectral feature learning by 3D convolutional autoencoder for hyperspectral classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 6808–6820, Sep. 2019.

[16] M. Paoletti, J. Haut, J. Plaza, and A. Plaza, "Deep&Dense convolutional neural network for hyperspectral image classification," *Remote Sens.*, vol. 10, no. 9, p. 1454, Sep. 2018.

[17] Z. Zhong, J. Li, Z. Luo, and M. Chapman, "Spectral–spatial residual network for hyperspectral image classification: A 3-D deep learning framework," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 2, pp. 847–858, Feb. 2018.

[18] H. Tang, Y. Li, Z. Huang, L. Zhang, and W. Xie, "Fusion of multidimensional CNN and handcrafted features for small-sample hyperspectral image classification," *Remote Sens.*, vol. 14, no. 15, p. 3796, Aug. 2022.

[19] M. Shi, R. Wang, and J. Ren, "Hierarchical capsule network for hyperspectral image classification," *Neural Comput. Appl.*, vol. 35, no. 25, pp. 18417–18443, Sep. 2023.

[20] F. Deng, S. Pu, X. Chen, Y. Shi, T. Yuan, and S. Pu, "Hyperspectral image classification with capsule network using limited training samples," *Sensors*, vol. 18, no. 9, p. 3153, Sep. 2018.

[21] M. E. Paoletti, J. M. Haut, R. Fernandez-Beltran, J. Plaza, A. Plaza, J. Li, and F. Pla, "Capsule networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 4, pp. 2145–2160, Apr. 2019.

[22] A. Raza, H. Huo, S. Sirajuddin, and T. Fang, "Diverse capsules network combining multiconvolutional layers for remote sensing image scene classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 5297–5313, 2020.

[23] J. M. Haut, M. E. Paoletti, J. Plaza, A. Plaza, and J. Li, "Visual attention-driven hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 10, pp. 8065–8080, Oct. 2019.

[24] J. Wang, J. Zhou, and W. Huang, "Attend in bands: Hyperspectral band weighting and selection for image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 12, pp. 4712–4727, Dec. 2019.

[25] H. Sun, X. Zheng, X. Lu, and S. Wu, "Spectral–spatial attention network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 5, pp. 3232–3245, May 2020.

[26] C. Shi, J. Sun, T. Wang, and L. Wang, "Hyperspectral image classification based on a 3D octave convolution and 3D multiscale spatial attention network," *Remote Sens.*, vol. 15, no. 1, p. 257, Jan. 2023.

[27] R. Lei, C. Zhang, S. Du, C. Wang, X. Zhang, H. Zheng, J. Huang, and M. Yu, "A non-local capsule neural network for hyperspectral remote sensing image classification," *Remote Sens. Lett.*, vol. 12, no. 1, pp. 77–86, 2021.

ZHANG XIAOXIA received the Ph.D. degree from the College of Geophysics, Chengdu University of Technology. She is currently a Lecturer with the School of Software Engineering, Chengdu University of Information Technology, Chengdu, China. Her research interests include scene classification, high-resolution image processing, and computer vision.

ZHANG XIA received the M.S. and Ph.D. degrees from Chongqing University, Chongqing, China. She is currently an Associate Professor with Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences. Her research interests include remote sensing image processing and spatial analysis algorithms.