Center Attention Network for Hyperspectral Image Classification

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

***Abstract*—Classification is one of the most important research topics in hyperspectral image (HSI) analyses and applications. Although convolutional neural networks (CNNs) have been widely introduced into the study of HSI classification with appreciable performance, the misclassification problem of the pixels on the boundary of adjacent land covers is still significant due to the interfering neighboring pixels whose categories are different from the target pixel. To address this challenge, in this article, we propose a center attention network for HSI classification. The proposed method simultaneously captures spectral-spatial features of the target pixel and its neighboring pixels for classification. Specifically, the method adopts a center attention module (CAM) that pays more attention to the features which are more correlated with the target pixel, that is, the central pixel of the sample, and then sums up the weighted features to generate more relevant and discriminative features. In this way, our method has a high potential for improving the performance of HSI classification. In addition, the CAM greatly reduces the number of parameters in the network via weighted sum of the spectral-spatial features, thus improving the computing efficiency while still maintaining classification accuracy. We evaluate the proposed method on three public datasets, and the experimental results demonstrate the superiority of our method on accuracy and efficiency compared with several state-of-the-art methods.**

***Index Terms*—Attention mechanism, convolutional neural network, deep learning, hyperspectral image classification, spectral-spatial feature extraction.**

I. Introduction

HYPERSPECTRAL image (HSI) classification is an active research topic in remote sensing and earth observation fields due to its important role in land-use and land-cover applications [1]–[3]. HSI is a 3D cube with 1D spectral information and 2D spatial information. While the spatial information reflects the location and structure of objects, the abundant spectral information can be used to distinguish different materials, which is beneficial for analyzing and detecting the earth's surface. Therefore, HSI has gained wide application in miscellaneous domains, such as land scene classification [4], environment monitoring [5],[6], precision agriculture [7], and mineral exploration [8]. Since each HSI pixel can be regarded as a high-dimensional vector, HSI classification, as a significant direction of HSI study, aims to assign each pixel with a proper land-cover class label [9]. However, the high dimensionality of HSI and the large quantity of data compose great challenges for traditional methods to achieve ideal classification results.

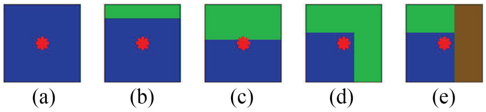

Fig. 1. Different cases of the relationship between the target pixel and its neighboring pixels in subcube samples of hyperspectral images. "*" (red asterisk) denotes the target pixel, and different colors represent different classes of neighborhood pixels.

Recently, deep learning has been recognized as a powerful feature-extraction tool and has shown great advantages in HSI classification [10]–[12]. In terms of whether spatial information is used, deep learning methods for HSI classification fall into spectral-based classification methods and spectral-spatial-based classification methods. The spectral-based methods [13], [14] treat hyperspectral data as a collection of spectral signatures and only use the spectral information when classifying HSIs. As a result, the spatial information of HSI data is ignored so that it is difficult to attain a breakthrough in classification performance. In contrast, the spectral-spatial-based methods [15]–[17] comprehensively integrate the spectral information and spatial information of HSI data. 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 addition, Zheng *et al.* [18] proposed a fast patch-free learning framework which took the whole image as global spatial information. By simultaneously utilizing both the spatial information and spectral information of the subcube samples, the distinguishability of the features is significantly enhanced, thus improving the performance of classification.

Generally, the class labels of all pixels or most of them in a subcube sample are the same, as shown in Fig. 1(a) and (b). However, when the target pixel is located on the boundary of adjacent land covers of different classes, many of its neighboring pixels may actually have different labels, as shown in Fig. 1(c)–(e). In these cases, the classifier may give the target

Manuscript received December 14, 2020; revised January 24, 2021; accepted March 9, 2021. Date of publication March 12, 2021; date of current version April 2, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 41672323. (*Corresponding authors: 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 BRIC, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599 USA (e-mail: hd@bnu.edu.cn).

Hao Wang and Xianchuan Yu are with the 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/JSTARS.2021.3065706

pixel a label to which most pixels in the neighborhood belong rather than its real label, leading to classification mistakes, especially when the neighborhood is large. Furthermore, spectral-spatial features extracted from a subcube sample for HSI classification contain many redundant features. Not all these features have a positive effect for HSI classification, but some of them may heavily interfere with the classification performance. It is difficult to distinguish these unfavorable features because the weights of all the features are equal. These problems bring great challenges to the classification algorithms and affect the further improvement of their performance.

To address these challenges, in this article, we propose a novel method, the center attention network (CAN), for HSI classification. First, the proposed method employs 3D CNN to extract basic spectral-spatial features of the sample. Then, the method adopts a center attention module (CAM) that pays more attention to the features which are more correlated with the target pixel, i.e., the central pixel of the sample, and assigns them different weights according to their correlation levels. The CAM sums up these weighted features to generate new spectral-spatial features and meanwhile reduces the number of the features. Finally, the sample is classified by the classification module with new spectral-spatial features.

To sum up, the major contributions of this article are listed as follows. We propose a novel end-to-end method for HSI classification and first present the CAM. Specially, the CAM focuses on the features that are more correlated with the target pixel and generates more relevant and discriminative spectral-spatial features. Furthermore, the proposed method considerably reduces the number of parameters in the network by reducing the number of spectral-spatial features, thus improving the computing efficiency while still maintaining classification accuracy. Finally, experiments on three public datasets show the superiority of our method on accuracy compared with several state-of-the-art methods.

The remainder of this article is organized as follows. First, Section II reviews the works related to HSI classification. Then, the proposed method and its rationale are detailed in Section III. Next, Section IV illustrates a series of experiments and results. Finally, conclusions are drawn in Section V.

II. Related Work

CNNs have been widely applied in HSI classification, as attention mechanisms have become increasingly active in this field for effective feature selection. In this section, CNNs and basic attention mechanism related to HSI classification are reviewed.

A. CNNs for HSI Classification

In recent years, CNNs have been successfully applied in the field of image processing, such as image classification [19], image recognition [20], and image inpainting [21]. CNNs are usually a multilayer network structure. When CNNs are used for classification, they mainly include two parts: a feature-extraction (FE) network and a classification network. The FE network aims to learn high-level representations of the inputs, and the classification network performs the final classification task to assign each input sample with a certain class [3]. There are two main types of approaches for CNNs applied in hyperspectral data classification: spectral analysis and spectral-spatial analysis. The methods based on spectral signatures regard the original spectral vectors or a reasonable number of spectral channels as the input data for HSI classification. In [13], [22], 1D CNNs were employed to capture deep spectral features of pixels for HSI classification. Charmisha *et al.* [23] proposed a vectorized CNN to perform dimension reduction. Zhan *et al.* [14] used 1D generative adversarial network to learn the spectral features. 1D CNN only uses the spectral information, but the spatial information is ignored. Actually, spatial information has been reported to be very useful in improving the representation of hyperspectral data and increasing the classification accuracies [2], [24]. Some works have explored 2D CNN for extracting spatial features of pixels. In 2D CNN framework, the spectral features are usually processed by dimension reduction methods. In [25], [26], the authors extracted the first principal component as spectral features and then employed the 2D CNN to extract the spatial features for HSI classification. Song *et al.* [27] adopted residual learning to extract deep features and fused the features of hierarchical layers to improve the classification accuracy. Zhu *et al.* [28] proposed a deformable CNN-based method, in which the authors compressed adjacent similar structural information into fixed grids to extract features. In [29], the authors decoupled the feature maps of input patches into multiple response maps and adaptively selected the meaningful maps for classification.

However, the existing spectral-based methods and some 2D spectral-spatial-based methods only use spectral features or capture local spatial features of the pixels. The performance of these methods is restricted as a result of not exploring both spectral and spatial features simultaneously. Recently, 3D CNN can extract the spectral-spatial features of HSI concurrently, which has attracted the interest of many researchers. Ying *et al.* [15] directly employed 3D CNN to extract deep spectral-spatial features for HSI classification. Chen *et al.* [26] used 3D CNN with regularization to obtain spectral-spatial features for HSI classification. Zhong *et al.* [30] designed a 3D spectral and spatial residual block which can consecutively learn the deep spectral-spatial features. Mei *et al.* [31] used a 3D convolutional autoencoder to learn spectral-spatial features without supervision. HSIs are data cubes in which spectral and spatial information coexist, and 3D CNN filters are a natural method for discovering the spectral-spatial features within such images. To explore the spectral-spatial features as a whole, our method employs 3D CNN for extracting basic spectral-spatial features. The existing methods directly use the basic features or select key features from them for HSI classification. Different from them, our method focuses on the features that are more relevant to the target pixel and assign them different weights according to their correlation levels through CAM, and then sums them up to generate new spectral-spatial features with more discriminative characteristics.

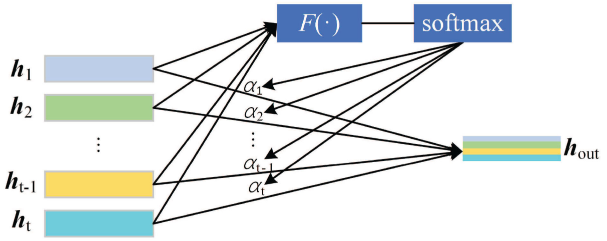

Fig. 2. Process of the basic attention mechanism. $\boldsymbol{h}_1, \boldsymbol{h}_2 \ldots, \boldsymbol{h}_{t-1}$, and $\boldsymbol{h}_t$ are input features. $\alpha_1, \alpha_2 \ldots, \alpha_{t-1}$, and α_t are their weights. $\boldsymbol{h}_{\text{out}}$ is the output feature.

B. Attention Mechanism

As a research hotspot in computer vision, attention mechanism has been widely used in various fields of deep learning, such as machine translation [32], object recognition [33], pose estimation [34], saliency detection [35], and scene segmentation [36]. Fu *et al.* [36] adopted the position attention module and channel attention module to learn the spatial and channel information separately. Hu *et al.* [37] employed squeeze and excitation operations to assign different weights to different channels for selecting important feature maps.

Attention mechanism is a method that simulates human visual perception. When a person observes an object, the vision quickly scans the global image, focuses on the key area, and suppresses other useless information and background information. Attention mechanism in computer vision is similar to that in human vision. Its purpose is to focus on some important features of the target and select more critical features from a large number of features. Fig. 2 shows the process of the basic attention mechanism.

As shown in Fig. 2, under the basic attention mechanism, the output feature is the weighted sum of each input feature according to its importance. The formula is as follows:

$$\boldsymbol{h}_{\text{out}} = \sum_{i=1}^{t} \alpha_i \boldsymbol{h}_i \tag{1}$$

where $\boldsymbol{h}_{\text{out}}$ is the output feature, $\boldsymbol{h}_1 \ldots, \boldsymbol{h}_t$ are the input features, $\alpha_1 \ldots, \alpha_t$ are the corresponding weights, and t is the number of input features. α_i is obtained by a softmax function. It is defined as

$$\alpha_i = \frac{\exp(F(\boldsymbol{h}_i))}{\sum_{i=1}^{t} \exp(F(\boldsymbol{h}_i))} \tag{2}$$

where $F(\cdot)$ denotes the scoring function and $\exp(\cdot)$ denotes the exponential function.

Recently, the attention mechanism has shown great potential in the field of remote sensing. Some researchers have introduced it into HSI classification. Fang *et al.* [38] exploited 3D dilated convolutions to capture the spectral-spatial features, and then adopted spectralwise attention to enhance the distinguishability of spectral features. Ma *et al.* [39] applied two types of attention mechanism in two branches to extract spectral and spatial features and then concatenated them for classification. The work in [16] applied the spectral attention Bi-RNN branch for spectral features and applied the spatial attention CNN branch for spatial features. Sun *et al.* [9] embedded the attention module after both the spectral module and spatial module to suppress the impact of interfering pixels. In early works, the attention mechanism was independently applied to spectral and spatial features and then merged the outputs, or it was sequentially used after spectral modules and spatial modules to select key features. In our method, the CAM is exploited to seek the desired spectral-spatial features that are more correlated with the target pixel and assign them different weights. Then, the CAM sums of the weighted features to get more discriminative features, which not only introduces a target focused strategy but also reduces the number of parameters.

III. Proposed Method

In this section, we describe the proposed CAN in detail. CAN contains three parts: a 3D CNN module, a CAM, and a classification module. The 3D CNN module is used to capture the basic spectral-spatial features of the target pixel and its adjacent pixels; the CAM aims to fuse these features and generate more discriminative features; the target pixel is classified by the classification module with a softmax function. Fig. 3 illustrates the architecture of our proposed CAN.

A. 3D CNN Module for Spectral-Spatial Features

An HSI is represented in a 3D cube. In the proposed method, to explore both spectral and spatial information simultaneously, the 3D CNN module is employed as a feature extractor, consisting of convolution layers, batch normalization layers, nonlinearity layers, and pooling layers.

1) 3D Convolution Layer: The convolution layer is a layer where each neuron computes the dot product between its weights and a small region of the input volume matched to it. The layer's goal is to identify certain features from the previous layer and transform them to feature maps. It is formulated as follows [17]:

$$\boldsymbol{O}^{d' \times r' \times l'} = \boldsymbol{I}^{d \times r \times l} \otimes \boldsymbol{W}^{k_1 \times k_2 \times k_3} + \boldsymbol{b} \tag{3}$$

where $\boldsymbol{I}$ is the input volume, $\boldsymbol{O}$ is the output volume, $\boldsymbol{W}$ is the filter (neuron or kernel) with the size $k_1 \times k_2 \times k_3$, $\boldsymbol{b}$ is the bias, $d \times r \times l$ represents the size of the input volume, and $d' \times r' \times l'$ represents the size of the output volume. $\otimes$ denotes the convolution operation. Fig. 4 shows the process of 3D convolution.

Generally, multiple 3D convolution filters are stacked in one layer to explore different kinds of spectral-spatial features. The 3D convolutional layer can produce many spectral-spatial feature maps. When 3D convolutional layers are connected sequentially, more abstract spectral-spatial features are extracted.

2) Batch Normalization Layer: This layer is often used to improve the numerical stability. The batch normalization [40] is represented as

$$\text{BN}(x) = \frac{x - \text{mean}[\hat{\boldsymbol{x}}]}{\sqrt{\text{var}[\hat{\boldsymbol{x}}] + \epsilon}} * \gamma + \beta \tag{4}$$

where $\hat{\boldsymbol{x}}$ is a minibatch of inputs, $\text{mean}[\hat{\boldsymbol{x}}]$ and $\text{var}[\hat{\boldsymbol{x}}]$ represent the mean and standard deviation of $\hat{\boldsymbol{x}}$ which are calculated over

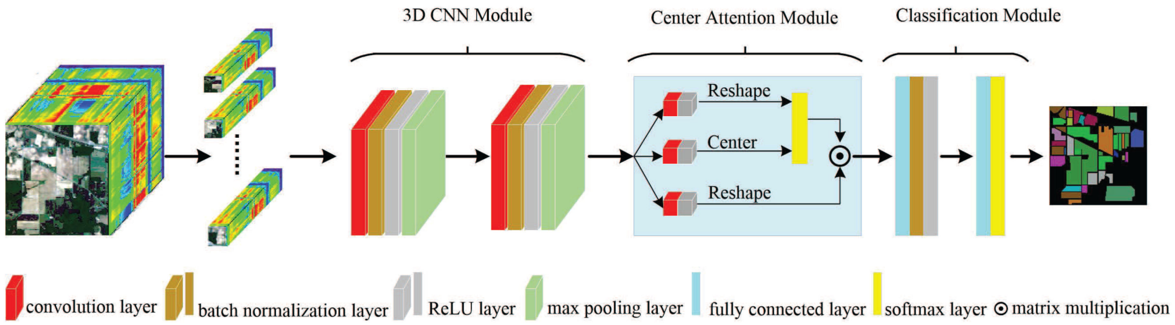

Fig. 3. Architecture of the proposed CAN method. It is an end-to-end method. First, subcube samples are cropped from the dataset. Next, basic spectral-spatial features are extracted from the 3D CNN module. Then, the center attention module (CAM) processes these features and weights them together. Finally, the classification module assigns labels to samples.

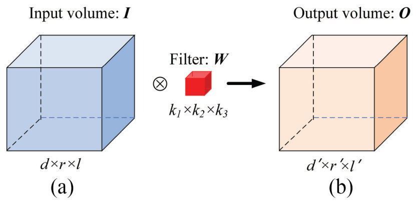

Fig. 4. Process of 3D convolution. $\boldsymbol{I}$ is the input volume, $\boldsymbol{O}$ is the output volume, $\boldsymbol{W}$ is the filter, and $\otimes$ denotes the convolution operation.

a minibatch, γ and β are the learnable parameters, and ϵ is a very small constant value.

3) Nonlinearity Layer: This layer is applied to learn the nonlinear relationship contained in the previous volume by leveraging a nonlinear function. In this article, we adopt the rectified linear unit (ReLU) [41] as the nonlinear function. It is defined as

$$\sigma(x) = \max(0, x). \tag{5}$$

4) Pooling Layer: This layer is often used to summarize the features and reduce the feature dimensions through a pooling function. In our proposed method, 3D max pooling is applied to extract spectral-spatial features after the nonlinear layer. The 3D max pooling operation takes the maximum value within a small spatial region of the input volumes, and it is defined as

$$O_{p,q,z} = \max(I_{p+\delta_p, q+\delta_q, z+\delta_z}) \tag{6}$$

where $I_{p+\delta_p, q+\delta_q, z+\delta_z}$ represents the input values at position (p, q, z) with a region of size $(\delta_p, \delta_q, \delta_z)$ and $O_{p,q,z}$ represents the output value at position (p, q, z) after 3D max pooling.

B. Center Attention Module

The basic attention mechanism automatically selects the key features and ignores trivial features. The key features selected usually represent the major information of the samples. However, when the samples contain considerable disturbing information, the key features selected may not correctly represent the salient information of the target, thus leading to classification mistakes. Taking HSI classification as an example, when the target pixel is on the edge of two or more classes of land covers, as shown in Fig. 1(c)–(e), there will be many interfering pixels around it. The interfering pixels often have different labels from the target pixel. Sometimes, they occupy the majority in the neighborhood of the target pixel. Under these cases, the key features selected by the basic attention may not correctly represent the target pixel, leading to classification mistakes. How to accurately extract and choose better features representing the target pixel becomes the core of improving classification accuracy.

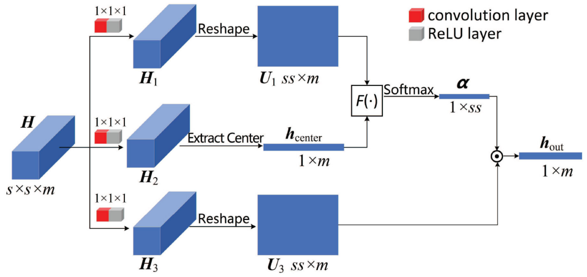

Fig. 5. Details of the CAM. The input feature is a cube of size $s \times s \times m$. The output feature is a vector of size $1 \times m$. $\odot$ denotes matrix multiplication.

To address this problem, we propose a novel CAM to seek the desired spectral-spatial features which are more discriminative for the classification. The CAM focuses more on the features that are highly correlated with the central pixel (i.e., the target pixel) in the subcube sample. Since the convolution filters scan the samples sequentially, the central filtered features can often better represent the central (target) pixel. So we calculate the correlation scores between all the features and the central features to evaluate the contribution of different features for the classification of the target pixel. Then, the CAM exerts unequal weights on these features according to their correlation scores. The stronger the correlation, the greater the weight, and vice versa. Finally, it sums up these features of different weights to reduce the number of features and generate new spectral-spatial features. These new features are more discriminative for classification in that they are more relevant to the target pixel. The details of the CAM are shown in Fig. 5.

As illustrated in Fig. 5, first, we take an output of the prior convolution block as the input feature map $\boldsymbol{H} \in \mathbb{R}^{s \times s \times m}$ for CAM. Then, we perform three convolution layers with the kernel size of $(1 \times 1 \times 1)$ [42], [43] and three ReLU layers on $\boldsymbol{H}$ separately, and generate three different new feature maps $\boldsymbol{H}_1$, $\boldsymbol{H}_2$, and $\boldsymbol{H}_3$. The $(1 \times 1 \times 1)$ convolution layers and ReLU layers are used to enhance the nonlinear representation of the features. In fact, $\boldsymbol{H}_1$ and $\boldsymbol{H}_2$ are to calculate the attention vector that softly weights the importance of different features, and the goal of $\boldsymbol{H}_3$ is a simple nonlinear transformation of the input feature map but with suitable dimensions. In order to do matrix operations, $\boldsymbol{H}_1$ and $\boldsymbol{H}_3$ are reshaped into $\boldsymbol{U}_1, \boldsymbol{U}_3$, where $\boldsymbol{U}_1, \boldsymbol{U}_3 \in \mathbb{R}^{ss \times m}$ and $ss = s \times s$. The center feature vector $\boldsymbol{h}_{\text{center}}$ is extracted from the center of $\boldsymbol{H}_2$, where $\boldsymbol{h}_{\text{center}} \in \mathbb{R}^{1 \times m}$. Both $\boldsymbol{U}_1$ and $\boldsymbol{h}_{\text{center}}$ are fed into the scoring function $F(\cdot)$ to calculate the correlation scores between them, and then a softmax function are applied on correlation scores to calculate the attention vector $\boldsymbol{\alpha}$. Finally, the output feature vector $\boldsymbol{h}_{\text{out}}$ of the CAM is obtained by multiplying matrices $\boldsymbol{\alpha}$ and $\boldsymbol{U}_3$, where $\boldsymbol{h}_{\text{out}} \in \mathbb{R}^{1 \times m}$. It is formulated as

$$\boldsymbol{h}_{\text{out}} = \boldsymbol{\alpha} \boldsymbol{U}_3 \tag{7}$$

$$\alpha_i = \frac{\exp(F(g_i))}{\sum_{i=1}^{ss} \exp(F(g_i))} \tag{8}$$

where $\boldsymbol{\alpha} \equiv [\alpha_1, \ldots, \alpha_{ss}]$, $\boldsymbol{U}_1 \equiv [\boldsymbol{h}_1, \ldots, \boldsymbol{h}_{ss}]^{\mathrm{T}}$, α_i is obtained by the softmax function, $\boldsymbol{h}_{\text{center}}$ is the center feature vector, and $\exp(\cdot)$ denotes the exponential function. $F(\cdot)$ denotes the scoring function, which is implemented by a full connection layer, parameterized by a weight matrix, $\boldsymbol{W} \in \mathbb{R}^{ss \times ss}$. g_i are used to calculate the correlation between $\boldsymbol{h}_i$ and $\boldsymbol{h}_{\text{center}}$, and m is the length of $\boldsymbol{h}_{\text{center}}$. The correlation scores are obtained by multiplying all the g_i with $\boldsymbol{W}$ and activating the results by a nonlinear function $\delta(\cdot)$, i.e., ReLU. They are formulated as

$$F(g_i) = \delta \left(\sum_{i=1}^{ss} g_i \boldsymbol{W} \right) \tag{9}$$

$$g_i = \frac{1}{m} \left\| \boldsymbol{h}_i - \boldsymbol{h}_{\text{center}} \right\|_2^2 . \tag{10}$$

C. Center Attention Network

CAN is an end-to-end method based on patch for HSI classification. It takes the target pixel and its neighbor pixels together as a subcube sample (i.e., patch) whose class label is that of its central pixel. The method mainly contains three parts: the 3D CNN module, the CAM, and the classification module. Fig. 3 portrays the architecture of the proposed method.

First, many subcube samples are cropped from the dataset. Next, basic spectral-spatial features are extracted from the 3D CNN module built with two sequential 3D convolution blocks. Each block consists of a convolutional layer, a batch normalization layer, a ReLU layer, and a max pooling layer. Then, the CAM assigns different weights to these spectral-spatial features according to their relevance to the target pixel and then sums up these weighted features to generate more discriminative features. Its detailed process is shown in Fig. 5. Finally, these new spectral-spatial features are fed into the classification module. The label values are determined by a classifier with a softmax function. The classifier is composed of fully connected layers, a batch normalization layer, a ReLU layer, and a softmax layer. The categorical cross entropy is employed as the loss function, defined as

TABLE I
DETAILED PARAMETERS OF EACH LAYER IN THE PROPOSED METHOD

Layer#	Layer Name	Kernel Number	Kernel Size
1	3D convolution	32	$3 \times 3 \times 7$
2	batch normalization	-	-
3	ReLU	-	-
4	3D maxpooling	-	$1 \times 1 \times 3$
5	3D convolution	64	$3 \times 3 \times 7$
6	batch normalization	-	-
7	ReLU	-	-
8	3D maxpooling	-	$1 \times 1 \times 3$
9	CAM	-	-
10	dense	300	-
11	batch normalization	-	-
12	ReLU	-	-
13	dense	c	-
14	softmax	-	-

$$\text{Loss} = -\sum_{i=1}^{c} y_i \log(p_i) \tag{11}$$

where c is the number of land-cover classes, p_i is the output of the CAN, y_i is the label value, and $y_i \in \{0, 1\}$ (if y_i is the ith class $y_i = 1$, else $y_i = 0$). For its robustness in learning, the Adam [44] optimizer is adopted. Table I specifies the detailed parameters of each layer in the proposed method. In Table I, c is the number of land-cover classes.

IV. Experiments

To evaluate the effectiveness of our proposed method for HSI classification, we conducted a series of experiments on three public datasets. Experimental results demonstrate that the proposed method achieved better results compared with several state-of-the-art methods.

A. Datasets

The datasets used in the experiments were Indian Pines (IP), University of Pavia (UP), and Salinas Valley (SV), which are widely used in the validation of HSI classification methods. Next, we introduce these datasets in detail.

1) Indian Pines: The IP dataset was collected in northwest Indiana by the AVIRIS sensor in 1992. It includes 220 spectral bands from wavelengths of 400–2500 nm with an interval of 10 nm. There are 200 usable bands left after the removal of the water absorption and null bands. The size of the image is 145 × 145, and the spatial resolution is 20 m. In this dataset, 16 different land-cover categories are included, with a total of 10 249 labeled pixels. Fig. 6 shows the pseudocolor image and the ground-truth map of the IP dataset.

2) University of Pavia: The UP dataset was acquired through the ROSIS sensor in 2003. It includes 115 bands. After the noise

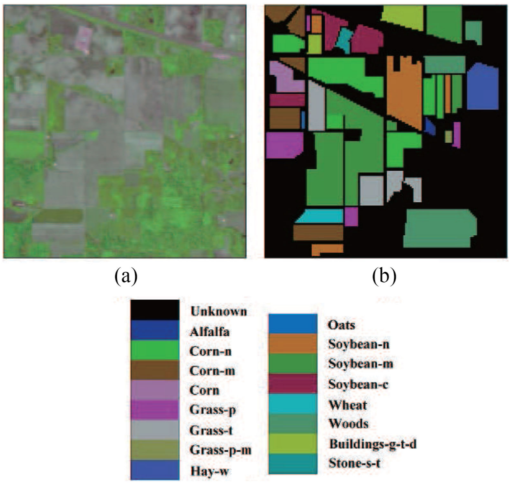

Fig. 6. (a) Pseudocolor image of the Indian Pines dataset. (b) Ground-truth map of the Indian Pines dataset.

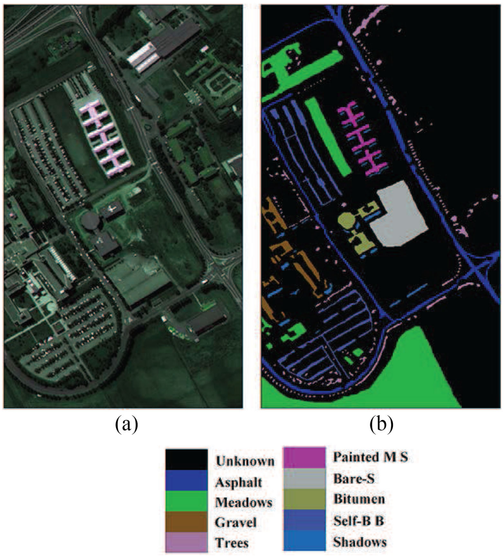

Fig. 7. (a) Pseudocolor image of the University of Pavia dataset. (b) Ground-truth map of the University of Pavia dataset.

bands were removed, 103 available bands remained. The wavelength range is from 380 to 860 nm, and the spatial resolution is 1.3 m. The size of the image is 610 × 340. There are nine types of land cover and a total of 42 776 labeled samples in the UP dataset. Fig. 7 shows the pseudocolor map and the ground-truth map of the UP dataset.

3) Salinas Valley: The Salinas Valley dataset was acquired by the AVIRIS sensor in 1998. It includes 224 bands, with a wavelength range from 400 to 2500 nm. After removing the water absorption and noise bands, 204 bands remained. The size of the image is 512 × 217, and its ground resolution is 3.7 m. In this dataset, there are 16 types of land cover and a total of 54 129 labeled samples. Fig. 8 shows the pseudocolor map and the ground-truth map of the SV dataset.

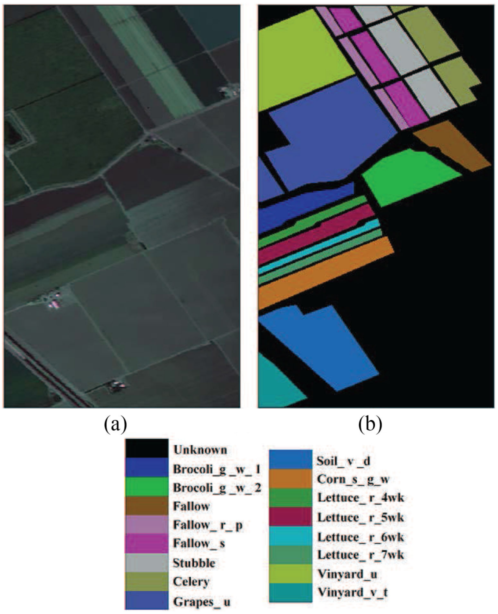

Fig. 8. (a) The pseudocolor image of the Salinas Valley dataset. (b) The ground-truth map of the Salinas Valley dataset.

B. Experimental Settings and Measures

In this section, data preprocessing and data augmentation methods are introduced. In data preprocessing, we normalize the data with maximum and minimum values. Then, the normalized data are subtracted from the average value of the corresponding band.

In data augmentation, we reverse and rotate the subcubes to alleviate the overfitting problem due to insufficient labeled samples. First, a subcube sample is flipped horizontally and vertically; second, the subcube sample is rotated 90, 180, and 270 degrees around the central pixel. After these operations, each subcube sample generates five additional samples. In addition, batch normalization adopted in the proposed method can also relieve the overfitting problem.

To quantitatively analyze the performance of the algorithm, we use the overall accuracy (OA), average accuracy (AA), and kappa as evaluation measures. OA refers to the proportion of all correctly classified samples in the test samples; AA refers to the average classification accuracy of different categories; and kappa measures the consistency between classification results and ground truth. The larger the value of OA, AA, and kappa, the better the results.

C. Comparing With Other Methods

To verify the performance of the proposed CAN method, we perform a comparison between the proposed method and several state-of-the-art methods, including 1D CNN, 2D CNN [26], SMBN (squeeze multibias network) [29], DFFN (deep feature fusion network) [27], DHCNet (deformable HSI classification networks) [28], SSRN (spectral-spatial residual network) [30], and SSAN (spectral-spatial attention networks) [9], and they are all based on deep learning with CNN modules. To make a fair

TABLE II
NUMBER OF TRAINING SAMPLES, TESTING SAMPLES, AND TOTAL SAMPLES ON THE INDIAN PINES DATASET

Class#	Name	Training	Testing	Total
1	Alfalfa	5	41	46
2	Corn-n	143	1285	1428
3	Corn-m	83	747	830
4	Corn	24	213	237
5	Grass-p	48	435	483
6	Grass-t	73	657	730
7	Grass-p-m	3	25	28
8	Hay-w	48	430	478
9	Oats	2	18	20
10	Soybean-n	97	875	972
11	Soybean-m	246	2209	2455
12	Soybean-c	59	534	593
13	Wheat	21	184	205
14	Woods	127	1138	1265
15	Buildings-g-t-d	39	347	386
16	Stone-s-t	9	84	93
	Sum	1027	9222	10249

TABLE III
NUMBER OF TRAINING SAMPLES, TESTING SAMPLES, AND TOTAL SAMPLES ON THE UNIVERSITY OF PAVIA DATASET

Layer#	Name	Training	Testing	Total
1	Asphalt	133	6498	6631
2	Meadows	373	18276	18649
3	Gravel	42	2057	2099
4	Trees	61	3003	3064
5	Painted-M-S	27	1318	1345
6	Bare-S	101	4928	5029
7	Bitumen	27	1303	1330
8	Self-B-B	74	3608	3682
9	Shadows	19	928	947
	Sum	857	41919	42776

TABLE IV
NUMBER OF TRAINING SAMPLES, TESTING SAMPLES, AND TOTAL SAMPLES ON THE SALINAS VALLEY DATASET

Class#	Name	Training	Testing	Total
1	Brocoli_g_w_1	40	1969	2009
2	Brocoli_g_w_2	75	3651	3726
3	Fallow	40	1936	1976
4	Fallow_r_p	28	1366	1394
5	Fallow_s	54	2624	2678
6	Stubble	79	3880	3959
7	Celery	72	3507	3579
8	Grapes_u	225	11046	11271
9	Soil_v_d	124	6079	6203
10	Corn_s_g_w	66	3212	3278
11	Lettuce_r_4wk	21	1047	1068
12	Lettuce_r_5wk	39	1888	1927
13	Lettuce_r_6wk	18	898	916
14	Lettuce_r_7wk	21	1049	1070
15	Vinyard_u	145	7123	7268
16	Vinyard_v_t	36	1771	1807
	Sum	1083	53046	54129

comparison, our method and comparison methods proposed in this article use the same experimental settings, including data preprocessing and data augmentation. The detailed parameters are set as follows. The spatial size of the HSI subcube of all methods is set to 7 × 7. The number of training epochs is set to 200, but 1000 for 1D CNN because it is trained without data augmentation. The number of batch sizes is 100. The weight parameters of each method are optimized by Adam [44]. The learning rates of the competitive methods are the same as those of the original paper. The learning rate of the proposed method is 0.001.

These experiments are conducted on the IP, UP, and SV datasets. On the IP dataset, we randomly select 10% of the labeled samples in each land-cover category as training samples, and the rest are test samples. On the UP and SV datasets, 2% of the labeled samples are randomly selected as training samples, and the rest of the labeled samples are test samples. The number of training and test samples belonging to different categories on the IP, UP, and SV datasets are shown in Tables II–IV. Table V shows the classification results of different methods on the IP dataset, Table VI on the UP dataset, and Table VII on the SV dataset [16]. We highlight the best results in italic.

From the results in Tables V–VII, it is obvious that the methods based on spectral-spatial features show superior performance over the method based on only spectral features (1D CNN). This demonstrates that the spatial information is helpful for improving classification performance.

We also find that the SSAN method and the proposed method outperform other methods based on spectral-spatial features because the two methods manage to select the required features and suppress unwanted ones. This indicates that some features extracted by CNN are redundant, and many of them are useless or even counterproductive. This also affirms that the CAM in our method is necessary for selecting and summing up these basic spectral-spatial features.

From these classification results, we further discover that the classification accuracies vary greatly among different categories because the number of samples belonging to different classes is unequal, resulting in an imbalance among the training samples, especially on the IP dataset. The category with the fewest samples is "Oats," which has only 2 samples, but the "Soybean-m" category has 246 samples. This imbalance between the number of training samples poses a major challenge to classification methods. In terms of AA, the proposed method is better than comparative methods when the dataset is unbalanced.

D. Impact of Spatial Size

The spatial size of the subcube has an important impact on the classification results [3]. In this section, we conduct several experiments on the IP, UP, and SV datasets to explore the impact of size on the classification results. The ratios of labeled samples on the IP, UP, and SV datasets are 10%, 2%, and 2%, respectively. The spatial sizes of the subcube are set to 5 × 5, 7 × 7, 9 × 9, 11 × 11, and 13 × 13. The number of training epochs is 100, and the number of batch sizes is 100. All the other parameters retain the settings of the previous experiments. Fig. 9 shows the OAs of the proposed method on the three datasets with different spatial sizes.

From Fig. 9, we find that the classification performance gradually improves as the spatial size expands. The reason is that a larger sample may contain more spatial information. However, the effect of the increased spatial size on the classification performance is different on the IP, UP, and SV datasets. When the size

TABLE V
Classification Results (%) on the Indian Pines Dataset Under Different Methods

Class	1D CNN	2D CNN	SMBN	DFFN	DHCNet	SSRN	SSAN	Proposed Method
1	14.63	78.05	82.93	**100**	82.93	82.93	80.49	87.80
2	74.47	79.77	82.80	91.44	91.44	92.37	90.82	**98.05**
3	69.75	83.94	92.77	93.44	88.76	97.86	93.84	**97.99**
4	55.87	90.14	84.51	94.37	83.10	62.44	89.20	**94.37**
5	87.59	**99.54**	97.93	97.24	96.31	97.47	99.08	98.39
6	97.41	98.63	99.70	99.24	99.24	**99.85**	99.24	99.70
7	56.00	76.00	96.00	68.00	84.00	60.00	96.00	**100**
8	96.98	99.77	96.51	99.77	97.21	**100**	98.14	**100**
9	11.11	**100**	**100**	88.89	88.89	44.44	**100**	77.78
10	76.91	89.24	89.36	93.82	92.86	91.30	94.62	**98.17**
11	80.67	95.74	97.69	95.88	96.15	97.69	98.10	**98.33**
12	61.42	76.55	85.74	84.80	86.30	91.37	94.56	**97.94**
13	98.37	**100**	94.57	97.83	95.65	**100**	**100**	**100**
14	95.69	**99.21**	97.19	98.42	97.28	98.59	98.42	98.77
15	56.77	85.30	82.42	**96.54**	88.47	82.71	82.71	92.51
16	84.52	**100**	95.18	98.80	92.77	**100**	91.57	98.81
OA	80.00	91.31	92.74	95.01	93.68	94.77	95.49	**98.10**
AA	69.89	90.74	92.21	93.65	91.32	87.44	94.17	**96.16**
kappa	77.14	90.06	91.70	94.31	92.78	94.03	94.85	**97.84**

TABLE VI
Classification Results (%) on the University of Pavia Dataset Under Different Methods

Class	1D CNN	2D CNN	SMBN	DFFN	DHCNet	SSRN	SSAN	Proposed Method
1	90.10	93.63	94.51	94.24	96.03	97.78	98.68	**99.54**
2	96.57	98.20	99.56	99.10	98.68	98.59	99.44	**99.78**
3	70.15	92.32	78.80	86.49	83.33	**95.77**	86.00	93.15
4	89.24	**99.00**	95.74	97.57	98.83	95.07	98.33	98.63
5	99.39	99.62	94.08	98.33	97.95	**100**	99.92	**100**
6	84.48	99.39	96.96	96.63	96.88	98.94	99.11	**99.78**
7	73.91	94.93	86.88	87.87	86.80	93.32	96.55	**98.08**
8	80.04	83.76	83.12	92.82	93.04	90.96	94.07	**96.06**
9	99.57	99.25	99.14	98.92	**100**	99.68	**100**	99.89
OA	90.36	96.12	95.19	96.41	96.47	97.36	98.02	**98.97**
AA	87.05	95.57	92.09	94.66	94.62	96.68	96.90	**98.32**
kappa	87.15	94.89	93.61	95.24	95.32	96.51	97.37	**98.64**

TABLE VII
Classification Results (%) on the Salinas Valley Dataset Under Different Methods

Class	1D CNN	2D CNN	SMBN	DFFN	DHCNet	SSRN	SSAN	Proposed Method
1	99.03	**100**	81.20	99.49	99.54	**100**	98.78	**100**
2	99.73	98.63	99.73	99.81	99.89	97.89	**99.97**	99.95
3	93.34	94.37	89.72	96.07	97.42	99.48	98.66	**99.64**
4	98.10	98.17	98.83	99.34	98.10	99.63	99.05	**99.85**
5	95.92	77.06	98.02	99.35	97.98	**99.39**	**99.39**	98.36
6	99.61	**100**	99.97	99.90	99.97	**100**	99.97	99.92
7	99.17	99.91	99.37	99.63	99.71	**100**	99.91	99.97
8	95.06	87.93	89.13	91.44	89.37	88.26	92.46	96.36
9	99.46	**100**	99.42	98.93	**100**	**100**	99.95	99.97
10	90.20	95.83	96.67	95.98	96.82	97.70	96.33	**99.28**
11	88.73	99.33	97.99	98.37	99.04	**99.90**	99.43	99.43
12	99.79	98.94	97.03	**100**	99.79	**100**	**100**	**100**
13	97.88	99.89	99.44	98.55	95.88	**100**	**100**	95.77
14	91.90	97.52	98.47	99.52	**100**	99.33	99.81	98.47
15	22.70	91.01	85.73	88.46	85.41	93.30	91.39	**94.33**
16	94.35	89.21	97.91	**99.32**	98.70	99.15	98.19	98.93
OA	86.71	94.08	94.03	95.96	95.21	96.27	96.81	**98.18**
AA	91.56	95.49	95.54	97.76	97.35	98.38	98.33	**98.76**
kappa	85.08	93.41	93.36	95.50	94.66	95.85	96.54	**97.97**

is 13 × 13, the performance on the IP dataset begins to decline slightly. This is because as the size increases, there will be more interfering pixels in subcube samples, which may affect the classification performance of the method. Therefore, a suitable spatial size is very important. In the following experiments, the spatial size is set to 11 × 11.

E. Effectiveness of CAM

The CAM plays an essential role in the proposed method. It effectively fuses the spectral-spatial features, which greatly reduces the number of parameters and improves the training efficiency of the method. Under the same configurations, the

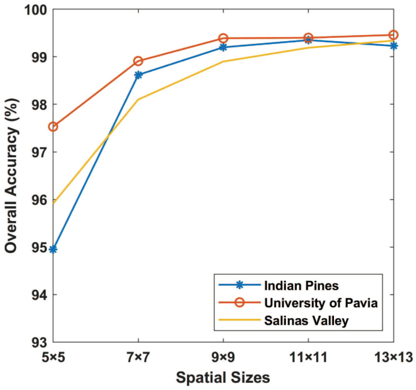

Fig. 9. The OAs (%) of the proposed method on the IP, UP, and SV datasets with different spatial sizes.

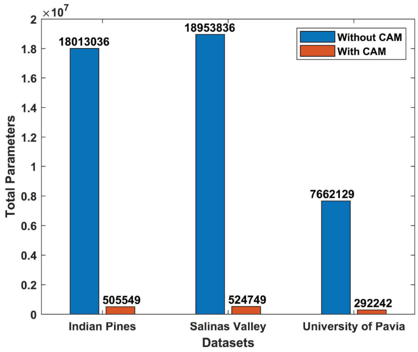

Fig. 10. Number of parameters of our method with and without CAM.

TABLE VIII
TRAINING TIME OF THE METHOD WITH AND WITHOUT THE CAM (S)

	Indian Pines	University Pavia	Salinas Valley
Without CAM	883	401	902
With CAM	768	377	792

number of parameters of our method with and without CAM is shown in Fig. 10, and their training times are shown in Table VIII. The results in Fig. 10 suggest that the number of parameters of the method with CAM is much less than that of the method without CAM. According to Table VIII, the time spent by the method with CAM is less than that of the method without CAM. The reduction in parameter quantity and the improvement in training efficiency benefit from weighted sum of spectral-spatial features by the CAM.

The CAM helps generate more relevant and discriminative spectral-spatial features and improves the classification performance. To verify the effectiveness of the CAM, the experiments are conducted on the IP, UP, and SV datasets with and without CAM for comparison. In these experiments, the spatial size is 11 × 11, the number of training epochs is 100, the batch size is 100, and the optimizer is Adam. Table IX shows the OAs of the results on the IP, UP, and SV datasets at 10%, 2%, and 2% of the labeled samples, respectively. From Table IX, it is confirmed that the CAM significantly improved accuracy.

TABLE IX
OAS (%) OF THE METHOD WITH AND WITHOUT CAM ON THE IP, UP, AND SV DATASETS

	Indian Pines	University Pavia	Salinas Valley
without CAM	97.97	98.65	98.59
with CAM	99.35	99.40	99.19

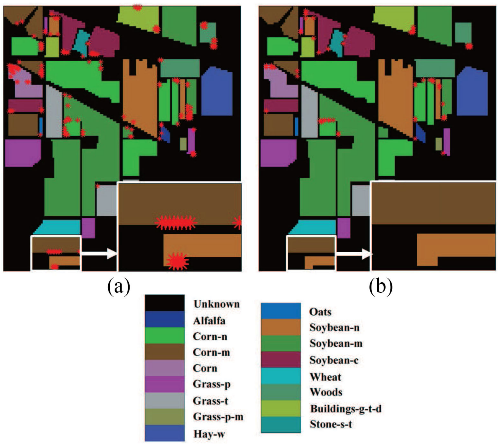

Fig. 11. The visual results of our methods with and without CAM on the IP dataset. (a) The method without CAM. (b) The method with CAM.

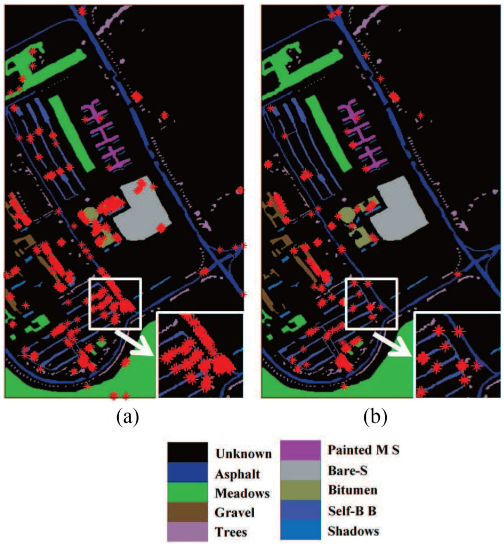

Fig. 12. The visual results of our methods with and without CAM on the UP dataset. (a) The method without CAM. (b) The method with CAM.

To visually display and verify the classification results, Figs. 11–13 portray the classification results of the method with and without CAM on the IP, UP, and SV datasets. In these figures, "*" (red asterisk) represents the misclassified labeled samples, and others are correctly identified labeled samples (the black area is the background pixels).

From Figs. 11 to 13, it can be seen that the CAM is effective and helpful for improving the classification performance, especially at the boundary of different classes. There are many

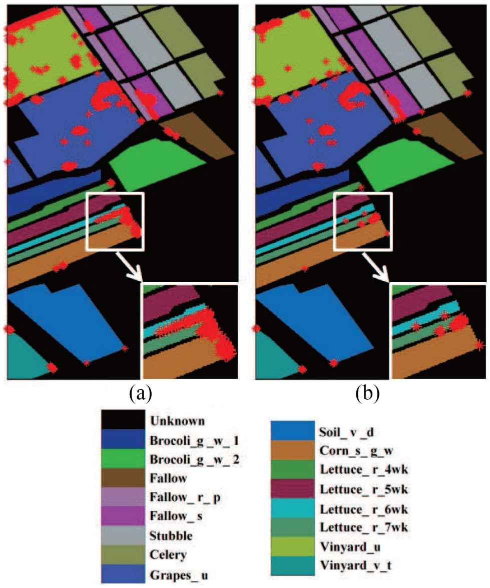

Fig. 13. The visual results of our methods with and without CAM on the SV dataset. (a) The method without CAM. (b) The method with CAM.

TABLE X
OAs (%) of CAN With Different Ratios of Labeled Samples as Training Samples on the IP, UP, and SV Datasets

Datasets	2%	5%	10%	15%	20%
Indian Pines	-	97.62	99.35	99.43	99.62
University of Pavia	99.40	99.81	99.97	99.97	99.97
Salinas Valley	99.19	99.70	99.97	99.97	99.97

misclassified samples in Figs. 11–13(a) that are properly corrected in Figs. 11–13(b), especially on the boundary. The main reasons are as follows: 1) the proposed method finds the internal correlation between the target pixel and its neighboring pixels, and the pixels with higher correlation are more contributive to the classification; 2) through the CAM, the proposed method effectively fuses the spectral-spatial features and generates more relevant and discriminative features.

F. Impact of Training Ratios

In actual applications, the number of training samples is an important factor for classification accuracy. In this section, we explore the performance of the proposed method with different ratios of labeled samples. The ratios of labeled samples are set as 2%, 5%, 10%, 15%, and 20%, respectively. The result of 2% on the IP dataset is null because some categories of samples are so small that there are no samples. Table X shows the classification performance of the proposed method on the three datasets with different percentages of labeled samples as training samples.

In Table X, we can observe that the classification accuracies improve as the ratios of labeled samples increase. This proves that the spectral-spatial features learned by the proposed method are effective for HSI classification. We also find that when a small number of samples (the IP dataset is 5%, the UP and SV datasets are 2%) are available, satisfactory results can be obtained by the proposed method as well. Achieving good results with fewer training samples is crucial for HSI classification since the labeled samples are often difficult to collect in actual situations.

V. Conclusion

In this article, we propose an end-to-end hyperspectral image classification method by introducing a CAM into 3D CNN to enhance the classification accuracy of hyperspectral images. Specifically, this method effectively learns the internal correlation between the central pixel and its neighboring pixels in a subcube sample and generates more discriminative spectral-spatial features. Experimental results demonstrate that our method has exceeded several state-of-the-art HSI classification methods based on deep learning, and it still retains its functionality even with an inadequate number of labeled samples. In addition, the method effectively fuses the basic spectral-spatial features extracted by the 3D CNN module, which significantly reduces the number of parameters and improves the training efficiency.

References

[1] D. Chutia, D. K. Bhattacharyya, K. K. Sarma, R. Kalita, and S. Sudhakar, “Hyperspectral remote sensing classifications: A perspective survey,” *Trans. GIS*, vol. 20, no. 4, pp. 463–490, 2016.

[2] H. Lin, 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. 99, pp. 1579–1597, Mar. 2018.

[3] M. Paoletti, J. Haut, J. Plaza, and A. Plaza, “Deep learning classifiers for hyperspectral imaging: A review,” *ISPRS J. Photogrammetry Remote Sens.*, vol. 158, pp. 279–317, 2019.

[4] F. Zhang, B. Du, and L. Zhang, “Scene classification via a gradient boosting random convolutional network framework,” *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 3, pp. 1793–1802, Mar. 2016.

[5] X. Yang and Y. Yu, “Estimating soil salinity under various moisture conditions: An experimental study,” *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 5, pp. 2525–2533, May 2017.

[6] Awad and M. , “Sea water chlorophyll-a estimation using hyperspectral images and supervised artificial neural network,” *Ecological Inform.*, vol. 24, pp. 60–68, 2014.

[7] M. Dalponte, H. O. Orka, T. Gobakken, D. Gianelle, and E. Naesset, “Tree species classification in boreal forests with hyperspectral data,” *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 5, pp. 2632–2645, May 2013.

[8] E. Saralioglu, E. T. Gormus, and O. Gungor, “Mineral exploration with hyperspectral image fusion,” in *Proc. Signal Process. Commun. Appl. Conf.*, 2016, pp. 1281–1284.

[9] 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.

[10] Z. Gong, P. Zhong, Y. Yu, W. Hu, and S. Li, “A CNN with multiscale convolution and diversified metric for hyperspectral image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 6, pp. 3599–3618, Jun. 2019.

[11] S. Li, W. Song, L. Fang, Y. Chen, and J. A. Benediktsson, “Deep learning for hyperspectral image classification: An overview,” *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 6690–6709, Sep. 2019.

[12] H. Zhang, Y. Li, Y. Jiang, P. Wang, and C. Shen, “Hyperspectral classification based on lightweight 3D-CNN with transfer learning,” *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 8, pp. 5813–5828, Aug. 2019.

[13] H. Wei, H. Yangyu, W. Li, Z. Fan, and L. Hengchao, “Deep convolutional neural networks for hyperspectral image classification,” *J. Sensors*, vol. 2015, pp. 1–12, 2015.

[14] Y. Zhan, D. Hu, Y. Wang, and X. Yu, “Semisupervised hyperspectral image classification based on generative adversarial networks,” *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 2, pp. 1–5, Feb. 2018.

[15] L. Ying, Z. Haokui, and S. Qiang, “Spectral-spatial classification of hyperspectral imagery with 3 d convolutional neural network,” *Remote Sens.*, vol. 9, no. 67, pp. 1–21, 2017.

[16] X. Mei *et al.*, “Spectral-spatial attention networks for hyperspectral image classification,” *Remote Sens.*, vol. 11, no. 963, pp. 1–18, 2019.

[17] 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. 99, pp. 8065–8080, Oct. 2019.

[18] Z. Zheng, Y. Zhong, A. Ma, and L. Zhang, "FPGA: Fast patch-free global learning framework for fully end-to-end hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 8, pp. 5612–5626, Aug. 2020.

[19] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.

[20] K. He, X. Zhang, S. Ren, and S. Jian, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[21] C. Yang, X. Lu, Z. Lin, E. Shechtman, O. Wang, and H. Li, "High-resolution image inpainting using multi-scale neural patch synthesis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4076–4084.

[22] W. Li, G. Wu, F. Zhang, and Q. Du, "Hyperspectral image classification using deep pixel-pair features," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 2, pp. 844–853, Feb. 2017.

[23] K. S. Charmisha, V. Sowmya, and K. P. Soman, "Dimensionally reduced features for hyperspectral image classification using deep learning," in *Proc. Int. Conf. Commun. Cyber Phys. Eng.*, 2018, pp. 171–179.

[24] P. Ghamisi *et al.*, "New frontiers in spectral-spatial hyperspectral image classification: The latest advances based on mathematical morphology, Markov random fields, segmentation, sparse representation, and deep learning," *IEEE Geosci. Remote Sens. Mag.*, vol. 6, no. 3, pp. 10–43, Sep. 2018.

[25] J. Yue, W. Zhao, S. Mao, and H. Liu, "Spectral-spatial classification of hyperspectral images using deep convolutional neural networks," *Remote Sens. Lett.*, vol. 6, no. 4–6, pp. 468–477, 2015.

[26] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, "Deep feature extraction and classification of hyperspectral images based on convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 10, pp. 6232–6251, Oct. 2016.

[27] 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.

[28] J. Zhu, L. Fang, and P. Ghamisi, "Deformable convolutional neural networks for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 8, pp. 1254–1258, Aug. 2018.

[29] L. Fang, G. Liu, S. Li, P. Ghamisi, and J. A. Benediktsson, "Hyperspectral image classification with squeeze multibias network," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 3, pp. 1291–1301, 2019.

[30] 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.

[31] 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, 2019.

[32] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," Ser. Proceedings of Machine Learning Research, D. Precup and Y.W. Teh, Eds., vol. 70. Sydney, Australia: International Convention Centre, Aug. 2017, pp. 1243–1252.

[33] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu, "Recurrent models of visual attention," in *Advances in Neural Information Processing Systems*, Z. M. Ghahramani, C. Welling, N. Cortes, D. Lawrence, and K. Q. Weinberger, Eds. Montreal, QC, Canada: Curran Associates, Inc., 2014, pp. 2204–2212.

[34] X. Chu, W. Yang, W. Ouyang, C. Ma, A. L. Yuille, and X. Wang, "Multi-context attention for human pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5669–5678.

[35] X. Zhang, T. Wang, J. Qi, H. Lu, and G. Wang, "Progressive attention guided recurrent network for salient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 714–722.

[36] J. Fu *et al.*, "Dual attention network for scene segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3146–3154.

[37] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-excitation networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 8, pp. 2011–2023, Aug. 2020.

[38] B. Fang, Y. Li, H. Zhang, and J. Chan, "Hyperspectral images classification based on dense convolutional networks with spectral-wise attention mechanism," *Remote Sens.*, vol. 11, no. 159, pp. 1–18, 2019.

[39] W. Ma, Q. Yang, Y. Wu, W. Zhao, and X. Zhang, "Double-branch multi-attention mechanism network for hyperspectral image classification," *Remote Sens.*, vol. 11, no. 11, 2019, Art. no. 1307.

[40] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[41] S. I. Krizhevsky Alex and H. G, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1–9.

[42] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.

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

[44] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Comput. Sci.*, 2014, pp. 1–15.

[45] 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.

Zhengang Zhao received the B.Sc. and M.Sc. degrees in computer science and technology from the Hebei Normal University, Shijiazhuang, China, in 2010 and 2013, respectively. He is working toward the Ph.D. degree at the School of Artificial Intelligence, Beijing Normal University, Beijing, China, in 2018.

His research interests include signal processing, remote image processing, and deep learning.

Dan Hu received the B.Sc. and M.Sc. degrees in mathematics from the Sichuan Normal University, Chengdu, China, in 1999 and 2002, respectively, and the Ph.D. degree in applied mathematics from Beijing Normal University, Beijing, China.

She is currently a Research Scholar with the Department of Radiology and BRIC, University of North Carolina at Chapel Hill, NC, USA, working in the field of machine learning.

Hao Wang received the B.Sc. degree in computer science and technology from the Beijing Language and Culture University, Beijing, China, in 2019. He is currently working toward the graduation machine learning at the School of Artificial Intelligence, Beijing Normal University, Beijing, China . His research interests include remote image analyses, deep learning, and pattern recognition.

Xianchuan Yu (Senior Member, IEEE) received the Ph.D. degree in mathematical geology from Jilin University, Jilin, China, in 1995.

Currently, he is a Professor and an Academic Leader with the School of Artificial Intelligence, Beijing Normal University, Beijing, China. His research interests include blind source separation, remote image processing, and mineral resources appraisal.

Dr. Yu has been a Vice Director of the China Mathematical Geology and Geological Information Processing Professional Committee, since 2012.