# Cascade Residual Capsule Network for Hyperspectral Image Classification

Zhiming Mei, Zengshan Yin, Xinwei Kong, Long Wang, and Han Ren

***Abstract*—The convolution neural network (CNN) has recently shown the good performance in hyperspectral image (HSI) classification tasks. Many CNN-based methods crop image patches from original HSI as inputs. However, the input HSI cubes usually contain background and many hyperspectral pixels with different land-cover labels. Therefore, the spatial context information on objects of the same category is diverse in HSI cubes, which will weaken the discrimination of spectral–spatial features. In addition, CNN-based methods still face challenges in dealing with the spectral similarity between HSI cubes of spatially adjacent categories, which will limit the classification accuracy. To address the aforementioned issues, we propose a cascade residual capsule network (CRCN) for HSI classification. First, a residual module is designed to learn high-level spectral features of input HSI cubes in the spectral dimension. The residual module is employed to solve the problem of the spectral similarity between HSI cubes of spatially adjacent categories. And then two 3-D convolution layers are exploited to extract high-level spatial–spectral features. Finally, a capsule structure is developed to characterize spatial context orientation representations of objects, which can effectively deal with the diverse spatial context information on objects of the same category in HSI cubes. The capsule module is composed of two 3-D convolution layers and the capsule structure, which is connected to the residual module in series to construct the proposed CRCN. Experimental results on four public HSI datasets demonstrate the superiority of the proposed CRCN over six state-of-the-art models.**

***Index Terms*—Cascade residual capsule network (CRCN), convolution neural network (CNN), high-level spatial–spectral feature extraction, hyperspectral image (HSI) classification, spatial context orientation representations.**

## I. Introduction

HYPERSPECTRAL image (HSI) usually contains hundreds of spectral channels from which rich spectral features can be extracted for the subsequent classification tasks. However, it is insufficient to obtain an excellent HSI classification performance solely via spectral signatures [1]–[4]. Subsequently, spatial–spectral methods are proposed to incorporate spatial position information into spectral signatures for HSI classification [5]–[7]. Many spatial–spectral methods take HSI cubes cropped from original HSI as inputs. HSI cubes are samples in HSI datasets and each HSI cube is assumed to have the same land-cover label with its center pixel. But HSI cubes usually contain interfering hyperspectral pixels whose land-cover labels are different from that of the center hyperspectral pixel of HSI cubes. Leading to diverse spatial context information on objects of the same category in HSI cubes, which will weaken the discrimination of spatial–spectral features. Besides, the issue of spectral similarity between HSI cubes of spatially adjacent categories are remained to be effectively addressed for HSI classification.

Recently, the deep learning model has shown its powerful capability to extract discriminative features for HSI classification [8]–[12]. Stacked autoencoders [13], sparse autoencoders [14], and deep belief networks [15] were first adopted to characterize spectral features. And Wu *et al.* [16] employed a deep convolution recurrent neural network to learn spectral features. However, these models did not consider the significant spatial context information on objects. Therefore, Chen *et al.* [5] proposed a three-dimensional (3-D) CNN to incorporate spatial context information into spectral signatures and extract spatial–spectral features, making great breakthrough in HSI classification.

In order to improve the performance of HSI classification, many 3-D CNN-based models have been proposed [17]–[22]. For instance, Li *et al.* [23] employed a pixel-pair method to increase the number of training samples. And in [24], a model was designed to merge spatial and spectral features extracted by a 3-D CNN and a balanced local discriminant embedding approach, respectively. In addition, Song *et al.* [25] proposed a deep feature fusion network (DFFN) to merge features learned by different hierarchical layers. And Fang *et al.* [26] built a squeeze multibias network (SMBN) to decouple features into multiple maps. Besides, Zhang *et al.* [27] exploited a 3-D CNN-based network to extract hierarchical spectral and spatial features. Zhong *et al.* [22] proposed a spatial–spectral residual network (SSRN) to extract spatial–spectral features from raw HSI cubes. In [28], a spatial–spectral attention network (SSAN) was employed to extract spatial–spectral features from attention areas of HSI cubes. And Zhang *et al.* [29] developed a 3-D CNN-based framework to encode semantic context-aware representations for obtaining spatial–spectral features. In [30], spatial

Manuscript received December 28, 2021; revised February 14, 2022, March 7, 2022, and March 27, 2022; accepted April 8, 2022. Date of publication April 12, 2022; date of current version May 3, 2022. This work was supported in part by the National Key Research and Development Program of the Ministry of Science and Technology under Grant 2017YFB0502902 and in part by the Shanghai Science and Technology Talents Program under Grant 18QA1404000. *(Corresponding author: Zhiming Mei.)*

Zhiming Mei and Han Ren are with the Innovation Academy for Microsatellites, Chinese Academy of Sciences, Shanghai 201210, China, and also with the University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: meizhm@shanghaitech.edu.cn; jamesrh@126.com).

Zengshan Yin and Long Wang are with the Innovation Academy for Microsatellites, Chinese Academy of Sciences, Shanghai 201210, China (e-mail: yinzs@microsate.com; wangprchina@hotmail.com).

Xinwei Kong is with the University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: kongxw@mail.ustc.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2022.3166972


Fig. 1. Two categories "Soybean-notill" and "Grass-pasture" of the Indian Pines image are used for illustration. The borders of the two categories are marked by red and green, respectively. As shown in (b) and (c), the spatial context information on objects of the two categories in four cropped HSI cubes are slightly different. This will produce diverse spatial context information on objects of the same category in HSI cubes of the Indian Pines datasets. (a) Part of ground truth of the Indian Pines image. (b) Land-cover map of four cropped cubes of the "Grass-pasture" category. (c) Land-cover map of four cropped cubes of the "Soybean-notill" category. (d) Color information.

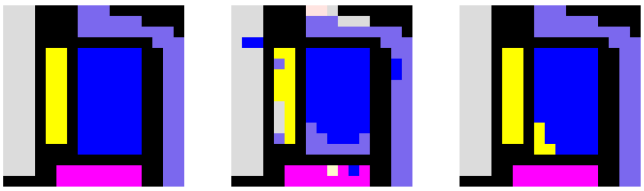



Fig. 2. Two classification submaps obtained by 3-D CNN and the CRCN on the Indian Pines image are selected for illustration. It mainly contains two spatially adjacent categories "Soybean-notill" and "Grass-pasture," which are labeled by blue and yellow, respectively. Spectral signatures of cropped HSI cubes between the two categories are quite similar. As can be seen in (b), the classification areas of "Soybean-notill" and "Grass-pasture" categories are greatly affected by the spectral similarity between them. (a) Part of ground truth of the Indian Pines image. (b) Classification sub-map of the Indian Pines image obtained by 3-D CNN. (c) Classification submap of the Indian Pines image obtained by the CRCN.

and spectral features extracted by a 3-D CNN-based network were enhanced by some multiple scale covariance functions. And in [31], spatial–spectral features of specific bands were extracted by a 3-D CNN-based network for HSI classification.

Moreover, many auxiliary methods were incorporated into the 3-D CNN to learn spatial–spectral representations [32]–[34]. In [35], an end-to-end deconvolution network with skip architecture was proposed to learn spatial–spectral features. And Cao *et al.* [36] combined the Markov random fields (MRF) with a 3-D CNN-based network to learn posterior distribution of all categories. In [37], two CNN-based networks were combined with a novel feature fusion scheme to learn spatial–spectral features. In addition, dimension reduction based methods were proposed to learn the intrinsic discriminative spatial–spectral representations for HSI classification [38], [39]. However, the diverse spatial context information on objects of the same category in HSI cubes will weaken the discrimination of spectral–spatial features, which is illustrated in Fig. 1. Besides, the spectral similarity between HSI cubes of spatially adjacent categories illustrated in Fig. 2 is still a challenging problem.

In this article, we propose a cascade residual capsule network (CRCN) to solve the aforementioned two problems. The CRCN consists of a residual module and a capsule module. The residual module is composed of four residual units, which is exploited to learn high-level spectral features in the spectral dimension. Each residual block contains three convolution layers with shortcut connection [40], batch normalization (BN), and activation operation. In addition, the residual module is able to prevent the degradation of the CRCN due to gradient vanishing. The capsule module contains two 3-D convolution layers and a capsule structure. The capsule structure is developed on capsule networks (CapsNets) [41]–[43], which is employed to characterize spatial context orientation representations in the spatial dimension.

The contributions of this article are summarized as follows.

1) A residual module is used to learn high-level spectral features in the spectral dimension, which is exploited to deal with the spectral similarity between HSI cubes of spatially adjacent categories.
2) A capsule module is exploited to characterize spatial context orientation representations in the spatial dimension, which is designed to deal with the diverse spatial context information.
3) We join the residual module and the capsule module together into a whole serial network named CRCN with a new strategy that high-level spectral features are first extracted in the spectral dimension and then spatial context orientation representations are learned in the spatial dimension.

The rest of this article is organized as follows. In Section II, we briefly review the 3-D CNN model and the 3-D CNN-based method for HSI classification. And then we introduce the CRCN model in Section III. The experimental results are given in Section IV. Finally, Section V concludes this article.

## II. Related Works

CNN is an effective model for extracting features of images. And it has achieved outstanding performance in image classification tasks [44]–[47]. A 2-D convolution layer convolves feature maps in the spatial dimension with 2-D convolution kernels. While a 3-D convolution layer convolves feature maps in both the spatial dimension and the spectral dimension with 3-D convolution kernels. The difference between the 2-D convolution and 3-D convolution are shown in Fig. 3.

### A. 3-D CNN

The key part of 3-D CNN is the convolution layer, which exploits convolution kernels to characterize feature maps [48]. In the $k$th convolution layer, the value at location $(x, y, z)$ in the $l$th feature map obtained by convolution kernel $K$ is expressed as follows:

$$U_{kl}^{xyz} = \delta\left(\sum_{n=0}^{N_f} f(U) + B_{kl}\right) \tag{1}$$

$$f(U) = \sum_{w=0}^{W_K}\sum_{h=0}^{H_K}\sum_{g=0}^{C} K_{kl}^{whg} U_{(k-1)n}^{(xP_w+w)(yP_h+h)(zP_g+g)} \tag{2}$$

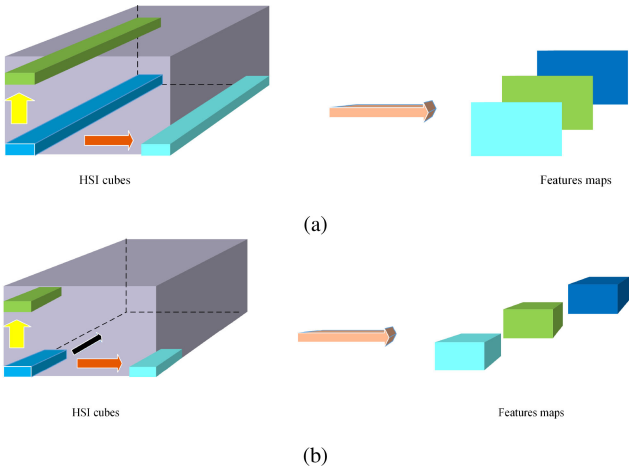



Fig. 3. Comparison of (a) 2-D convolution and (b) 3-D convolution.

where $N_f$ is the number of feature maps in the $(k-1)$th convolution layer. $W_K$ and $H_K$ represent the width and height of convolution kernel $K$, respectively. $C$ refers to the number of channels and $K_{kl}^{whg}$ denotes the value of convolution kernel $K$ at index $(w, h, g)$. $P_w$, $P_h$, and $P_g$ are the sliding intervals of convolution kernel $K$ in the spatial dimension and the spectral dimension. And $B_{kl}$ is the bias parameter of the $l$th feature map in the $k$th layer. Function $\delta(\cdot)$ is employed to active the $k$th convolution layer. Generally, rectified linear unit (ReLU) [49] is used as the activation function, which is denoted as

$$\delta(x) = \max(x, 0). \tag{3}$$

In 3-D CNN, the covariance shift of feature maps may lead to gradient vanishing [50]. BN layer is exploited to reduce the covariance shift of feature maps, which is usually connected next to a convolution layer. The function of BN layer is expressed as follows:

$$\mathrm{BN}(F_m) = \gamma \cdot \frac{F_m - \mu}{\sqrt{\delta^2 + \epsilon}} + \beta \tag{4}$$

where $F_m$ is the input feature maps and $\epsilon$ is a constant which approaches to 0. $\mu$ and $\delta$ represent the mean and standard deviation of $F_m$, respectively. $\gamma$ and $\beta$ are weight parameters, which are learned in the training process.

The 3-D max pooling layer is usually used to reduce feature map, which is represented as

$$x_{whg} = \max(S[w : \hat{w}][h : \hat{h}][g : \hat{g}]) \tag{5}$$

$$\hat{w} = P_w \times w + R_w, \hat{h} = P_h \times h + R_h, \hat{g} = P_g \times g + R_g \tag{6}$$

where $S$ is a feature map, $(R_w, R_h, R_g)$ and $(P_w, P_h, P_g)$ are the pooling kernel and stride vectors of the 3-D max pooling layer, respectively. $S[w : \hat{w}][h : \hat{h}][g : \hat{g}]$ denotes all elements from index $(w, h, g)$ to $(\hat{w}, \hat{h}, \hat{g})$ in the spatial dimension and spectral dimension of $S$.

### B. 3-D CNN-Based HSI Classification

The last layer of 3-D CNN is usually activated by the Softmax$(\cdot)$ function to produce the output vector $o$, which is represented as

$$\mathrm{Softmax}(o_i) = \frac{e^{o_i}}{\sum_{i=0}^{N-1} e^{o_i}} \tag{7}$$

where $o_i$ is the element of vector $o$ at index $i$ and $N$ is the number of categories. The cross-entropy loss function is usually exploited by 3-D CNN for HSI classification, which is defined as

$$\mathrm{CE} = -\sum_{i=0}^{N-1} T_i \log(\mathrm{Softmax}(o_i)) \tag{8}$$

where $T_i = 1$ if an input HSI cube is labeled as $i$th category. The value of the defined loss function will be decreased by the stochastic gradient descent (SGD) algorithm until it converges [51]–[53].

## III. CRCN Model for HSI Classification

An input HSI cube $X \in R^{H \times W \times C}$ consists of $H \times W$ hyperspectral pixels. And each hyperspectral pixel $x_{i,j} \in R^{1 \times 1 \times C}$ contains $C$ spectral channels. To solve the problem of diverse spatial context information on land-cover areas with the same label and spectral similarity between HSI cubes of spatially adjacent categories, we propose a CRCN to learn spectral features first in spectral dimension by a residual module and then characterize spatial context orientation representations in spatial dimension via a capsule module.

The spectral similarity between HSI cubes of spatially adjacent categories is defined as follows:

$$\mathrm{SS}(A, B) = \frac{\sum_{i=1}^{N_A} \sum_{j=1}^{N_B} \mathrm{Num}(X_i, X_j)}{H \times W \times N_A \times N_B} \tag{9}$$

$$\mathrm{Num}(X_i, X_j) = \sum_{n=1}^{N} \min(X_{in}, X_{jn}) \tag{10}$$

where $A$ and $B$ are any two adjacent categories of all $N$ categories, $N_A$ and $N_B$ are the number of HSI cubes of categories $A$ and $B$, respectively. $X_i$ is an HSI cube of category $A$ and $X_j$ is an HSI cube of category $B$. $X_{\mathrm{in}}$ represents the number of hyperspectral pixels of the $n$th category in HSI cube $X_i$. $X_{jn}$ represents the number of hyperspectral pixels of the $n$th category in HSI cube $X_j$. The $n$th category means each category of all $N$ categories, and n ranges from $[1, N]$.

The network structure of the CRCN consists of the residual module and the capsule module, which is shown in Fig. 4. The structures and parameters of the residual and capsule module are different from other residual and CapsNets proposed for HSI classification. The outputs of the CRCN are the length of activity vectors and a reconstructed HSI cube. The detailed structure and specific parameters of the residual module and the capsule module will be given in experimental settings of Section IV.

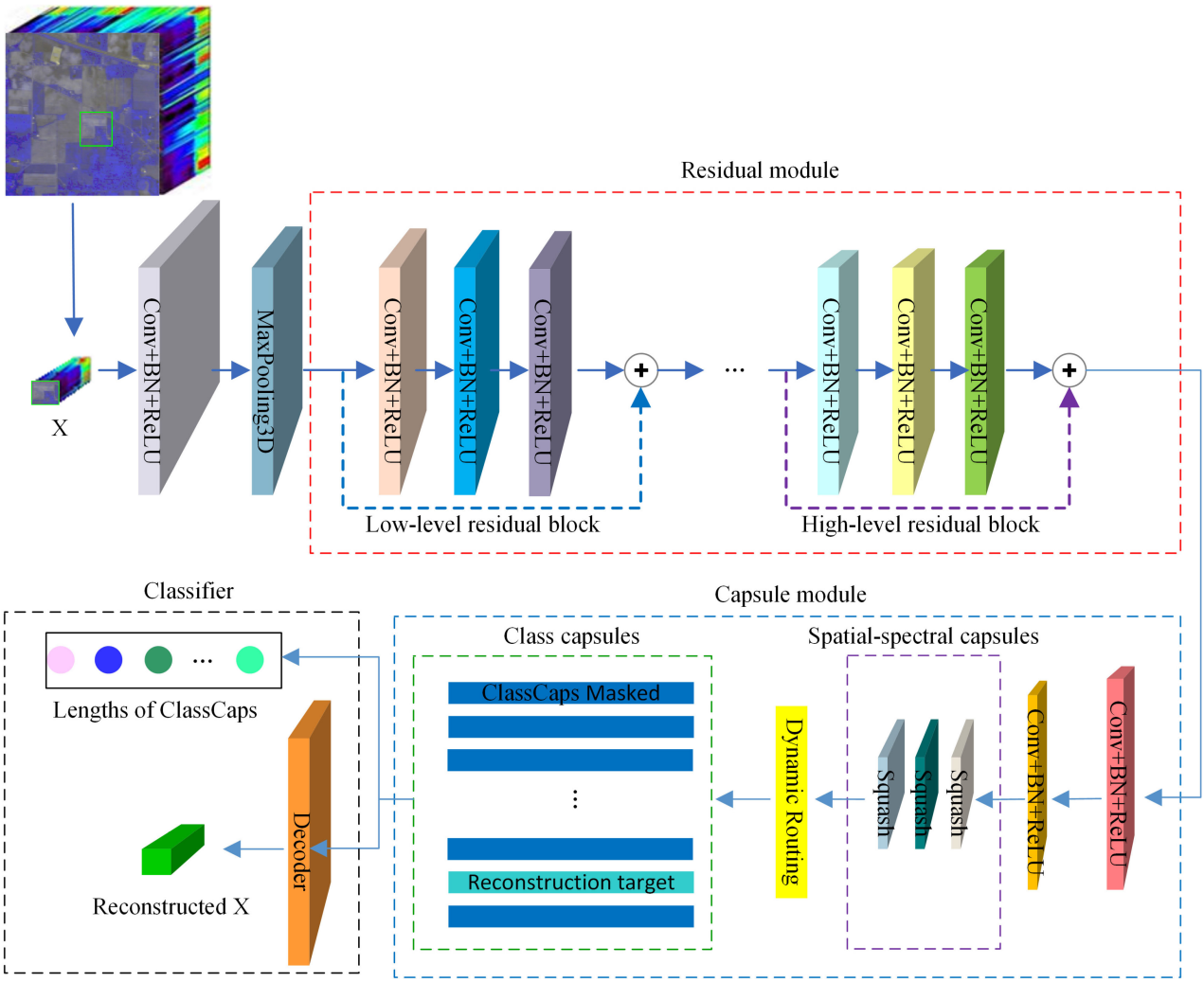



Fig. 4. Network structure of the CRCN model. The Indian Pines image is used as an instance to show the network structure.

### A. *Residual Module*

In the residual module, four residual units are connected in sequence and each residual unit contains three residual blocks. Each residual block is composed of three BN, ReLU, and 3-D convolution layers, which is shown in Fig. 5. The spatial size of feature maps are invariant while the number of spectral channels of feature maps are reduced from low-level to high-level residual blocks.

A residual block is defined as

$$y = F(x, \{R_i\}) + R_s x, i = 1, 2, 3 \tag{11}$$

$$F(x, \{R_i\}) = R_3\sigma(R_2\sigma(R_1\sigma(x) + B_1) + B_2) + B_3 \tag{12}$$

where $x$ is the input feature map and $y$ is the corresponding output. Function $F(\cdot)$ is exploited to learn the mapping of the residual block. And $R_i$ is the weight parameter matrix of the $i$th 3-D convolution layer. $R_s$ is employed to adjust the dimension of $x$ for the shortcut connection with $y$. $B_i$ and $\sigma(\cdot)$ are bias and the ReLU activation function, respectively. The identity mapping implemented by shortcut connection is able to prevent the gradients of parameters from vanishing, which will not add extra variables or computation complexity.

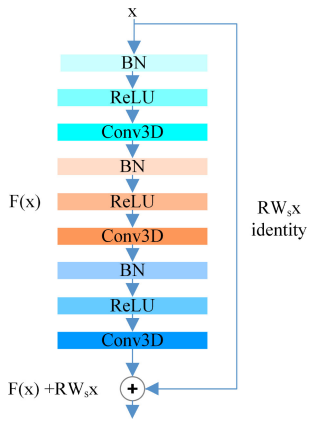



Fig. 5. Sketch of one residual block of the residual module.

In the residual module, residual blocks of different levels stacked in series. Input feature map $x$ is mapped as low-level

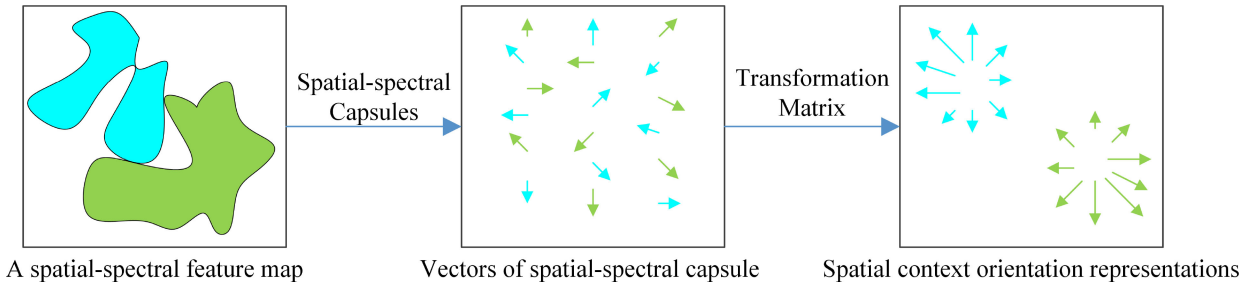



Fig. 6. Spatial–spectral feature map, which contains features of HSI cubes of two categories is used for illustration. Feature regions of HSI cubes of the two categories are marked as blue and green, respectively. The feature map is converted into spatial–spectral capsules, which contain vectors of recognized feature regions of HSI cubes. And then vectors of spatial–spectral capsules are linearly transformed by transformation matrix to characterize spatial context orientation representations. The lengths of the spatial context orientation representations of blue and green feature regions denote the probabilities that land-cover areas of the two categories exist, respectively.

spectral features in low-level residual blocks and then mapped as high-level spectral features in high-level residual blocks. The residual module integrates low, mid, and high-level spectral features extracted from input HSI cubes. And the levels of spectral features can be enriched by the number of residual blocks [40].

### *B. Capsule Module*

As illustrated in Fig. 4, two 3-D convolution layers are first used to extract high-level spatial–spectral features in the capsule module. And then spatial–spectral feature maps are converted into spatial–spectral capsules, which are composed by vectors of instantiated parameters of recognized fragments of HSI cubes. Afterwards, vector $u_i = [u_{i,1}, u_{i,2}, \ldots, u_{i,d}] \in R^d$ of spatial–spectral capsules is linearly transformed by the transformation matrix $W_{ij}$ to characterize spatial context orientation representations. The process is illustrated in Fig. 6.

The transformation matrix $W_{ij}$ is able to learn the intrinsic spatial relationship between the part of objects in HSI cubes and the whole objects in original HSI. And the transformation matrix $W_{ij}$ constitutes invariant spatial knowledge, which automatically generalizes to fragments of objects in HSI cubes. The invariant spatial knowledge unifies the diverse spatial context information on objects of the same category in HSI cubes.

In fact, the elements of $u_i$ represent different properties of spatial context information on objects in HSI cubes. Transformed vector $u_{ij}$ makes predictions for vector $u_i$ of spatial–spectral capsules via transformation matrix. It is obtained by multiplying the transformation matrix $W_{ij}$ by vector $u_i$ of spatial–spectral capsules, which is expressed as

$$u_{ij} = W_{ij} u_i. \tag{13}$$

Vector $s_j = [s_{j,1}, s_{j,2}, \ldots, s_{j,f}] \in R^f$ is the weighted sum of all transformed vectors, which is denoted as follows:

$$s_j = \sum_i c_{ij} u_{ij} \tag{14}$$

where $c_{ij}$ is the coupling coefficient, which is initialized by the Softmax$(\cdot)$ function at the beginning of dynamic routing process. The initial value $b_{ij}$ of the coupling coefficient $c_{ij}$ represents the log prior probabilities that the $i$th spatial–spectral capsule is coupled to the $j$th class capsule. The initialization of

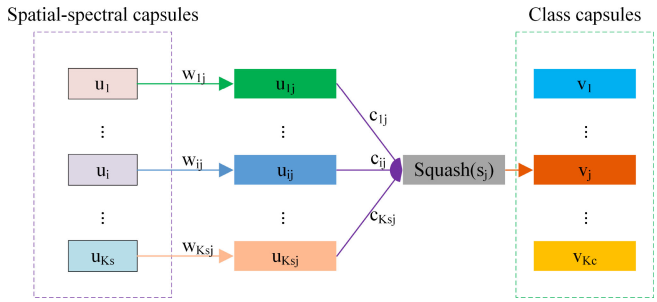



Fig. 7. One step of the iterative dynamic routing process. All vectors of spatial–spectral capsules are transformed by transformation matrices. The transformed vectors are weighted and summed to obtain vectors of class capsules. Every vector of class capsule is activated by the squash function. Activity vectors of class capsules have a big scalar product with the prediction coming from the vectors of spatial–spectral capsules.

$c_{ij}$ is expressed as

$$c_{ij} = \frac{e^{b_{ij}}}{\sum_k e^{b_{ij}}}. \tag{15}$$

To obtain the $j$th activity vector $v_j$ of class capsule, the length of vector $s_j$ is scaled down via a nonlinear squash function, which is used as activation function. Unlike classical activation function, such as ReLU, the squash function normalizes vector $s_j$ to unit vector with a squeeze coefficient, which is defined as

$$v_j = \frac{\|s_j\|^2}{1 + \|s_j\|^2} \frac{s_j}{\|s_j\|} \tag{16}$$

where $\frac{\|s_j\|^2}{1+\|s_j\|^2}$ is the squeeze coefficient.

The sketch of the dynamic routing process is shown in Fig. 7. The elements of the transformation matrix $W_{ij}$ are network parameters, which are updated in the training process. The coupling coefficients $c_{ij}$ are initialized and determined in the iterative dynamic routing process. The key point of the dynamic routing algorithm is the agreement between the current squashed vector $v_j$ of the $j$th class capsule and the transformed vector $u_{ij}$ of the $i$th spatial–spectral capsule. The agreement is actually an iterative and consistent transformation via the coupling coefficient $c_{ij}$ between transformed vector $u_{ij}$ and squashed vector $v_j$. The initial value $b_{ij}$ of coupling coefficient is assigned to 0 in the beginning of the dynamic routing process. The scalar product

**Algorithm 1:** Dynamic Routing.

**Require:** The activity vector of spatial-spectral capsule $u_{ij}$, iteration times $r$, and the initial coupling coefficient $b_{ij}$ between the $i$th spatial-spectral capsule and the $j$th class capsule.

1: $b_{ij} = 0$
2: **for** $p = 0$ to $r$ **do**
3: for all spatial-spectral capsules:$c_i = softmax(b_i)$
4: for all class capsules:$s_j = \sum_i c_{ij} u_{ij}$
5: for all class capsules: $v_j = squash(s_j)$
6: for all spatial-spectral capsules and class capsules:$b_{ij} = b_{ij} + u_{ij} v_j$
7: **end for**
8: **return** $v_j$

disparity between activity vectors is expanded by the dynamic routing process. The detailed procedure of the dynamic routing is described in Algorithm 1.

The length of activity vector is exploited to predict the category of an input HSI cube. Long activity vectors of class capsules are expected to be more related to spatial context orientation presentations, which are characterized by transformation matrix. Therefore, we try to minimize a separate margin loss function to make the length of activity vectors of class capsule whose indexes are the same with the label of input HSI cube longer. If the predicted category corresponds to the index of short vector, the margin loss will be large. On the contrary, if the predicted category corresponds to the index of long vector, the margin loss will be small. The margin loss function is defined as

$$L_c = \sum_{i=1}^{N} (T_i \max(0, \alpha^+ - \|v_i\|)^2 + \lambda(1 - T_i) \max(0, \|v_i\| - \alpha^-)^2) \tag{17}$$

where $T_i$ is the element of land-cover label vector of the input HSI cube, $\alpha^+$ and $\alpha^- = 1 - \alpha^+$ are threshold values. And $N$ is the number of categories. Weight hyperparameter $\lambda$ can prevent the length of activity vectors from being shrunk in the initial stage of training process. The value of $T_i$ obeys the following distribution:

$$T_i = \begin{cases} 1, & i = t \\ 0, & i \neq t \end{cases} \tag{18}$$

where $t$ denotes the corresponding land-cover category of an input HSI cube.

### C. HSI Classifier

A convolution decoder is employed to reconstruct the input HSI cube from the reconstruction target of class capsules, which is use as a regularization item. The decoder consists of several 3-D deconvolution layers followed with BN layers and ReLU activation function. The input of the decoder is the reconstruction target, which is obtained by masking all activity vectors of class capsules with land-cover label of the input HSI cube. We exploit the mean squared error between input HSI cubes and reconstructed outputs as the reconstruction loss function which is defined as

$$L_r = \frac{1}{m} \sum_{i=1}^{m} \|D(X_i) - X_i\|_2^2 \tag{19}$$

where $m$ is the number of HSI cubes in one batch. Function $D(\cdot)$ represents the convolution decoder. And $X_i \in R^{H \times W \times C}$ is the $i$th HSI cube. The loss function of the CRCN is the weighted sum of the reconstruction and the margin loss functions, which is denoted as

$$L = L_r + \theta L_c \tag{20}$$

where $\theta$ is a weight hyperparameter. And the classifier of the CRCN is defined as

$$\hat{j} = \arg \max_{j \in 1, \ldots, K_c} \|v_j\| \tag{21}$$

where $K_c$ is the number of vectors of class capsules and $\hat{j}$ is the predicted label of an input HSI cube. All network parameters of the CRCN are tuned automatically by the backpropagation and SGD algorithm [54]–[56], which is exploited to optimize the loss function $L$.

## IV. Experimental Results and Analysis

Sixty-five experiments on each comparison method were conducted to evaluate the classification performance of the CRCN model on each public HSI dataset.

### A. Hyperspectral Image Datasets

The Indian Pines image is captured by the AVIRIS sensor on the Indian Pines test site in 1992. It contains 220 spectral bands and 16 land-cover categories. The wavelengths of its spectral bands range from $0.4 \times 10^{-6}$ to $2.5 \times 10^{-6}$ m. And 20 water absorption bands are removed in experiments. The spatial size and resolution of the Indian Pines image are $145 \times 145$ and 20 m/pixel, respectively. A total of 10% pixels of each land-cover category are randomly selected for training and the rest of them are used for testing. The detailed quantities are listed in Table I. The false-color composite and the corresponding ground truth are shown in Fig. 8.

The University of Pavia image is collected by the ROSIS sensor on Northern Italy. It contains 103 spectral bands and 9 land-cover categories. The wavelengths of its spectral bands range from $0.38 \times 10^{-6}$ to $0.86 \times 10^{-6}$ m. The spatial size and resolution of the University of Pavia image are $610 \times 340$ and 1.3 m/pixel, respectively. A total of 2% pixels of each land-cover category are randomly chosen as training samples and the rest of them are used as testing samples. The detailed quantities are listed in Table II. The false-color composite of the University of Pavia image and the corresponding ground truth are shown in Fig. 9.

The Salinas image is acquired by the AVIRIS sensor on Salinas Valley. It contains 224 spectral bands and 16 land-cover

TABLE I
NUMBERS OF TRAINING AND TESTING SAMPLES OF THE INDIAN PINES DATASET (10%, 90%)

| Class | Name | Training | Testing |
|---|---|---|---|
| 1 | Alfalfa | 4 | 42 |
| 2 | Corn-notill | 142 | 1286 |
| 3 | Corn-mintill | 83 | 747 |
| 4 | Corn | 23 | 214 |
| 5 | Grass-pasture | 48 | 435 |
| 6 | Grass-trees | 73 | 657 |
| 7 | Grass-pasture-mowed | 2 | 26 |
| 8 | Hay-windrowed | 47 | 431 |
| 9 | Oats | 2 | 18 |
| 10 | Soybean-notill | 97 | 875 |
| 11 | Soybean-mintill | 245 | 2210 |
| 12 | Soybean-clean | 59 | 534 |
| 13 | Wheat | 20 | 185 |
| 14 | Woods | 126 | 1139 |
| 15 | Buildings-Grass-Trees-Drives | 38 | 348 |
| 16 | Stone-Steel-Towers | 9 | 84 |
| | Total | 1018 | 9231 |

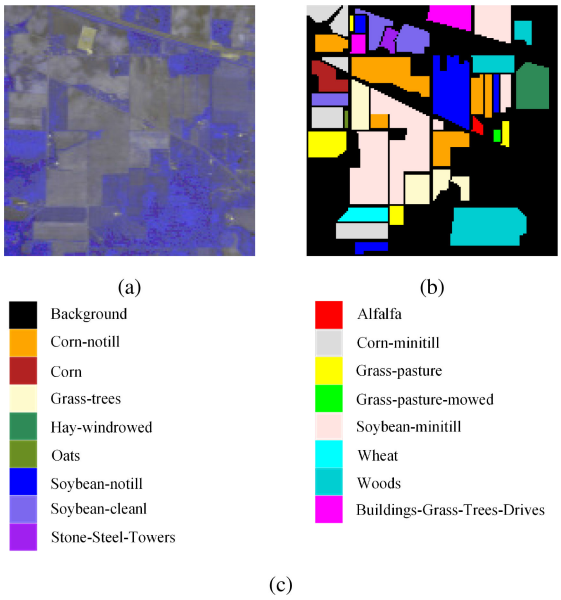



Fig. 8. Indian Pines image. (a) False color composite. (b) Ground truth. (c) Color code.

categories. The wavelengths of its spectral bands range from $0.4 \times 10^{-6}$ to $2.5 \times 10^{-6}$ m. The spatial size and resolution of the Salinas image are $512 \times 217$ and 3.7 m/pixel, respectively. A total of 0.8% pixels of each land-cover category are randomly picked as training samples and the rest of them are used for testing. The detailed quantities are listed in Table III. The false-color composite of the Salinas image and the corresponding ground truth are shown in Fig. 10.

The University of Houston image is gathered by the ITRES-CASI on the University of Houston campus and the neighboring urban areas in 2012. It contains 144 spectral bands and 15 land-cover categories. The wavelengths of its spectral bands

TABLE II
NUMBERS OF TRAINING AND TESTING SAMPLES OF THE UNIVERSITY OF PAVIA DATASET (2%, 98%)

| Class | Name | Training | Testing |
|---|---|---|---|
| 1 | Asphalt | 132 | 6499 |
| 2 | Meadows | 372 | 18277 |
| 3 | Gravel | 41 | 2058 |
| 4 | Trees | 61 | 3003 |
| 5 | Painted-Metal-Sheets | 26 | 1319 |
| 6 | Bare-Soil | 100 | 4929 |
| 7 | Bitumen | 26 | 1304 |
| 8 | Self-Blocking-Bricks | 73 | 3609 |
| 9 | Shadows | 18 | 929 |
| | Total | 849 | 41927 |

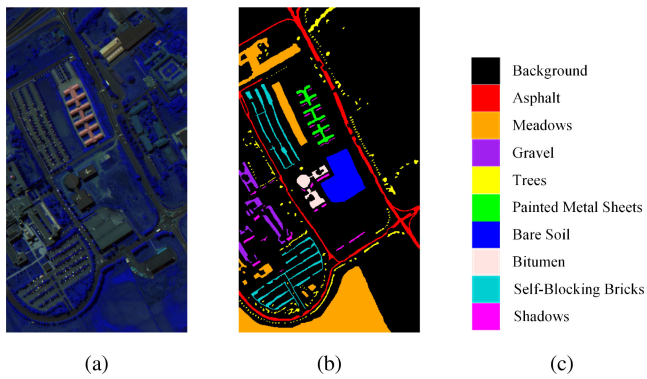



Fig. 9. University of Pavia image. (a) False color composite. (b) Ground truth. (c) Color code.

TABLE III
NUMBERS OF TRAINING AND TESTING SAMPLES OF THE SALINAS DATASET (0.8%, 99.2%)

| Class | Name | Training | Testing |
|---|---|---|---|
| 1 | Brocoli_green_weeds_1 | 16 | 1993 |
| 2 | Brocoli_green_weeds_2 | 29 | 3697 |
| 3 | Fallow | 15 | 1961 |
| 4 | Fallow_rough_plow | 11 | 1383 |
| 5 | Fallow_smooth | 21 | 2657 |
| 6 | Stubble | 31 | 3928 |
| 7 | Celery | 28 | 3551 |
| 8 | Grapes_untrained | 90 | 11181 |
| 9 | Soil_vinyard_develop | 49 | 6154 |
| 10 | Corn_senesced_green_weeds | 26 | 3252 |
| 11 | Lettuce_romaine_4wk | 8 | 1060 |
| 12 | Lettuce_romaine_5wk | 15 | 1912 |
| 13 | Lettuce_romaine_6wk | 7 | 909 |
| 14 | Lettuce_romaine_7wk | 8 | 1062 |
| 15 | Vinyard_untrained | 58 | 7210 |
| 16 | Vinyard_vertical_trellis | 14 | 1793 |
| | Total | 426 | 53703 |

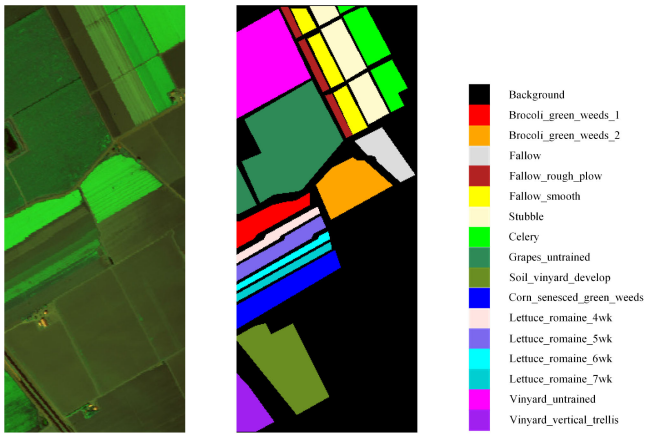



Fig. 10. Salinas image. (a) False color composite. (b) Ground truth. (c) Color code.

TABLE IV
NUMBERS OF TRAINING AND TESTING SAMPLES OF THE UNIVERSITY OF HOUSTON DATASET (5%, 95%)

| Class | Name | Training | Testing |
|---|---|---|---|
| 1 | Healthy_grass | 62 | 1189 |
| 2 | Stressed_grass | 62 | 1192 |
| 3 | Synthetic_grass | 34 | 663 |
| 4 | Trees | 62 | 1182 |
| 5 | Soil | 62 | 1180 |
| 6 | Water | 16 | 309 |
| 7 | Residential | 63 | 1205 |
| 8 | Commercial | 62 | 1182 |
| 9 | Road | 62 | 1190 |
| 10 | Highway | 61 | 1166 |
| 11 | Railway | 61 | 1174 |
| 12 | Parking_Lot_1 | 61 | 1172 |
| 13 | Parking_Lot_2 | 23 | 446 |
| 14 | Tennis_Court | 21 | 407 |
| 15 | Running_Track | 33 | 627 |
| | Total | 745 | 14284 |

range from 0.36 $\times$ $10^{-6}$ to 1.05 $\times$ $10^{-6}$ m. The spatial size and resolution of the University of Houston image are 349 $\times$ 1905 and 2.5 m/pixel, respectively. A total of 5% pixels of each land-cover category are randomly selected as training samples and the rest of them are used for testing. The detailed quantities are listed in Table IV. The false-color composite of the University of Houston image and the corresponding ground truth are shown in Fig. 11.

The percentages of training pixels of the four datasets vary widely because each HSI have different spatial context and spectral signatures. Experiment results show that 10%, 2%, 0.8%, and 5% are approaching the lower bounds on the premise of a good performance on the four datasets, respectively. In order to mitigate the problem of limited training HSI cubes, all HSI cubes in datasets are horizontally and vertically flipped and rotated 90°, 180°, and 270°. Then, the number of HSI cubes is increased by 400%.

### B. Evaluation Measures

We use the overall accuracy (OA), average accuracy (AA), kappa coefficient (Kappa), and the accuracy of each category to quantify the classification performance of the CRCN model. OA is employed to measure the overall classification accuracy, which is the ratio of the number of correctly classified HSI cubes to the number of all HSI cubes. AA represents the average classification accuracy in all categories. And Kappa reflects the degree of agreement between classification results and ground truth.

Assuming that $M \in R^{N\times N}$ denotes the error matrix of classification results, where $N$ is the number of categories. And $M_{ij}$ denotes the value of $M$ in index $(i, j)$. It means that there are $M_{ij}$ HSI cubes in the $i$th category, which are classified to the $j$th category. Then, the formula of OA, AA, and Kappa is defined as follows:

$$\mathrm{OA} = \frac{\sum_{i=1}^{N} M_{ii}}{\sum_{i=1}^{N}\sum_{j=1}^{N} M_{ij}} \tag{22}$$

$$\mathrm{AA} = \frac{1}{N}\sum_{i=1}^{N}\frac{M_{ii}}{\sum_{j=1}^{N} M_{ij}} \tag{23}$$

$$P = \frac{\sum_{k=1}^{N}\sum_{i=1}^{N}\sum_{j=1}^{N} M_{ik}M_{kj}}{(\sum_{i=1}^{N}\sum_{j=1}^{N} M_{ij})^2} \tag{24}$$

$$\mathrm{Kappa} = \frac{\mathrm{OA} - P}{1 - P}. \tag{25}$$

In the data process stage, we normalized the values of all HSI cubes to the range from 0 to 1, which is expressed as

$$\hat{X}(x, y, z) = \frac{X(x, y, z) - \min(X)}{\max(X) - \min(X)} \tag{26}$$

where $X$ is the original HSI cube and $X(x, y, z)$ denotes the value of $X$ in position $(x, y, z)$. $\hat{X}(x, y, z)$ represents the normalized $X(x, y, z)$. The $\max(X)$ and $\min(X)$ are the maximum and minimum of $X$, respectively.

### C. Experimental Settings

The performance of the CRCN is compared with that of other models, which include 3-D CNN [5], SSRN [22], DFFN [25], SMBN [26], CapsNets [41], and SSAN [28]. The detailed parameter information of each layer in the CRCN on the Indian Pines, University of Pavia, and Salinas datasets are listed in Table IV. Besides, the residual module and capsule module are also evaluated on the four datasets. For a fair comparison, all models in our experiments use the same experimental settings, including data processing, data augmentation, and parameter settings. In experiments on the residual module, the capsule module is replaced with several 3-D convolution layers whose convolution kernel's spectral size is set to 1. And the classifier of the residual module is a fully connected layer activated by the Softmax$(\cdot)$ function, we denote the residual module and its classifier as RM+classifier. And in experiments on the capsule module, shortcut connections in all residual blocks are not retained and the convolution kernel's spatial size in all residual

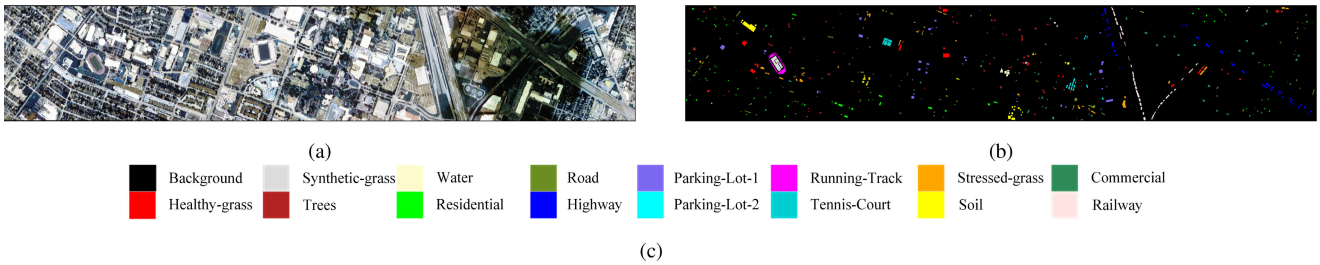



Fig. 11. University of Houston image. (a) False color composite. (b) Ground truth. (c) Color code.

blocks is set to $1 \times 1$. And the classifier of the capsule module is the same with the CRCN, we denote the capsule module and its classifier as CM+classifier.

All models are trained for 300 epochs and the batch size is set to 18. The spatial size of input HSI cubes is set to $11 \times 11$. The threshold value $\alpha^+$ is set to 0.9. The iteration times $r$ of dynamic routing is set to 2. Weight hyperparameters $\lambda$ and $\theta$ are set to 0.5 and 1, respectively. The initial learning rate and weight decay of each epoch are both set to $10^{-4}$. We optimize all models via the Adam optimizer with default parameters on TensorFlow. All experiments are conducted with Intel Core i7-9700K, 32GB RAM, GeForce GTX Titan X, TensorFlow 1.15.0, cuda 10.0, cudnn 7.6.0, and python 3.7.9. The detailed network architecture of the proposed CRCN on the Indian Pines, University of Pavia, and Salinas images is listed in Table V.

### *D. Classification Results and Analysis*

As can be seen in Table I, the distribution of the number of training samples for all categories in the Indian Pines image is quite uneven. In the "Grass-pasture-mowed" and the "Oats" categories, there are merely 2 pixels for training. In the "Alfalfa" and the "Stone-Steel-Towers" categories, there are just 4 and 9 training pixels, respectively. The classification results of all compared models on the Indian Pines dataset are shown in Table VI. The CRCN achieves an accuracy of 100.00% in the categories of "Alfalfa," "Grass-pasture-mowed," "Oats," and "Stone-Steel-Towers." SSAN obtains an accuracy of 100.00% in both "Grass-pasture-mowed" and "Oats" categories but only gets an accuracy of 85.71% in the "Alfalfa" category. 3-D CNN achieves an accuracy of 96.15% in the "Grass-pasture-mowed" category and an accuracy of 100.00% in the "Stone-Steel-Towers" category. But it just gains an accuracy of 90.48% in the "Alfalfa" category and an accuracy of 88.89% in the "Oats" category. DFFN obtains the same performance with 3-D CNN in the "Grass-pasture-mowed," "Oats," and "Stone-Steel-Towers" categories but merely gets an accuracy of 88.10% in the "Alfalfa" category. The accuracies achieved by SMBN in the "Grass-pasture-mowed," "Oats," and "Stone-Steel-Towers" categories are 96.15%, 100.00%, and 98.18%, respectively. While it just gets an accuracy of 88.10% in the "Alfalfa" category. Similarly, SSRN and CapsNets only obtain the accuracies of 88.10% and 85.71% in the "Alfalfa" category but both achieve an accuracy of 100.00% in the "Stone-Steel-Towers" category. From the classification results, we can observe that the CRCN achieves a superior performance in the "Alfalfa," "Grass-pasture-mowed," "Oats," and "Stone-Steel-Towers" categories, which all have limited training samples. The capsule module with its classifier outperforms other compared methods in OA, AA, and Kappa. Besides, the CRCN improves the OA, AA, and Kappa of the capsule module with its classifier by 0.28%, 1.22%, and 0.32%, respectively, which indicates the effectiveness of the CRCN.

According to Table II, the distribution of the number of training samples for all categories in the University of Pavia image is also uneven. In the "Shadows" category, there are only 18 training pixels. In the "Painted-Metal-Sheets" and "Bitumen" categories, there are just 26 pixels for training. As shown in Table VII, the CRCN provides competitive results to these compared state-of-the-art methods. The CRCN achieves the maximum accuracy of 99.92% while 3-D CNN obtains the minimum accuracy of 90.41% in the "Bitumen" category. The accuracies obtained by DFFN, SMBN, SSAN, SSRN, and CapsNets in the "Bitumen" category are 93.79%, 98.85%, 99.54%, 94.94%, and 96.70%, respectively. SMBN achieves the maximum accuracy of 100.00% while SSAN gets the minimum accuracy of 95.16% in the "Shadows" category. The CRCN achieves an accuracy of 99.89% in the "Shadows" category. The accuracies achieved by 3-D CNN, DFFN, SSRN, and CapsNets in the "Shadows" category are 99.14%, 99.46%, 99.25%, and 99.78%, respectively. All models achieve the accuracy of 100.00% in the "Painted-Metal-Sheets" category. Both the residual module with its classifier and the capsule module with its classifier outperform other compared methods in OA, AA, and Kappa, which illustrates the validity of the two modules. In addition, the CRCN improves the OA, AA, and Kappa of the capsule module with its classifier by 0.14%, 0.35%, and 0.23%, respectively, which indicates the effectiveness of the capsule module and the residual module.

From Tables III and IV, we can see that the distribution of the number of training samples for all categories in the Salinas and University of Houston images is slightly more balanced than that in the other two images. There are 90 training pixels in the "Grapes_untrained" category, while in "Lettuce_romaine_6wk" category, there are merely 7 pixels for training. In the "Lettuce_romaine_4wk" and "Lettuce_romaine_7wk" categories, there are only 8 pixels for training. From the classification results reported in Table VIII, we can see that both the CRCN and SSAN obtain an accuracy of 100.00% in the "Lettuce_romaine_6wk" category. The accuracies obtained by SMBN, SSAN, SSRN in the "Lettuce_romaine_4wk" category are 99.15%, 98.58%,

TABLE V
NETWORK ARCHITECTURE OF THE CRCN ON THE INDIAN PINES, UNIVERSITY OF PAVIA, AND SALINAS IMAGES

| HSI | Tensor size | Layer | | Number | Kernel number | stride | Batch normalization | Activation function |
|---|---|---|---|---|---|---|---|---|
| Indian Pines | $11 \times 11 \times 200$ | input | | 1 | – | – | – | – |
| | $11 \times 11 \times 100$ | 3-D convolution | | 2 | 16 | $(1, 1, 2)$ | Yes | ReLU |
| | $11 \times 11 \times 50$ | 3-D MaxPooling | | 3 | – | $(1, 1, 2)$ | – | – |
| | $11 \times 11 \times 25$ | residual module | residual units 1 | 4-9 | 64 | $(1, 1, 2)$ | Yes | ReLU |
| | $11 \times 11 \times 13$ | | residual units 2 | 10-15 | 128 | $(1, 1, 2)$ | Yes | ReLU |
| | $11 \times 11 \times 7$ | | residual units 3 | 16-18 | 256 | $(1, 1, 2)$ | Yes | ReLU |
| | $11 \times 11 \times 4$ | | residual units 4 | 19-21 | 512 | $(1, 1, 2)$ | Yes | ReLU |
| | $7 \times 7 \times 4$ | capsule module | 3-D convolution | 22 | 256 | $(1, 1, 1)$ | Yes | ReLU |
| | $4 \times 4 \times 4$ | | 3-D convolution | 23 | 128 | $(1, 1, 1)$ | Yes | ReLU |
| | $128 \times 64$ | | spatial-spectral capsule | 24 | – | – | – | Squashing function |
| | $16 \times 16$ | | class capsule | 25 | – | – | – | Squashing function |
| | 8192 | decoder | Fully-connected | 26 | – | – | – | – |
| | $4 \times 4 \times 64$ | | 3-D deconvolution | 27-29 | 32 | $(1, 1, 1)$ | Yes | ReLU |
| | $7 \times 7 \times 96$ | | 3-D deconvolution | 30 | 16 | $(1, 1, 1)$ | Yes | ReLU |
| | $11 \times 11 \times 200$ | | 3-D deconvolution | 31 | 1 | $(1, 1, 2)$ | Yes | Sigmoid |
| University of Pavia | $11 \times 11 \times 103$ | input | | 1 | – | – | – | – |
| | $11 \times 11 \times 52$ | 3-D convolution | | 2 | 16 | $(1, 1, 2)$ | Yes | ReLU |
| | $11 \times 11 \times 26$ | 3-D MaxPooling | | 3 | – | $(1, 1, 2)$ | – | – |
| | $11 \times 11 \times 13$ | residual module | units 1 | 4-9 | 64 | $(1, 1, 2)$ | Yes | ReLU |
| | $11 \times 11 \times 7$ | | residual units 2 | 10-15 | 128 | $(1, 1, 2)$ | Yes | ReLU |
| | $11 \times 11 \times 4$ | | residual units 3 | 16-21 | 256 | $(1, 1, 2)$ | Yes | ReLU |
| | $11 \times 11 \times 2$ | | residual units 4 | 22-27 | 512 | $(1, 1, 2)$ | Yes | ReLU |
| | $7 \times 7 \times 2$ | capsule module | 3-D convolution | 28 | 256 | $(1, 1, 1)$ | Yes | ReLU |
| | $4 \times 4 \times 2$ | | 3-D convolution | 29 | 128 | $(1, 1, 1)$ | Yes | ReLU |
| | $128 \times 32$ | | spatial-spectral capsule | 30 | – | – | – | Squashing function |
| | $9 \times 16$ | | class capsule | 31 | – | – | – | Squashing function |
| | 4096 | decoder | Fully-connected | 32 | – | – | – | – |
| | $4 \times 4 \times 41$ | | 3-D deconvolution | 33-35 | 32 | $(1, 1, 1)$ | Yes | ReLU |
| | $7 \times 7 \times 50$ | | 3-D deconvolution | 36 | 16 | $(1, 1, 1)$ | Yes | ReLU |
| | $11 \times 11 \times 103$ | | 3-D deconvolution | 37 | 1 | $(1, 1, 2)$ | Yes | Sigmoid |
| Salinas | $11 \times 11 \times 204$ | input | | 1 | – | – | – | – |
| | $11 \times 11 \times 102$ | 3-D convolution | | 2 | 16 | $(1, 1, 2)$ | Yes | ReLU |
| | $11 \times 11 \times 51$ | 3-D MaxPooling | | 3 | – | $(1, 1, 2)$ | – | – |
| | $11 \times 11 \times 26$ | residual module | residual units 1 | 4-9 | 64 | $(1, 1, 2)$ | Yes | ReLU |
| | $11 \times 11 \times 13$ | | residual units 2 | 10-12 | 128 | $(1, 1, 2)$ | Yes | ReLU |
| | $11 \times 11 \times 7$ | | residual units 3 | 13-15 | 256 | $(1, 1, 2)$ | Yes | ReLU |
| | $11 \times 11 \times 4$ | | residual units 4 | 16-18 | 512 | $(1, 1, 2)$ | Yes | ReLU |
| | $7 \times 7 \times 4$ | capsule module | 3-D convolution | 19 | 256 | $(1, 1, 1)$ | Yes | ReLU |
| | $4 \times 4 \times 4$ | | 3-D convolution | 20 | 128 | $(1, 1, 1)$ | Yes | ReLU |
| | $128 \times 64$ | | spatial-spectral capsule | 21 | – | – | – | Squashing function |
| | $16 \times 16$ | | class capsule | 22 | – | – | – | Squashing function |
| | 8192 | decoder | Fully-connected | 23 | – | – | – | – |
| | $4 \times 4 \times 64$ | | 3-D deconvolution | 24-26 | 32 | $(1, 1, 1)$ | Yes | ReLU |
| | $7 \times 7 \times 96$ | | 3-D deconvolution | 27 | 16 | $(1, 1, 1)$ | Yes | ReLU |
| | $11 \times 11 \times 204$ | | 3-D deconvolution | 28 | 1 | $(1, 1, 2)$ | Yes | Sigmoid |

TABLE VI
CLASSIFICATION ACCURACIES (IN PERCENTAGES) ON THE INDIAN PINES IMAGE OBTAINED BY 3-D CNN [5], DFFN [25], SMBN [26], SSAN [28], SSRN [22], CAPSNETS [41], RM + CLASSIFIER, CM + CLASSIFIER, AND CRCN WITH 10% TRAINING SAMPLES PER CLASS

| Class | 3-D CNN [5] | DFFN [25] | SMBN [26] | SSAN [28] | SSRN [22] | CapsNets [41] | RM + classifier | CM + classifier | CRCN |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 90.48 | 88.10 | 88.10 | 85.71 | 88.10 | 85.71 | 69.05 | 95.24 | **100.00** |
| 2 | 95.88 | 97.82 | 98.37 | 98.83 | 96.50 | 97.82 | 99.07 | 98.99 | **99.22** |
| 3 | 95.85 | 97.46 | 99.46 | 99.60 | 94.51 | 98.66 | 99.33 | 99.33 | **99.73** |
| 4 | 98.13 | 96.26 | **100.00** | 97.20 | 97.66 | 98.60 | **100.00** | **100.00** | **100.00** |
| 5 | 98.62 | 99.54 | 98.85 | 97.01 | 96.78 | 99.77 | 99.31 | 97.93 | **100.00** |
| 6 | 98.48 | 99.85 | 99.85 | 99.39 | 99.85 | **100.00** | **100.00** | **100.00** | **100.00** |
| 7 | 96.15 | 96.15 | 96.15 | **100.00** | 92.31 | 84.62 | **100.00** | 96.15 | **100.00** |
| 8 | 99.77 | 99.77 | **100.00** | **100.00** | 99.54 | **100.00** | **100.00** | **100.00** | **100.00** |
| 9 | 88.89 | 88.89 | **100.00** | **100.00** | 94.44 | **100.00** | **100.00** | 94.44 | **100.00** |
| 10 | 91.77 | 98.86 | 98.74 | **99.54** | 96.34 | 96.69 | 99.09 | 99.31 | 99.43 |
| 11 | 97.65 | 99.14 | 98.82 | 99.41 | 99.32 | 99.55 | 99.37 | 99.46 | **99.77** |
| 12 | 91.76 | 95.32 | 97.19 | 97.57 | 90.26 | 95.51 | 98.13 | 98.50 | **98.69** |
| 13 | **100.00** | 99.46 | **100.00** | **100.00** | **100.00** | 99.46 | **100.00** | **100.00** | **100.00** |
| 14 | 98.86 | 98.77 | 99.12 | 98.42 | 96.22 | 98.60 | 99.12 | **99.65** | 99.30 |
| 15 | 98.28 | **100.00** | 99.14 | 99.43 | 97.70 | **100.00** | 99.71 | **100.00** | **100.00** |
| 16 | **100.00** | **100.00** | 98.81 | 97.62 | **100.00** | **100.00** | **100.00** | 97.62 | **100.00** |
| OA(%) | 96.76 | 98.53 | 98.88 | 98.93 | 97.12 | 98.57 | 99.18 | 99.32 | **99.60** |
| AA(%) | 96.29 | 97.21 | 98.29 | 98.11 | 96.22 | 97.19 | 97.64 | 98.54 | **99.76** |
| Kappa | 96.31 | 98.32 | 98.73 | 98.78 | 96.71 | 98.37 | 99.06 | 99.22 | **99.54** |

TABLE VII
CLASSIFICATION ACCURACIES (IN PERCENTAGES) ON THE UNIVERSITY OF PAVIA IMAGE OBTAINED BY 3-D CNN [5], DFFN [25], SMBN [26], SSAN [28], SSRN [22], CAPSNETS [41], RM + CLASSIFIER, CM + CLASSIFIER, AND CRCN WITH 2% TRAINING SAMPLES PER CLASS

| Class | 3-D CNN [5] | DFFN [25] | SMBN [26] | SSAN [28] | SSRN [22] | CapsNets [41] | RM + classifier | CM + classifier | CRCN |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 98.32 | 99.92 | 99.37 | **100.00** | 99.05 | 99.46 | 99.88 | 99.95 | **100.00** |
| 2 | 99.90 | 99.96 | 99.93 | 99.97 | 99.94 | 99.99 | **100.00** | 99.97 | **100.00** |
| 3 | 93.83 | 95.09 | 96.99 | 97.52 | 92.27 | 95.77 | 98.49 | 98.93 | **99.32** |
| 4 | 96.40 | 97.50 | 98.10 | 95.70 | 96.90 | 96.77 | 97.50 | 97.67 | **98.53** |
| 5 | **100.00** | **100.00** | **100.00** | **100.00** | **100.00** | **100.00** | **100.00** | **100.00** | **100.00** |
| 6 | 99.92 | **100.00** | **100.00** | 99.86 | 99.78 | **100.00** | **100.00** | **100.00** | **100.00** |
| 7 | 90.41 | 93.79 | 98.85 | 99.54 | 94.94 | 96.70 | 99.00 | 99.46 | **99.92** |
| 8 | 93.68 | 99.03 | 98.95 | 99.58 | 97.34 | 99.31 | 99.72 | 99.09 | **99.86** |
| 9 | 99.14 | 99.46 | **100.00** | 95.16 | 99.25 | 99.78 | 99.78 | 99.89 | 99.89 |
| OA(%) | 98.26 | 99.26 | 99.46 | 99.39 | 98.80 | 99.31 | 99.67 | 99.66 | **99.84** |
| AA(%) | 96.85 | 98.31 | 99.13 | 98.59 | 97.72 | 98.64 | 99.38 | 99.44 | **99.73** |
| Kappa | 97.70 | 99.02 | 99.29 | 99.18 | 98.40 | 99.08 | 99.56 | 99.55 | **99.79** |

TABLE VIII
CLASSIFICATION ACCURACIES (IN PERCENTAGES) ON THE SALINAS IMAGE OBTAINED BY 3-D CNN [5], DFFN [25], SMBN [26], SSAN [28], SSRN [22], CAPSNETS [41], RM + CLASSIFIER, CM + CLASSIFIER, AND CRCN WITH 0.8% TRAINING SAMPLES PER CLASS

| Class | 3-D CNN [5] | DFFN [25] | SMBN [26] | SSAN [28] | SSRN [22] | CapsNets [41] | RM + classifier | CM + classifier | CRCN |
|---|---|---|---|---|---|---|---|---|---|
| 1 | **100.00** | **100.00** | **100.00** | **100.00** | 99.85 | 98.39 | **100.00** | **100.00** | **100.00** |
| 2 | 99.43 | 99.24 | 99.97 | 99.86 | 99.62 | 99.95 | 99.92 | **100.00** | **100.00** |
| 3 | 92.35 | 99.13 | 99.39 | **100.00** | 98.47 | 98.62 | 91.48 | 99.18 | 99.08 |
| 4 | 97.69 | 98.34 | 98.26 | 97.32 | 98.77 | 99.28 | 99.49 | 99.49 | **99.64** |
| 5 | 98.65 | 98.31 | 97.67 | 98.65 | 97.29 | 97.89 | 97.78 | **100.00** | 99.92 |
| 6 | 99.57 | 99.36 | 98.27 | 99.95 | 99.24 | 99.13 | 99.97 | 99.95 | **100.00** |
| 7 | 97.32 | 99.63 | 99.58 | 99.75 | 98.39 | 99.32 | **100.00** | **100.00** | **100.00** |
| 8 | 88.30 | 91.91 | 89.67 | **97.25** | 87.64 | 90.27 | 96.82 | 96.49 | 96.69 |
| 9 | 97.47 | 99.25 | 99.45 | 99.97 | 96.99 | 98.68 | 99.69 | **100.00** | **100.00** |
| 10 | 87.73 | 85.70 | 91.39 | 93.97 | 85.79 | 88.35 | 92.90 | 94.62 | **95.88** |
| 11 | 82.08 | 95.85 | **99.15** | 98.58 | 97.92 | 90.28 | 97.26 | 99.06 | 96.60 |
| 12 | 96.55 | 98.64 | 98.43 | 99.63 | 98.38 | **100.00** | **100.00** | **100.00** | **100.00** |
| 13 | 92.74 | 97.14 | 97.80 | **100.00** | 97.14 | 98.57 | 99.67 | **100.00** | **100.00** |
| 14 | 92.47 | 92.18 | 92.75 | **98.87** | 92.66 | 91.15 | 96.89 | 96.70 | 96.42 |
| 15 | 79.65 | 89.53 | 88.54 | **95.27** | 76.31 | 84.60 | 90.12 | 91.80 | 94.48 |
| 16 | 94.37 | 99.00 | **99.61** | 92.69 | 96.43 | 98.05 | 99.22 | 99.22 | 98.61 |
| OA(%) | 92.21 | 95.31 | 95.12 | 97.95 | 92.22 | 94.25 | 96.96 | 97.69 | **98.09** |
| AA(%) | 93.52 | 96.45 | 96.87 | 98.24 | 95.06 | 95.78 | 97.58 | 98.53 | **98.58** |
| Kappa | 91.32 | 94.78 | 94.57 | 97.72 | 91.32 | 93.60 | 96.61 | 97.42 | **97.87** |

and 97.92%, respectively. And SSAN achieves an accuracy of 98.87% in the "Lettuce_romaine_7wk" category. Although the CRCN achieves an accuracy of 98.87% in the "Lettuce_romaine_7wk" category, it outperforms SSAN in OA, AA, and Kappa by 0.14%, 0.34%, and 0.15%, respectively. In addition, the CRCN improves the OA, AA, and Kappa of the capsule module with its classifier by 0.40%, 0.05%, and 0.45%, respectively. From the classification results reported in Table IX, we can see that the CRCN achieves a better performance than other compared methods in OA, AA and Kappa. The bold entries in Tables VI–IX are the best classification accuracies of all compared methods on each category of the four datasets.

The classification maps of all compared models and the corresponding ground truths of the Indian Pines, University of Pavia and Houston, and Salinas images are shown in Figs. 12–15. As can be observed in Fig. 12, more intact edges of objects are preserved in the classification map of the CRCN than that of other compared approaches on the Indian Pines image. And smaller spatially adjacent land-cover areas are misclassified in the classification map of the residual module with its classifier than that of other compared methods on the Indian Pines image, which illustrates that the residual module with its classifier is effective to distinguish spectral similarity between HSI cubes of spatially adjacent categories. In Fig. 13, we can see that the classification map of all compared models are very nice on the University of Pavia image. But for several categories in which objects are scattered, the visual results of the CRCN are less noisy than that of other compared methods. In addition, misclassified borders between spatially adjacent categories in the classification map of the CRCN are less than that of other compared models in Figs. 14 and 15. In order to reflect the

TABLE IX
CLASSIFICATION ACCURACIES (IN PERCENTAGES) ON THE UNIVERSITY OF HOUSTON IMAGE OBTAINED BY 3-D CNN [5], DFFN [25], SMBN [26], SSAN [28], SSRN [22], CAPSNETS [41], RM + CLASSIFIER, CM + CLASSIFIER, AND CRCN WITH 5% TRAINING SAMPLES PER CLASS

| Class | 3-D CNN [5] | DFFN [25] | SMBN [26] | SSAN [28] | SSRN [22] | CapsNets [41] | RM + classifier | CM + classifier | CRCN |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 97.48 | 97.06 | **99.24** | 97.90 | 97.65 | 97.81 | 96.72 | 97.81 | 97.14 |
| 2 | 98.41 | 99.33 | 97.65 | 98.15 | 98.83 | 99.08 | 98.83 | **99.50** | 98.07 |
| 3 | **100.00** | 99.55 | **100.00** | **100.00** | 99.55 | **100.00** | **100.00** | **100.00** | **100.00** |
| 4 | **100.00** | 99.92 | 97.21 | 99.83 | 98.82 | 99.66 | **100.00** | **100.00** | **100.00** |
| 5 | 98.90 | 99.58 | **100.00** | 99.66 | 99.58 | 99.92 | 98.73 | 99.92 | 98.56 |
| 6 | 98.06 | 98.71 | **99.35** | 98.06 | 98.71 | 98.38 | 98.06 | 98.71 | 98.06 |
| 7 | 93.44 | 97.26 | 89.96 | 94.03 | 95.35 | **98.26** | 96.10 | 97.01 | 94.27 |
| 8 | 88.92 | 90.78 | 87.73 | 90.86 | 91.54 | 91.46 | 89.51 | 92.05 | 89.93 |
| 9 | 89.24 | 86.72 | 87.40 | 91.26 | 89.24 | 91.51 | 90.25 | 92.27 | 90.67 |
| 10 | 88.17 | 98.46 | 93.74 | **99.57** | 92.97 | 97.08 | 98.97 | 97.00 | 98.20 |
| 11 | 91.31 | 93.27 | 88.50 | 94.80 | 89.61 | 89.95 | 94.12 | 90.63 | **97.19** |
| 12 | 93.00 | 94.54 | 91.30 | 92.66 | 92.75 | 90.70 | 94.11 | 90.70 | **94.71** |
| 13 | 98.66 | 90.81 | 97.98 | 95.52 | 97.53 | 98.66 | **100.00** | 96.86 | **100.00** |
| 14 | 99.02 | **100.00** | 99.75 | 97.54 | 99.26 | **100.00** | **100.00** | **100.00** | **100.00** |
| 15 | 99.20 | 99.68 | 99.20 | **100.00** | **100.00** | **100.00** | **100.00** | **100.00** | **100.00** |
| OA(%) | 94.80 | 96.08 | 94.31 | 96.33 | 95.41 | 96.24 | 96.42 | 96.31 | **96.54** |
| AA(%) | 95.59 | 96.38 | 95.27 | 96.66 | 96.09 | 96.83 | 97.03 | 96.83 | **97.12** |
| Kappa | 94.38 | 95.76 | 93.85 | 96.03 | 95.04 | 95.94 | 96.13 | 96.01 | **96.26** |

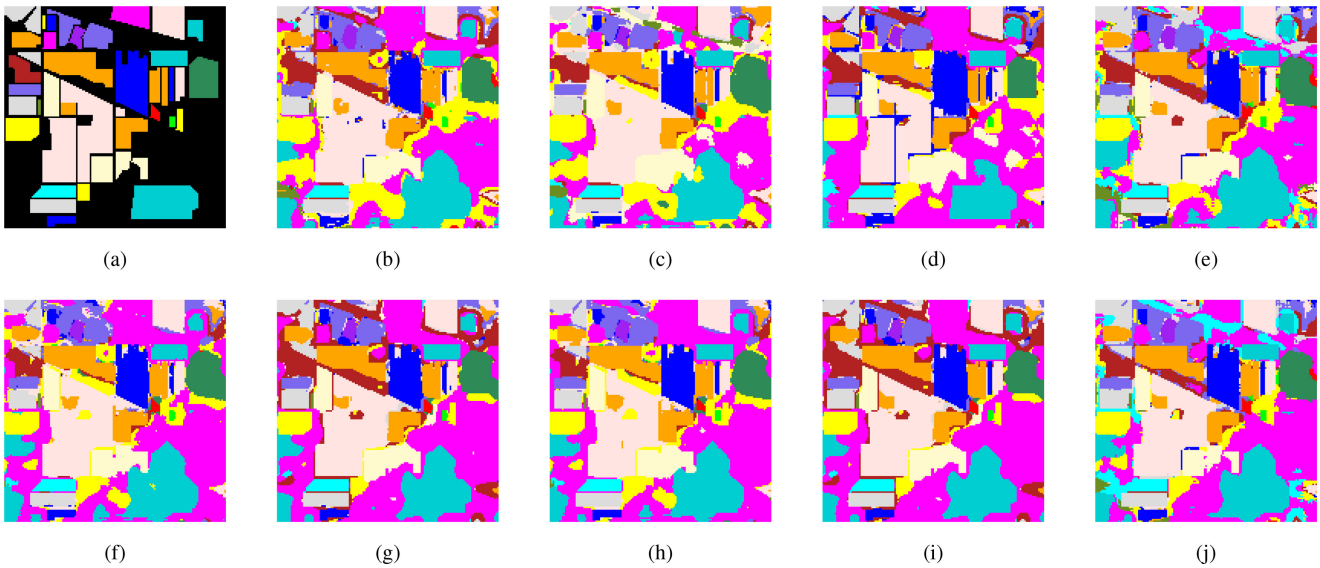



Fig. 12. Classification maps on the Indian Pines image obtained by (a) Groundtruth, (b) 3-D CNN, (c) DFFN, (d) SMBN, (e) SSAN, (f) SSRN, (g) CapsNets, (h) RM + classifier, (i) CM + classifier, and (j) CRCN.

computational efficiency of HSI classification models in our experiments. In Table X, we record the training and testing time of the CRCN and compared models.

### E. Effect of Depth of Residual Module

The depth of the residual module is a significant factor on the accuracy of HSI classification. Therefore, we conduct experiments to explore the performances of the CRCN with different depths of the residual module on the Indian Pines, University of Pavia and Houston, and Salinas datasets. The training ratios of each category on the four datasets are set to 10%, 2%, 5%, and 0.8%, respectively. The spatial size of input HSI cube is set to $11 \times 11$ and the depth of the residual module is set from 12 to 36. The OAs of the CRCN with different depths of the residual module on the four datasets are shown in Fig. 16. Generally, the OA is improved when increasing the depth of the residual module. But excessive depth also leads to a slight decline in accuracy because with the increasing of the network capacity, more training samples are required to achieve a good performance [57].

### F. Effect of Ratio of Training Samples

The ratio of training samples is also an important factor on the accuracy of HSI classification. In this part, we conduct

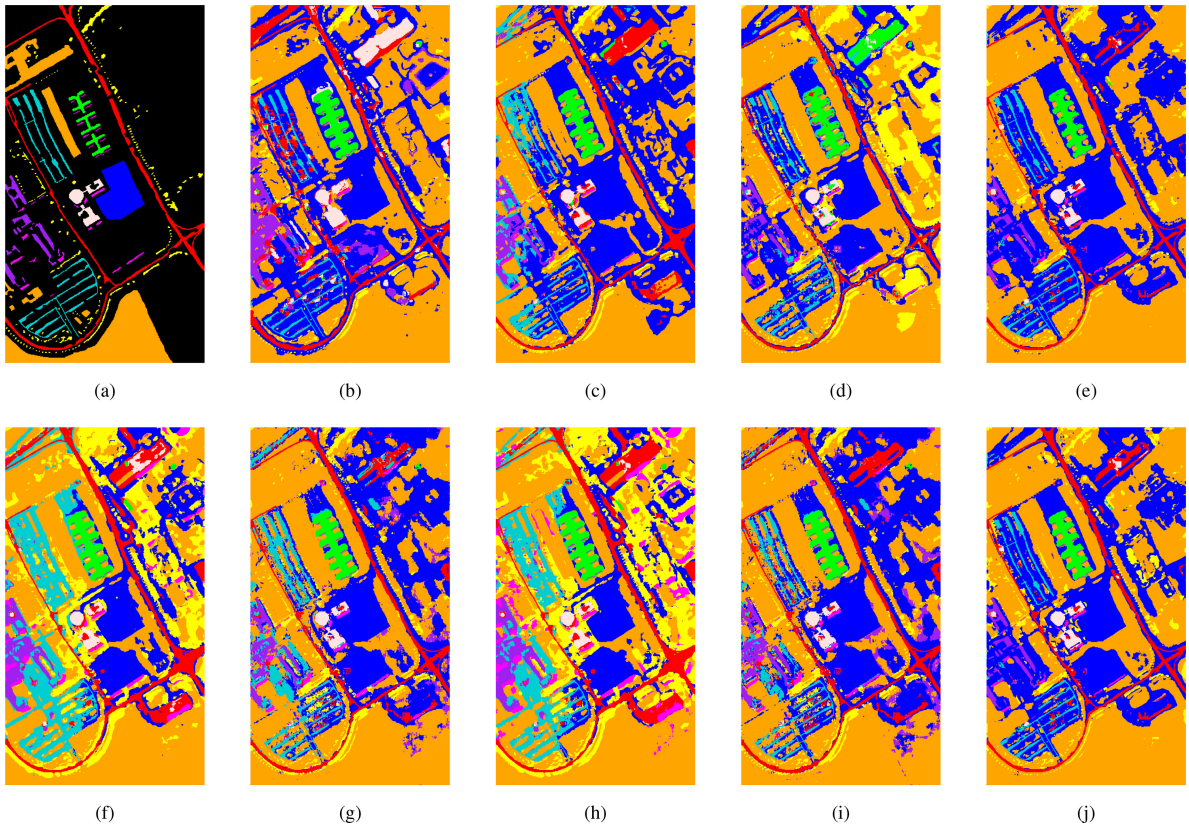



Fig. 13. Classification maps on the University of Pavia image obtained by (a) Groundtruth, (b) 3-D CNN, (c) DFFN, (d) SMBN, (e) SSAN, (f) SSRN, (g) CapsNets, (h) RM + classifier, (i) CM + classifier, and (j) CRCN.

TABLE X
TRAINING AND TESTING TIME (SECONDS) OF THE CRCN AND COMPARED MODELS ON EACH DATASET

| Data sets | Methods | 3-D CNN [5] | DFFN [25] | SMBN [26] | SSAN [28] | SSRN [22] | CapsNets [41] | RM + classifier | CM + classifier | CRCN |
|---|---|---|---|---|---|---|---|---|---|---|
| Indian Pines | Training time | 260.09 | 680.27 | 511.68 | 1821.71 | 1421.47 | 939.68 | 1485.16 | 1002.42 | 1608.26 |
| | Testing time | 1.18 | 1.56 | 1.69 | 3.87 | 3.36 | 3.16 | 3.22 | 3.41 | 3.45 |
| University of Pavia | Training time | 186.69 | 447.62 | 375.13 | 1491.59 | 1031.75 | 727.31 | 1128.94 | 783.63 | 1226.03 |
| | Testing time | 2.05 | 4.68 | 3.53 | 12.83 | 8.27 | 7.82 | 8.09 | 8.19 | 10.73 |
| Salinas | Training time | 339.02 | 678.58 | 552.37 | 2379.21 | 2550.83 | 986.92 | 2328.26 | 959.56 | 2346.08 |
| | Testing time | 3.81 | 6.45 | 5.34 | 22.34 | 20.17 | 9.67 | 18.95 | 9.55 | 20.27 |
| University of Houston | Training time | 188.23 | 450.31 | 376.22 | 1495.75 | 1038.15 | 733.53 | 1135.63 | 788.39 | 1228.86 |
| | Testing time | 1.53 | 2.05 | 2.27 | 5.31 | 4.86 | 4.18 | 4.27 | 4.52 | 4.66 |

experiments to explore the performances of all compared models with different ratios of training samples on the Indian Pines, University of Pavia and Houston, and Salinas datasets. First, 5%, 10%, 15%, and 20% labeled pixels in each category of Indian Pines image are randomly selected for training. And for the University of Pavia image, 0.5%, 1%, 1.5%, and 2% labeled pixels in each category are randomly selected as training samples. And 5%, 10%, 13%, and 18% labeled pixels in each category are randomly selected as training samples in the University of Houston image. At last, 0.8%, 1%, 2%, and 5% labeled pixels in each category of Salinas image are randomly selected for training. The performances of all compared models with different ratios of training samples on the four datasets are shown in Tables XI – XIV. To achieve an excellent classification performance, the CRCN requires only 1.5% training samples in the University of Pavia image but needs about 18%, 5%, and 15% training samples in the University of Houston, Salinas, and Indian Pines images, respectively.

### G. *Effect of Spatial Size*

The spatial size of input HSI cubes is also an important factor on the accuracy of HSI classification. In this part, we conduct experiments to explore the performances of all compared methods with different spatial sizes of input HSI cubes on the four datasets. The ratios of training samples of the Indian Pines, University of Pavia and Houston, and Salinas datasets are set to 10%, 2%, 5%, and 0.8%, respectively. The spatial sizes of

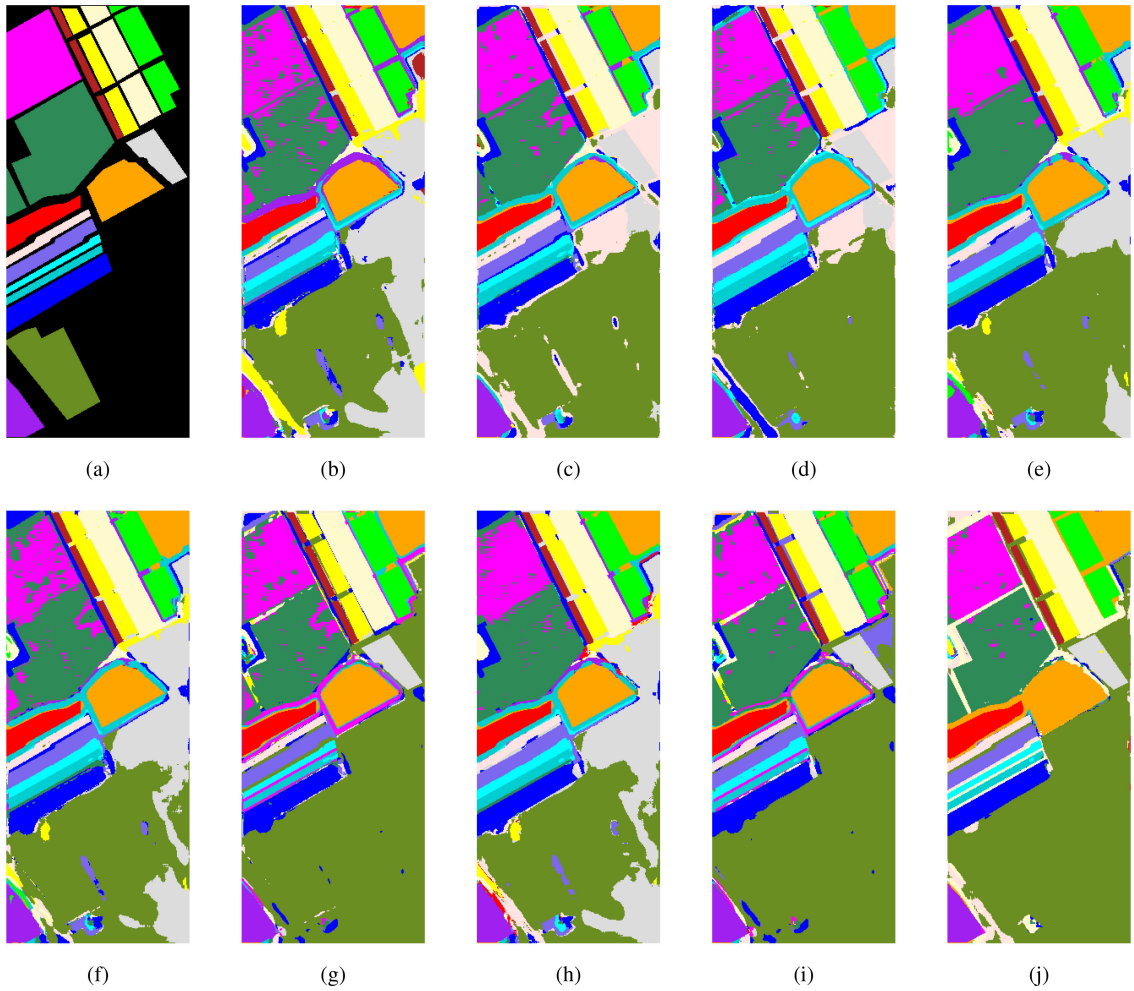



Fig. 14. Classification maps on the Salinas image obtained by (a) Groundtruth, (b) 3-D CNN, (c) DFFN, (d) SMBN, (e) SSAN, (f) SSRN, (g) CapsNets, (h) RM + classifier, (i) CM + classifier, and (j) CRCN.

TABLE XI
OA (IN PERCENTAGES) ON THE INDIAN PINES IMAGE OBTAINED BY 3-D CNN [5], DFFN [25], SMBN [26], SSAN [28], SSRN [22], CAPSNETS [41], RM + CLASSIFIER, CM + CLASSIFIER, AND CRCN WITH DIFFERENT TRAINING RATIOS

| Training Ratio | 3-D CNN [5] | DFFN [25] | SMBN [26] | SSAN [28] | SSRN [22] | CapsNets [41] | RM + classifier | CM + classifier | CRCN |
|---|---|---|---|---|---|---|---|---|---|
| 5% | 95.88 | 97.46 | 97.19 | 97.62 | 96.22 | 97.82 | 98.13 | 98.25 | 98.50 |
| 10% | 96.76 | 98.53 | 98.88 | 98.93 | 97.12 | 98.57 | 99.18 | 99.32 | 99.60 |
| 15% | 97.65 | 98.77 | 99.12 | 99.39 | 97.66 | 98.66 | 99.37 | 99.46 | 99.81 |
| 20% | 98.86 | 99.54 | 99.46 | 99.60 | 99.32 | 99.55 | 99.46 | 99.65 | 99.84 |

TABLE XII
OA (IN PERCENTAGES) ON THE UNIVERSITY OF PAVIA IMAGE OBTAINED BY 3-D CNN [5], DFFN [25], SMBN [26], SSAN [28], SSRN [22], CAPSNETS [41], RM + CLASSIFIER, CM + CLASSIFIER, AND CRCN WITH DIFFERENT TRAINING RATIOS

| Training Ratio | 3-D CNN [5] | DFFN [25] | SMBN [26] | SSAN [28] | SSRN [22] | CapsNets [41] | RM + classifier | CM + classifier | CRCN |
|---|---|---|---|---|---|---|---|---|---|
| 0.5% | 96.92 | 97.50 | 97.95 | 97.97 | 97.34 | 97.99 | 98.13 | 98.09 | 98.57 |
| 1% | 97.83 | 98.79 | 98.85 | 98.70 | 98.52 | 98.78 | 99.00 | 99.16 | 99.43 |
| 1.5% | 98.14 | 99.03 | 99.37 | 99.16 | 98.78 | 99.29 | 99.56 | 99.44 | 99.82 |
| 2% | 98.26 | 99.26 | 99.46 | 99.39 | 98.80 | 99.31 | 99.67 | 99.66 | 99.84 |

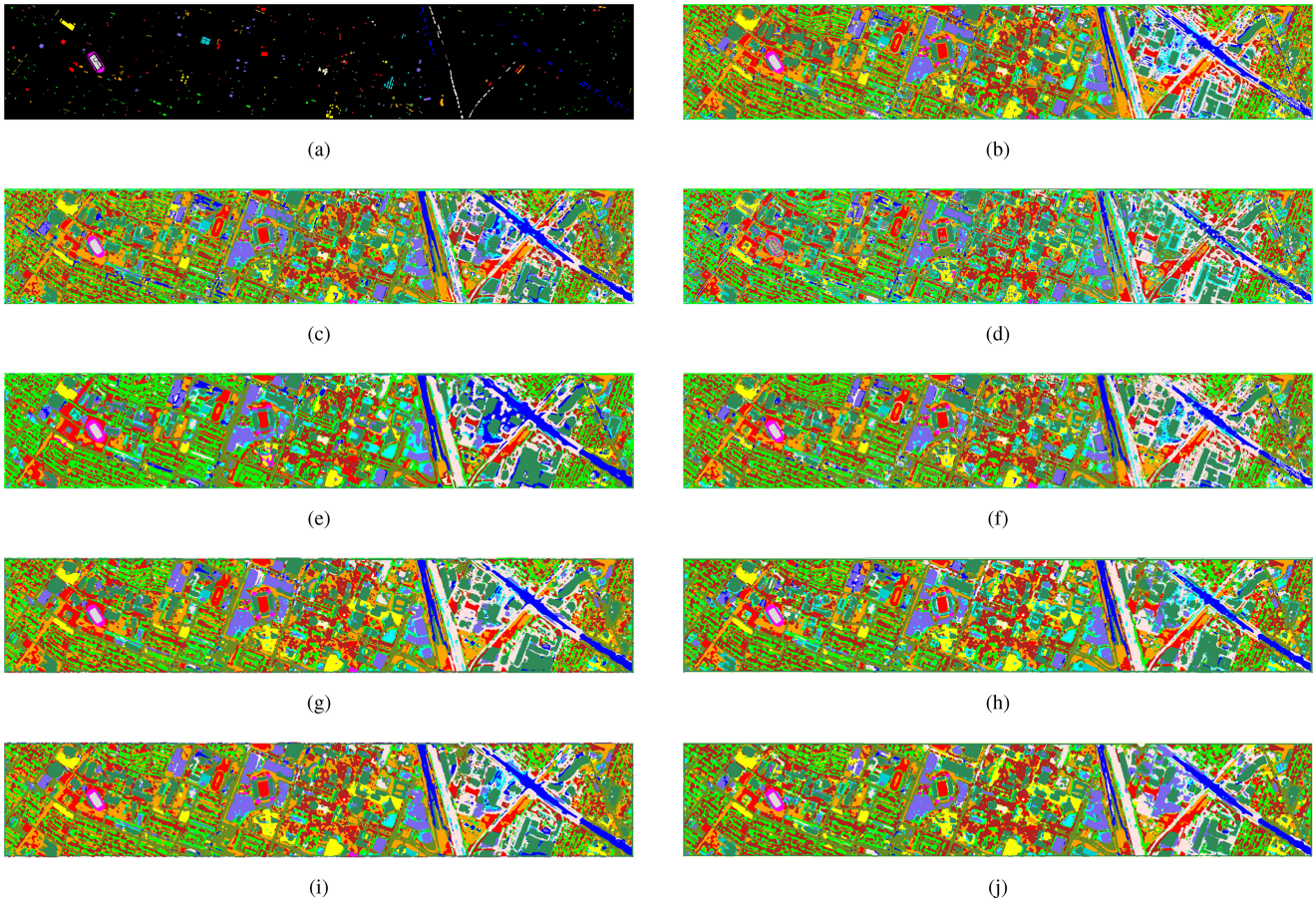



Fig. 15. Classification maps on the University of Houston image obtained by (a) Groundtruth, (b) 3-D CNN, (c) DFFN, (d) SMBN, (e) SSAN, (f) SSRN, (g) CapsNets, (h) RM + classifier, (i) CM + classifier, and (j) CRCN.

TABLE XIII
OA (IN PERCENTAGES) ON THE SALINAS IMAGE OBTAINED BY 3-D CNN [5], DFFN [25], SMBN [26], SSAN [28], SSRN [22], CAPSNETS [41], RM + CLASSIFIER, CM + CLASSIFIER, AND CRCN WITH DIFFERENT TRAINING RATIOS

| Training Ratio | 3-D CNN [5] | DFFN [25] | SMBN [26] | SSAN [28] | SSRN [22] | CapsNets [41] | RM + classifier | CM + classifier | CRCN |
|---|---|---|---|---|---|---|---|---|---|
| 0.8% | 92.21 | 95.31 | 95.12 | 97.95 | 92.22 | 94.25 | 96.96 | 97.69 | 98.09 |
| 1% | 93.52 | 96.14 | 95.87 | 98.58 | 93.64 | 95.31 | 97.58 | 98.42 | 98.67 |
| 2% | 94.69 | 96.63 | 96.39 | 98.86 | 94.92 | 95.78 | 98.33 | 98.85 | 99.08 |
| 5% | 95.73 | 97.91 | 97.24 | 99.53 | 95.85 | 96.57 | 99.21 | 99.49 | 99.79 |

TABLE XIV
OA (IN PERCENTAGES) ON THE UNIVERSITY OF HOUSTON IMAGE OBTAINED BY 3-D CNN [5], DFFN [25], SMBN [26], SSAN [28], SSRN [22], CAPSNETS [41], RM + CLASSIFIER, CM + CLASSIFIER, AND CRCN WITH DIFFERENT TRAINING RATIOS

| Training Ratio | 3-D CNN [5] | DFFN [25] | SMBN [26] | SSAN [28] | SSRN [22] | CapsNets [41] | RM + classifier | CM + classifier | CRCN |
|---|---|---|---|---|---|---|---|---|---|
| 5% | 94.80 | 96.08 | 94.31 | 96.33 | 95.41 | 96.24 | 96.42 | 96.31 | 96.54 |
| 10% | 97.56 | 97.87 | 97.68 | 98.69 | 97.98 | 98.53 | 98.95 | 99.08 | 99.34 |
| 13% | 98.21 | 98.42 | 98.15 | 98.86 | 98.29 | 98.76 | 99.18 | 99.39 | 99.51 |
| 18% | 98.58 | 98.66 | 98.62 | 98.95 | 98.43 | 98.97 | 99.43 | 99.56 | 99.78 |

input HSI cubes are set from $3 \times 3$ to $13 \times 13$. The OAs of all compared methods with different spatial sizes of input HSI cubes on the four datasets are shown in Figs. 17–20. The OAs on the four datasets gradually increases as the spatial size increases because large HSI cube contains more spatial context information. However, when the spatial size is increased to $13 \times 13$, the OAs of all methods begin to decline since too large spatial size contains more background and hyperspectral pixels of other categories and the number of HSI cubes of each category for training is not increased.

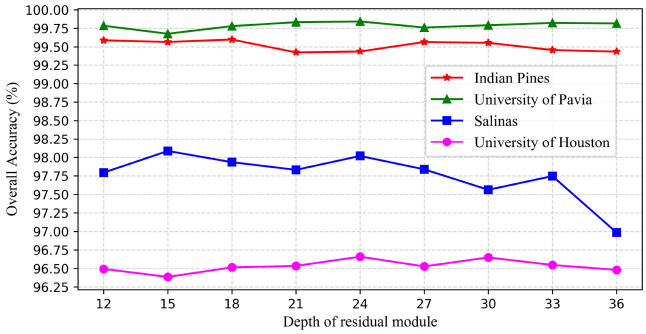



Fig. 16. Effect of depth on classification accuracy of the CRCN in the Indian Pines, University of Pavia, Salinas, and University of Houston images.

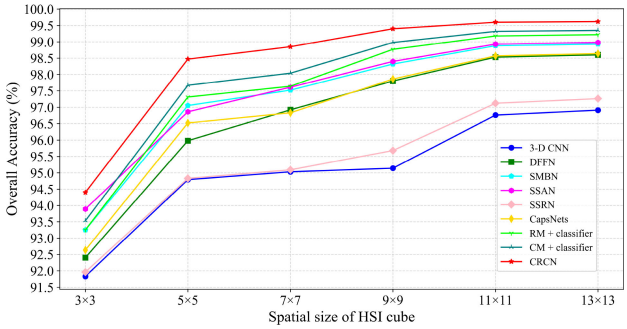



Fig. 17. Effect of spatial size on classification accuracy of all compared methods in the Indian Pines image.

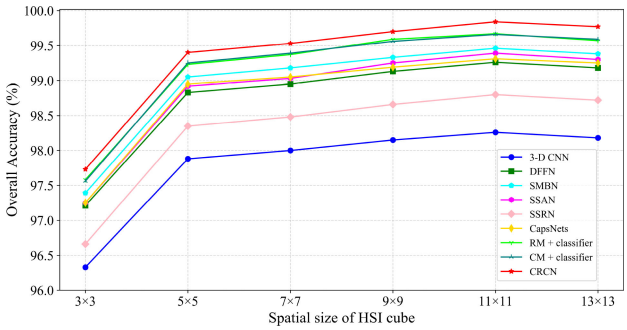



Fig. 18. Effect of spatial size on classification accuracy of all compared methods in the University of Pavia image.

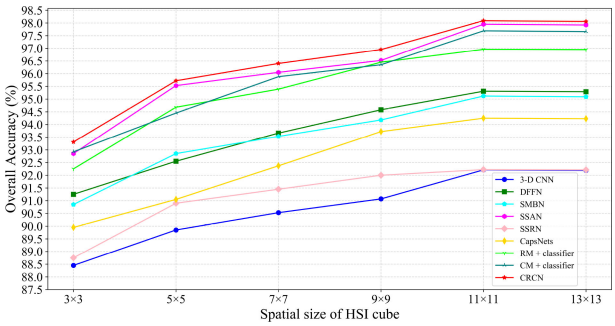



Fig. 19. Effect of spatial size on classification accuracy of all compared methods in the Salinas image.

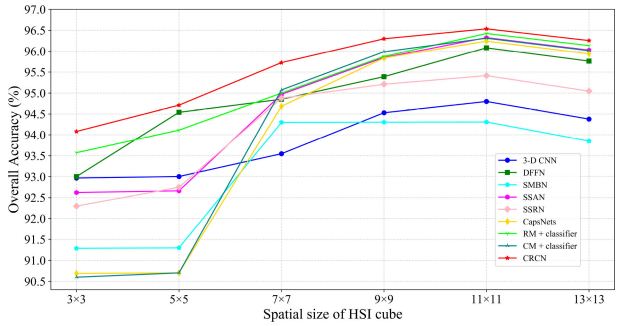



Fig. 20. Effect of spatial size on classification accuracy of all compared methods in the University of Houston image.

TABLE XV
OA (IN PERCENTAGES) ON THE INDIAN PINES (IP), UNIVERSITY OF PAVIA (UP), SALINAS (SA), AND UNIVERSITY OF HOUSTON (UH) IMAGES OBTAINED BY CRCN WITH DIFFERENT HYPERPARAMETERS IN LOSS FUNCTION

| $\alpha^+$ | $\lambda$ | $\theta$ | OA of IP | OA of UP | OA of SA | OA of UH |
|---|---|---|---|---|---|---|
| 0.8 | 0.5 | 1 | 99.48 | 99.73 | 97.96 | 96.43 |
| 0.95 | 0.5 | 1 | 99.56 | 99.81 | 98.05 | 96.51 |
| 0.9 | 0.5 | 0.2 | 99.52 | 99.77 | 98.03 | 96.47 |
| 0.9 | 0.5 | 5 | 99.45 | 99.71 | 97.95 | 96.39 |
| 0.9 | 0.1 | 1 | 99.43 | 99.68 | 97.92 | 96.34 |
| 0.9 | 0.5 | 1 | 99.60 | 99.84 | 98.09 | 96.54 |
| 0.9 | 2.5 | 1 | 99.37 | 99.63 | 97.89 | 96.32 |

### H. Determining the Values of Hyperparameters in Loss Function

The hyperparameters $\alpha^+$, $\lambda$, and $\theta$ in loss function $L$ can significantly affect the performance of the proposed method. To determine the value of the three hyperparameters for the better classification performance, we conduct a set of experiments for each of them, in which only the value of one hyperparameter is changed, and the other two hyperparameters remain unchanged. The final value of each hyperparameter is chosen via the results of the corresponding set of experiments. Except for the three hyperparameters, the values of other hyperparameters are the same as in the experimental settings. The value of $\alpha^+$ is set from 0.8, 0.9, and 0.95. The value of $\lambda$ is set from 0.1, 0.5, and 2.5. And the value of $\theta$ is set from 0.2, 1, and 5. We provide the OA of the proposed method with different values of the three hyperparameters on the four datasets in Table XV. And we denote the Indian Pines, University of Pavia, Salinas, and University of Houston images as IP, UP, SA, and UH, respectively.

## V. CONCLUSION

In this article, we propose a new deep learning-based model for HSI classification. The proposed CRCN is designed to deal with the spectral similarity between HSI cubes of spatially adjacent categories and the diverse spatial context information on objects of the same category. The residual module of the CRCN is designed to learn high-level spectral features only in the spectral dimension, which can effectively alleviate the influence of spectral similarity between HSI cubes of spatially adjacent

categories. And then the capsule module of the CRCN is designed to learn spatial context orientation representations, which can effectively deal with the diverse spatial context information on objects of the same category in HSI cubes. Experimental results on the Indian Pines, University of Pavia and Houston, and Salinas datasets demonstrate that the CRCN outperforms six compared state-of-the-art methods and achieves more robust HSI classification performance with limited training samples. Although the CRCN is able to suppress the effects of spectral similarity between HSI cubes of spatially adjacent categories, the architecture of the CRCN is rather complicated. In future works, we will seek simple and effective methods to address the issue of spectral similarity between HSI cubes of spatially adjacent categories and diverse spatial context information on objects of the same category.

## REFERENCES


[1] Q. Wang, J. Z. Lin, and Y. Yuan, "Salient band selection for hyperspectral image classification via manifold ranking," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 6, pp. 1279–1289, Jun. 2016.

[2] P. Zhong, Z. Q. Gong, S. T. Li, and C. B. Schonlieb, "Learning to diversify deep belief networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 6, pp. 3516–3530, Jun. 2017.

[3] X. T. Zheng, Y. Yuan, and X. Q. Lu, "Dimensionality reduction by spatial-spectral preservation in selected bands," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 9, pp. 5185–5197, Sep. 2017.

[4] N. Akhtar and A. Mian, "Nonparametric coupled Bayesian dictionary and classifier learning for hyperspectral classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 9, pp. 4038–4050, Sep. 2018.

[5] Y. S. Chen, H. L. Jiang, C. Y. Li, X. P. 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.

[6] F. L. Luo, B. Du, L. P. Zhang, L. F. Zhang, and D. C. Tao, "Feature learning using spatial-spectral hypergraph discriminant analysis for hyperspectral image," *IEEE Trans. Cybern.*, vol. 49, no. 7, pp. 2406–2419, Jul. 2019.

[7] T. Lu, S. T. Li, L. Y. Fang, L. Bruzzone, and J. A. Benediktsson, "Set-to-set distance-based spectral-spatial classification of hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 12, pp. 7122–7134, Dec. 2016.

[8] X. T. Zheng, Y. Yuan, and X. Q. Lu, "A deep scene representation for aerial scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 7, pp. 4799–4809, Jul. 2019.

[9] M. E. Paoletti, J. M. Haut, J. Plaza, and A. Plaza, "A new deep convolutional neural network for fast hyperspectral image classification," *ISPRS J. Photogramm. Remote Sens.*, vol. 145, pp. 120–147, 2018.

[10] X. X. Lu, B. Q. Wang, X. T. Zheng, and X. L. Li, "Exploring models and data for remote sensing image caption generation," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 4, pp. 2183–2195, Apr. 2018.

[11] L. C. 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.

[12] S. T. Li, W. W. Song, L. Y. Fang, Y. S. Chen, P. Ghamisi, 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.

[13] Y. S. Chen, Z. H. Lin, X. Zhao, G. Wang, and Y. F. Gu, "Deep learning-based classification of hyperspectral data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2094–2107, Jun. 2014.

[14] C. H. Zhao, X. Q. Wan, G. P. Zhao, B. Cui, W. Liu, and B. Qi, "Spectral-spatial classification of hyperspectral imagery based on stacked sparse autoencoder and random forest," *Eur. J. Remote Sens.*, vol. 50, no. 1, pp. 47–63, 2017.

[15] T. Li, J. P. Zhang, and Y. Zhang, "Classification of hyperspectral image based on deep belief networks," in *Proc. IEEE Int. Conf. Image Proces.*, 2014, pp. 5132–5136.

[16] H. Wu and S. Prasad, "Semi-supervised deep learning using pseudo labels for hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1259–1270, Mar. 2018.

[17] J. Zhu, L. Y. 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.

[18] H. Lee and H. Kwon, "Going deeper with contextual CNN for hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 26, no. 10, pp. 4843–4855, Oct. 2017.

[19] S. H. Mei, J. Y. Ji, J. H. Hou, X. Li, and Q. Du, "Learning sensor-specific spatial-spectral features of hyperspectral images via convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 8, pp. 4520–4533, Aug. 2017.

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

[21] Z. Q. Gong, P. Zhong, Y. Yu, W. D. Hu, and S. T. 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.

[22] Z. L. Zhong, J. Li, Z. M. 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.

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

[24] W. Z. Zhao and S. H. 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.

[25] W. W. Song, S. T. Li, L. Y. 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.

[26] L. Y. Fang, G. Y. Liu, S. T. 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, Mar. 2019.

[27] H. K. Zhang, Y. Li, Y. Z. Zhang, and Q. Shen, "Spectral-spatial classification of hyperspectral imagery using a dual-channel convolutional neural network," *Remote Sens. Lett.*, vol. 8, no. 5, pp. 438–447, 2017.

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

[29] M. M. Zhang, W. Li, and Q. Du, "Diverse region-based CNN for hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 27, no. 6, pp. 2623–2634, Jun. 2018.

[30] N. J. He *et al.*, "Feature extraction with multiscale covariance maps for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 2, pp. 755–769, Feb. 2019.

[31] A. Santara *et al.*, "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.

[32] E. Aptoula, M. C. Ozdemir, and B. Yanikoglu, "Deep learning with attribute profiles for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 12, pp. 1970–1974, Dec. 2016.

[33] Y. S. Chen, L. Zhu, P. Ghamisi, X. P. Jia, G. Y. Li, and L. Tang, "Hyperspectral images classification with Gabor filtering and convolutional neural network," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 12, pp. 2355–2359, Dec. 2017.

[34] S. Y. Hao, W. Wang, Y. X. Ye, E. Y. Li, and L. Bruzzone, "A deep network architecture for super-resolution-aided hyperspectral image classification with classwise loss," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 8, pp. 4650–4663, Aug. 2018.

[35] X. R. Ma, A. Y. Fu, J. Wang, H. Y. Wang, and B. C. Yin, "Hyperspectral image classification based on deep deconvolution network with skip architecture," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 8, pp. 4781–4791, Aug. 2018.

[36] X. Y. Cao, F. Zhou, L. Xu, D. Y. Meng, Z. B. Xu, and J. Paisley, "Hyperspectral image classification with Markov random fields and a convolutional neural network," *IEEE Trans. Image Process.*, vol. 27, no. 5, pp. 2354–2367, May 2018.

[37] S. Y. Hao, W. Wang, Y. X. Ye, T. Y. Nie, and L. Bruzzone, "Two-stream deep architecture for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 4, pp. 2349–2361, Apr. 2018.

[38] Y. L. Duan, H. Huang, and Y. X. Tang, "Local constraint-based sparse manifold hypergraph learning for dimensionality reduction of hyperspectral image," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 1, pp. 613–628, Jan. 2021.

[39] F. L. Luo, L. P. Zhang, B. Du, and L. F. Zhang, "Dimensionality reduction with enhanced hybrid-graph discriminant learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 8, pp. 5336–5353, Aug. 2020.
[40] K. M. He, X. Y. Zhang, S. Q. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[41] M. E. Paoletti *et al.*, "Capsule networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 4, pp. 2145–2160, Apr. 2019.
[42] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Proc. Adv. Neural Inf. Process. Syst*., 2017, pp. 3859–3869.
[43] A. Sepas-Moghaddam, A. Etemad, F. Pereira, and P. L. Correia, "CapsField: Light field-based face and expression recognition in the wild using capsule routing," *IEEE Trans. Image Process.*, vol. 30, pp. 2627–2642, 2021.
[44] A. Stuhlsatz, J. Lippel, and T. Zielke, "Feature extraction with deep neural networks by a generalized discriminant analysis," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 4, pp. 596–608, Apr. 2012.
[45] L. Shao, D. Wu, and X. L. Li, "Learning deep and wide: A spectral method for learning deep networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 12, pp. 2303–2308, Dec. 2014.
[46] S. Dittmer, E. J. King, and P. Maass, "Singular values for ReLU layers," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 9, pp. 3594–3605, Sep. 2020.
[47] X. F. Liu, Q. Q. Sun, Y. Meng, M. Fu, and S. Bourennane, "Hyperspectral image classification based on parameter-optimized 3D-CNNs combined with transfer learning and virtual samples," *Remote Sens.*, vol. 10, no. 9, 2018, Art. no. 1425.
[48] H. K. Zhang, Y. Li, Y. A. Jiang, P. Wang, Q. Shen, and C. H. Shen, "Hyperspectral classification based on lightweight 3-D-CNN with transfer learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 8, pp. 5813–5828, Aug. 2019.
[49] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2017.
[50] M. M. Kalayeh and M. B. Shah, "Training faster by separating modes of variation in batch-normalized models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 6, pp. 1483–1500, Jun. 2020.
[51] T. Sun, L. B. Qiao, Q. Liao, and D. S. Li, "Novel convergence results of adaptive stochastic gradient descents," *IEEE Trans. Image Process.*, vol. 30, no. 4, pp. 1044–1056, Nov. 2021.
[52] J. Hertz, A. Krogh, B. Lautrup, and T. Lehmann, "Nonlinear backpropagation: Doing backpropagation without derivatives of the activation function," *IEEE Trans. Neural Netw.*, vol. 8, no. 6, pp. 1321–1327, Nov. 1997.
[53] X. H. Yu, M. O. Efe, and O. Kaynak, "A general backpropagation algorithm for feedforward neural networks learning," *IEEE Trans. Neural Netw.*, vol. 13, no. 1, pp. 251–254, Jan. 2002.
[54] S. L. Goh and D. P. Mandic, "Stochastic gradient-adaptive complex-valued nonlinear neural adaptive filters with a gradient-adaptive step size," *IEEE Trans. Neural Netw.*, vol. 18, no. 5, pp. 1511–1516, Sep. 2007.
[55] V. J. Mathews and Z. H. Xie, "A stochastic gradient adaptive filter with gradient adaptive step-size," *IEEE Trans. Signal Process.*, vol. 41, no. 6, pp. 2075–2087, Jun. 1993.
[56] S. Bonnabel, "Stochastic gradient descent on Riemannian manifolds," *IEEE Trans. Autom. Control*, vol. 58, no. 9, pp. 2217–2229, Sep. 2013.
[57] C. Y. Ji and D. Psaltis, "Capacity of two-layer feedforward neural networks with binary weights," *IEEE Trans. Inf. Theory*, vol. 44, no. 1, pp. 256–268, Jan. 1998.

**Zhiming Mei** is currently working toward the Ph.D. degree in remote sensing with the Innovation Academy for Microsatellites, Chinese Academy of Sciences, Shanghai, China, and the University of Chinese Academy of Sciences, Beijing, China.

His research interests include deep learning, computer vision, high-dimensional data process, and their applications in the context of hyperspectral image.

**Zengshan Yin** received the Ph.D. degree in automation from Zhejiang University, Hangzhou, China, in 2001.

He is currently a Researcher with the Innovation Academy for Microsatellites, Chinese Academy of Sciences, Shanghai, China, and the University of Chinese Academy of Sciences, Beijing, China. He has carried out the mission of the first global carbon dioxide monitoring satellite of China and the high resolution micro satellite of Chinese Academy of Sciences. He is currently the Chief Commander of the programs of the science lead ASOS satellite and the innovation No. 6 satellite of the Chinese Academy of Sciences. His research interests include small satellite system design, satellite remote sensing image process, digital signal process, and spatial information communication.

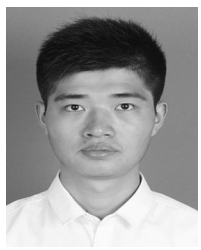
Dr. Yin was the recipient of the Second Prize of National Science and Technology Progress Award, the Outstanding Contribution Award of Chinese Academy of Sciences, the First Prize of Shanghai Science and Technology Progress Award, and the First Prize of Military Science and Technology Progress Award. He has been selected as a leading young and middle-aged scientific and technological innovation talent of the Ministry of Science and Technology.

**Xinwei Kong** received the M.E. degree in communication and information system from the Innovation Academy for Microsatellites, Chinese Academy of Sciences, Shanghai, China, in 2020, and the University of Chinese Academy of Sciences, Beijing, China.

Her research interests include deep learning, computer vision, and analysis of hyperspectral image.

**Long Wang** received the Ph.D. degree in communication engineering from the University of Chinese Academy of Sciences, Beijing, China, in 2019.

He is currently a Researcher with the Innovation Academy for Microsatellites, Chinese Academy of Sciences, Shanghai, China. His research interests include satellite networks, space optical communication, and space-based information network.



**Han Ren** is currently working toward the Ph.D. degree in remote sensing with the Innovation Academy for Microsatellites, Chinese Academy of Sciences, Shanghai, China, and the University of Chinese Academy of Sciences, Beijing, China.

His research interests include deep learning and digital signal processing.