Spectral–Spatial Exploration for Hyperspectral Image Classification via the Fusion of Fully Convolutional Networks

Liang Zou, *Member, IEEE*, Xingliang Zhu, Changfeng Wu, Yong Liu, *Member, IEEE*, and Lei Qu, *Member, IEEE*

***Abstract*—Due to its remarkable feature representation capability and high performance, convolutional neural networks (CNN) have emerged as a popular choice for hyperspectral image (HSI) analysis. However, the performances of traditional CNN-based patch-wise classification methods are limited by insufficient training samples, and the evaluation strategies tend to provide overoptimistic results due to training-test information leakage. To address these concerns, we propose a novel spectral–spatial 3-D fully convolutional network (SS3FCN) to jointly explore the spectral–spatial information and the semantic information. SS3FCN takes small patches of original HSI as inputs and produces the corresponding sized outputs, which enhances the utilization rate of the scarce labeled images and boosts the classification accuracy. In addition, to avoid the potential information leakage and make a fair comparison, we introduce a new principle to generate classification benchmarks. Experimental results on four popular benchmark datasets, including Salinas Valley, Pavia University, Indian Pines, and Houston University, demonstrate that the SS3FCN outperforms state-of-the-art methods and can be served as a baseline for future research on HSI classification.**

***Index Terms*—Hyperspectral image (HSI) classification, 3-D fully convolutional networks, spectral–spatial exploration.**

I. Introduction

With the rapid development of sensor technologies, it is feasible to capture hyperspectral images (HSI) containing hundreds of continuous spectral bands in a single acquisition [1], [2]. The abundant spectral information enables the successful applications of HSI in a broad range of areas, such as agriculture, military, environmental sciences, physics, and mineralogy [3], [4]. Most of these applications rely on robust and accurate classification of each pixel, i.e., HSI classification and segmentation [5]. However, it is costly and labor intensive to generate annotations for HSI due to the wide variety of sensors used. The current available HSI benchmark datasets only include far fewer labeled pixels in the whole image. Although various methods have been proposed in the last few decades, HSI classification is still a challenging problem due to limited annotated data and the complicate nature of these images [6].

A few traditional machine learning classifiers, such as random forest and support vector machines, have achieved great success in HSI classification [7], [8] by using only spectral information. However, the unbalance between the large dimension of spectral information and limited training samples impede the further improvement of classification performance [5]. It has been verified that the spatial correlation across HSIs can provide complementary information to spectral features, and should be taken into account [9]. An intuitive idea to boost the performance is to incorporate the spatial features and jointly explore the spectral–spatial information. Prior to the application of deep learning, there are two major categories of methods to exploit the spectral–spatial information: postprocessing and feature concatenation [9]. The first one assumes that the neighboring pixels with similar spectral characters are prone to be the same class and utilizes the spatial features in a postprocessing manner to further refine the spectral feature based classification. For instance, Markov random field and graph cut were employed to sharpen the classification boundaries in [10] and [11]. The second one extracts the spectral and spatial information separately and concatenates them before performing HSI classification [12]. Although these handcrafted features-based methods have made substantial improvements in understanding HSI, their poor generalization ability is generally observed due to the domain knowledge dependence nature of feature engineering [4], [13]. Those shallow handcrafted features have limited power in fully representing the abundant spectral and spatial information. In addition, considerable work has been done on improving the classification performance via feature transfer learning [14], [15].

Recently, deep learning based methods have shown promising results in HSI classification [5], [6], [9], [16]–[21]. In [18] and [19], the spectral–spatial information was exploited via stacked autoencoder (SAE) and deep belief network (DBN), respectively. However, both SAE and DBN only accept 1-D

Manuscript received August 16, 2019; revised November 27, 2019 and December 30, 2019; accepted January 16, 2020. Date of publication February 5, 2020; date of current version February 20, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61901003 and Grant 61871411, in part by the Natural Science Foundation of Jiangsu Province under Grant BK20190623, and in part by the University Synergy Innovation Program of Anhui Province under Grant GXXT-2019-008. *(Corresponding author: Lei Qu.)*

L. Zou is with the School of Information and Electrical Control Engineering, China University of Mining and Technology, Xuzhou 221116, China, and also with the School of Electronics and Information Engineering, Anhui University, Hefei 236601, China (e-mail: liangzou@ece.ubc.ca).

X. Zhu, C. Wu, and L. Qu are with the School of Electronics and Information Engineering, Anhui University, Hefei 236601, China (e-mail: leonzx7@163.com; ahwcf1995@163.com; qulei@ahu.edu.cn).

Y. Liu is with the Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly (LILY), Nanyang Technological University, Singapore 639798 (e-mail: stephenliu@ntu.edu.sg).

Digital Object Identifier 10.1109/JSTARS.2020.2968179

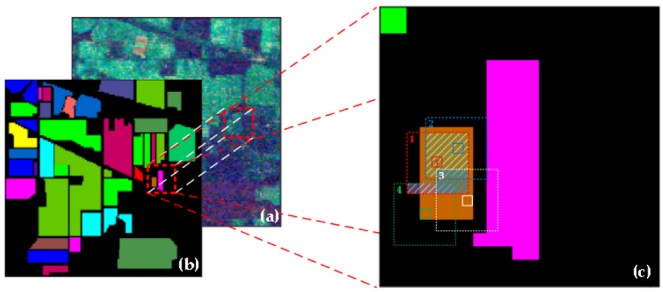


Fig. 1. Demonstration of traditional way to create training-test splits. (a) False color image. (b) Ground truth, different colors represent different classes. (c) Zoom-in of the selected area and demonstration of different splits for training/test set.

input data. The flatten layer has to be employed to collapse the spatial dimension into a vector whereas it may bring about spatial information loss. Considering the importance of spatial information in HSI classification and the remarkable achievements of convolutional neural networks (CNN) in the field of computer vision, CNN is becoming the most popular and effective way for HSI classification. In the past few years, many CNN-based studies have been reported and their impressive performances were witnessed [20], [22], [23]. For instance, Zhao *et al.* [23] applied 2-D CNN to extract spatial features from the first three principal components bands of the raw HSI. Then, the extracted spatial features were combined with the spectral features. Despite of the improved performance compared with previous studies, there are spectral–spatial information loss due to only three principal components are fed into CNN. In order to fully exploit the joint spectral–spatial information which is important for HSI classification, 3-D CNN were adopted to extract deep spectral–spatial features directly from 3-D cube of raw HSI [5], [9], [24]. It was demonstrated that 3-D CNN is able to provide robust and discriminative features. Similarly, some recent deep learning based methods for HSI classification can also be found in [25]–[29].

Although significant performance improvement have been achieved by deep learning based methods, there are still some issues need to be tackled.

First and foremost, the potential training-test information leakage renders overoptimistic performance, especially for CNN-based HSI classification methods [30]. Traditional HSI classification models aim to classify a given patch with the central pixel T, as shown in Fig. 1. The corresponding label of the patch is same as the label of T and the neighbor pixels are selected to assist the classification of T. However, in the previous routines of constructing test dataset, researchers always do not exclude all the neighborhoods of the pixels in the training set. Taking the scenario in Fig. 1 as an example, patch 1 is selected as the training sample, whereas the patches 2 and 3 which have overlaps with patch 1 may be selected as the test samples. The attempt utilizing partial samples in both training and test set leads to the training–testing information leakage. Although Nalepa *et al.* proposed a strategy to avoid the information leakage [30], there is still much room to improve the accuracy without information leakage.

Second, researchers in the remote sensing field used to employ patch-wise classification methods to predict the labels of each pixel, and ignore the difference between patch-wise classification and pixels-to-pixels classification [31], [32]. This may render the failure to fully explore the complicated nature of HSI. It should be note that there are significant differences between them. Patch-wise classification decides the label of the central pixel based on the information from the whole patch, and therefore only utilizes a mere portion of the limited labeled pixels in the case without training-test information leakage. In comparison, pixels-to-pixels classification aims to classify each pixel into a fixed set of categories, and this prediction strategy can fully utilize the limited annotated images.

At last, a more effective and robust way to extract the latent spectral–spatial features from limited labeled data is highly desired. For instance, a few researchers assume that the spectral information and the spatial information are of similar importance, and merely employ the 3-D CNN to joint explore the spectral–spatial information. However, for HSI with low spatial resolution, several objects may reside in one pixel. Therefore, the label information might be questionable and the spatial information is limited. In dealing with low spatial resolution HSIs, spectral information should be paid more attention than spatial information.

To address the above concerns, we propose a novel framework to classify each pixel via 3-D fully convolutional network. The main contributions of our spectral–spatial 3-D fully convolutional network (SS3FCN) lies in three folds.

1) We share our observations of the potential training-test information leakage in traditional patch-wise HSI classification and introduce a novel way to generate classification benchmarks without training-test information leakage, which make a fair comparison feasible.
2) We interpret the task of assigning label for each pixel as the pixels-to-pixels classification problem, rather than the traditional patch-wise classification. Our key insight is that the pixels-to-pixels prediction strategy can fully utilize the limited annotated images. To the best of the authors knowledge, we are the first few to employ FCN in HSI classification.
3) We exploit the spectral–spatial information by the fusion of 3-D and 1-D FCN, and demonstrate that the spectral information might be more powerful in analyzing HSIs with low spatial resolution. With simple structure, the proposed SS3FCN generalizes well on four popular HSI benchmarks and achieves state-of-the-art performance in term of both average accuracy (AA) and overall accuracy (OA).

II. Training-Test Splits and Data Enhancement

Considering the overoptimistic performance for the training-test information leakage and inspired by the patch-based split, in this article, we propose a novel data partition method and obtain a balanced training/test partition without information leakage. In addition, we employ data enhancement to enrich the limited training/test samples. Comparison experiments with state-of-the-art HSI classification methods are set up on four

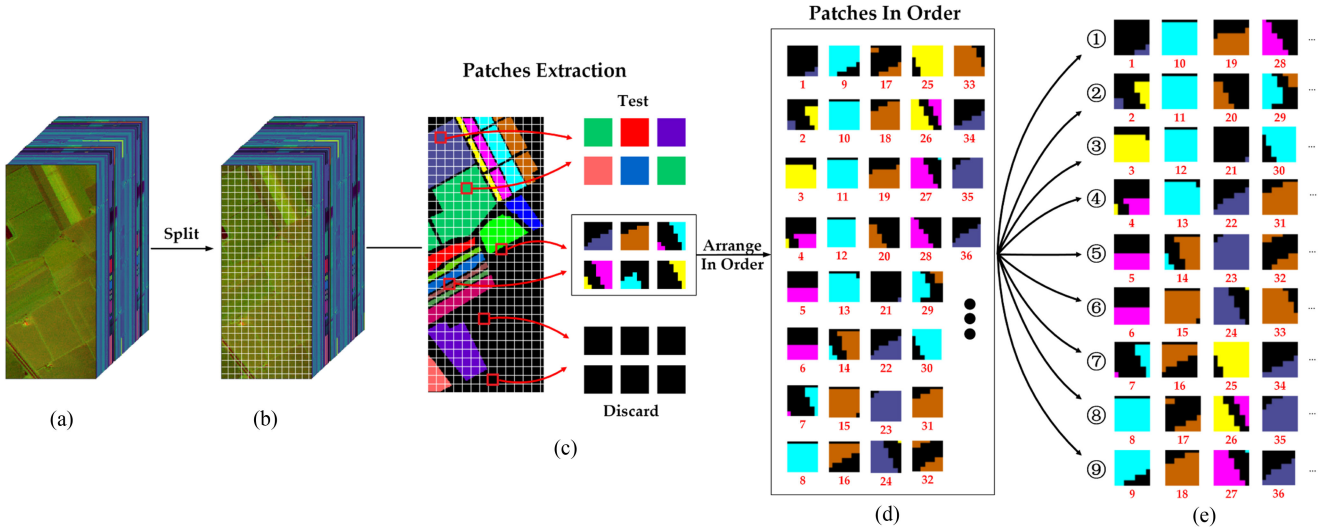


Fig. 2. Training-test blocks splits. (a) Input HSI. (b) Dividing the image into blocks. (c) Blocks extraction. (d) Arranging blocks in order. (e) Different folds of blocks from the input HSI.

benchmark hyperspectral datasets, including Salinas Valley, Pavia University, Indian Pines, and Houston University, to verify the effectiveness of the proposed method.

A. *Training-Test Splits*

Traditional ways via random splits do not exclude all neighborhoods of the training samples in constructing test set and therefore may lead to the training-test information leakage, as shown in Fig. 1. Recently, Nalepa *et al.* [30] developed a training-test partition method by patch-based algorithm from the input images. There is no overlap between the obtained training and test set. However, their patch-based method gives rise to the unbalanced pixels in training, validation, and test set. For instance, the C7 class of Pavia University is completely selected as the test set but is not present in the training set, and therefore the trained model cannot distinguish this class from the others. It directly affects the performance, especially for the class with a limited number of pixels.

In order to address the information leakage and unbalanced problem, we propose a novel data partition method to jointly exploit spectral–spatial information and achieve a more balanced training/test split. First, we divide the whole HSI into blocks with the size of $W \times H \times C$, where W and H are the width and height, C is the number of spectral bands, as illustrated in Fig. 2(b). The W and H are selected heuristically based on the tradeoff between the size and the number of blocks. In this article, we assume the width W is same to the height H. As shown in Fig. 2(c), we discard the blocks where all the pixels are unlabeled and select the blocks with only one kind of pixels as the test set, denoted as test set-1. The remaining blocks with more than one class of pixels are sorted column-wisely following their orders in the HSI, as demonstrated in Fig. 2(d). We further split these multiclass blocks into training set, validation set, and additional test set, denoted as test set-2. Different from the way in Monte Carlo cross validation, all the multiclass blocks are partitioned into K folds, where the subsequent two blocks in each fold is K apart in term of the order. The parameter K is limited by the percentage of pixels taken as training samples, which is smaller than 12% in this article. For instance, as shown in Fig. 2(e), we split the multiclass blocks in Salinas Valley into nine folds and the order of samples in the k-th fold is [k, $9 +$ k,..., $9N +$ k] where N represents the number of samples in this fold. We select a single fold as the training set, the other one as the validation set, and the remaining seven folds as the test set-2. The test set-1 and set-2 are combined into the test set. There is no overlap between training and test set avoiding the potential information leakage. We repeat this process nine times where each fold is used exactly once as the training data.

B. *Data Enhancement*

The combination of more samples and deep network always outperforms the combination of limited sample and shallow network [33]. The strategy which employs the sliding window through the entire HSI may cause the overlapping problem. In this article, we apply the sliding window on each block within training/test set separately. The window with the size of $D \times D$ ($D < W$) is utilized to slide in the spatial domain and thus can provide patches from each block, as shown in Fig. 3(a). In addition, we employ the traditional data augmentation methods, such as flip, random rotation, on each patch, as shown in Fig. 3(b). Although this method appears trivial and simple, in practically, it can significantly enhance the number of samples in the training set and ensure that there is no information leakage between training and test set.

In general, our data preparation method effectively avoids the potential training-test information leakage for the nonoverlap patches between training and test set. In addition, we split the data sequentially and thus alleviate the data imbalance problem between the training, validating, and testing set [30], [34], [35]. The data augmentation strategy enables us to train a deeper neural network with better performance.

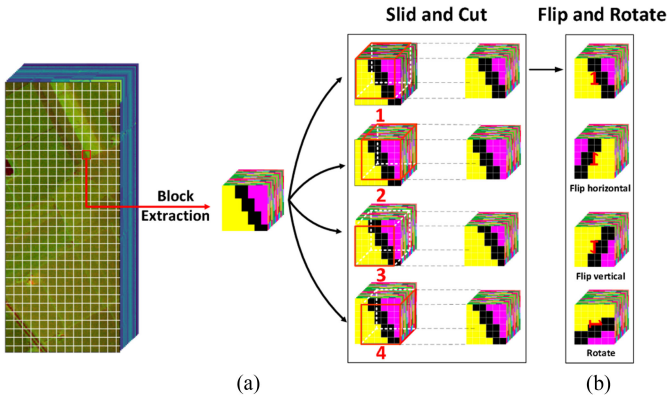


Fig. 3. Process of data enhancement via (a) sliding window to divide the blocks into patches. (b) Traditional data augmentation methods including flipping and rotating.

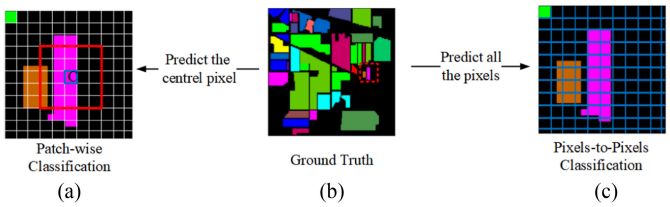


Fig. 4. Differences between patch-wise classification and pixels-to-pixels classification.

III. Proposed SS3FCN for Pixels-to-Pixels Classification

Many researchers in the field of hyperspectral imaging via deep learning, if not most, pay more attention to the patch-wise classification method and ignore the semantic segmentation method in computer vision. Although both aim to obtain label information from the input images, there is significant difference between them. In this article, we interpret the task of assigning label for each pixel as the pixels-to-pixels classification (which is similar in appearance to semantic segmentation) problem, rather than the traditional patch-wise classification.

A. Pixels-to-Pixels Classification Versus Patch-Wise Classification in HSI

Traditional HSI classification aims to assign a single class to the central pixel using the spectral–spatial information from the whole patch (i.e., patch-wise classification) whereas pixels-to-pixels classification aims to classify every pixel of the whole patch [32]. Since the pixels-to-pixels classification is able to provide the predictions of all pixels in a single trail, this task is always referred to as dense per-pixel prediction.

In traditional HSI patch-wise classification, the neighbors spectral–spatial information is employed to assist the prediction of the central pixel. Given a patch with a spatial size of $W \times H$, the classification model will output a single class for its central pixel, as shown in Fig. 4(a). However, if the patches used for training have overlap with the patches for test (i.e., the case in Fig. 1), it may lead to overoptimistic experimental insights. In case of the splits without overlap, the obtained patches may be too less to train the deep learning model in HSI analysis. Unlike the patch-wise classification, HSI pixels-to-pixels classification provides an efficient way to assign label to each pixel in the given HSI. It takes input of patch with the spatial size of $W \times H$ and produces the correspondingly sized output with the label information, as shown in Fig. 4(c). Compared with the traditional patch-wise classification which aim to assign a single label to the central pixel, it can utilize more annotated information without overlap, and provide more robust inference.

B. HSI Pixels-to-Pixels Classification via 3-D Fully Convolutional Networks

Convolutional neural networks have shown break-through performance on various vision-related tasks, such as object detection, image segmentation, and video classification, especially in the scenarios where local information are beneficial for the classification. As to the HSI analysis, the spatial features provide complementary information to the spectral features, and the joint extraction of spectral–spatial information can significantly improve the performance [9]. Due to the remarkable successes obtained by FCN in semantic segmentation of general images, in this work, we propose a novel architecture based on 3-D FCN to assign a label for each pixel in an HSI, shown in Fig. 5.

The basic unit of our network consists of convolution layer, normalization terms, and ReLU, as shown in Fig. 5(a). Convolution and ReLU layers are used to hierarchically extract high-level features and improve the nonlinear representation power of the network [36]. 3-D FCN is used to extract the spectral and spatial information simultaneously. ReLU, the most popular activation function, is also used to speed up the training [5]. Considering the fact that the annotated samples is limited and the network structure is relatively shallow, the initialization of parameters has a great impact on the performance of the deep learning model [37]. To address this concern, we introduce the batch normalization (BN) to increase the networks robustness against potential bad initialization. It also can alleviate the risk of overfitting and enable deeper network with limited training samples [38]. According to the findings in previous researches [36], [39], BN is a favorite technique in deep learning, which always can speed up the training and provide improved performance.

First, a convolution kernel with big stride is employed to reduce the spectral redundancy since a few spectral bands may be highly correlated [40]. To avoid the Hughes phenomenon and extract robust spectral features, we employ a convolution kernel rather than the PCA prior to further analysis [5], [41], [42]. Then, in order to better segment HSIs, two branches were designed in our network, including the 3-D branch to jointly extract spatial–spectral features [see Fig. 5(b)] and the 1-D branch to focus on spectral information [see Fig. 5(c)]. Then, these two branches are concatenated and the obtained tensors are sent to classification. In this study, we empirically set the spatial size of the 3-D kernels as 3×3 in the 3-D branch and spectral size of the kernels as three in both the 1-D branch and 3-D branch, as many previous attempts in vision-related tasks [5], [43]. Given that the input patch size is relatively small, we only employ four

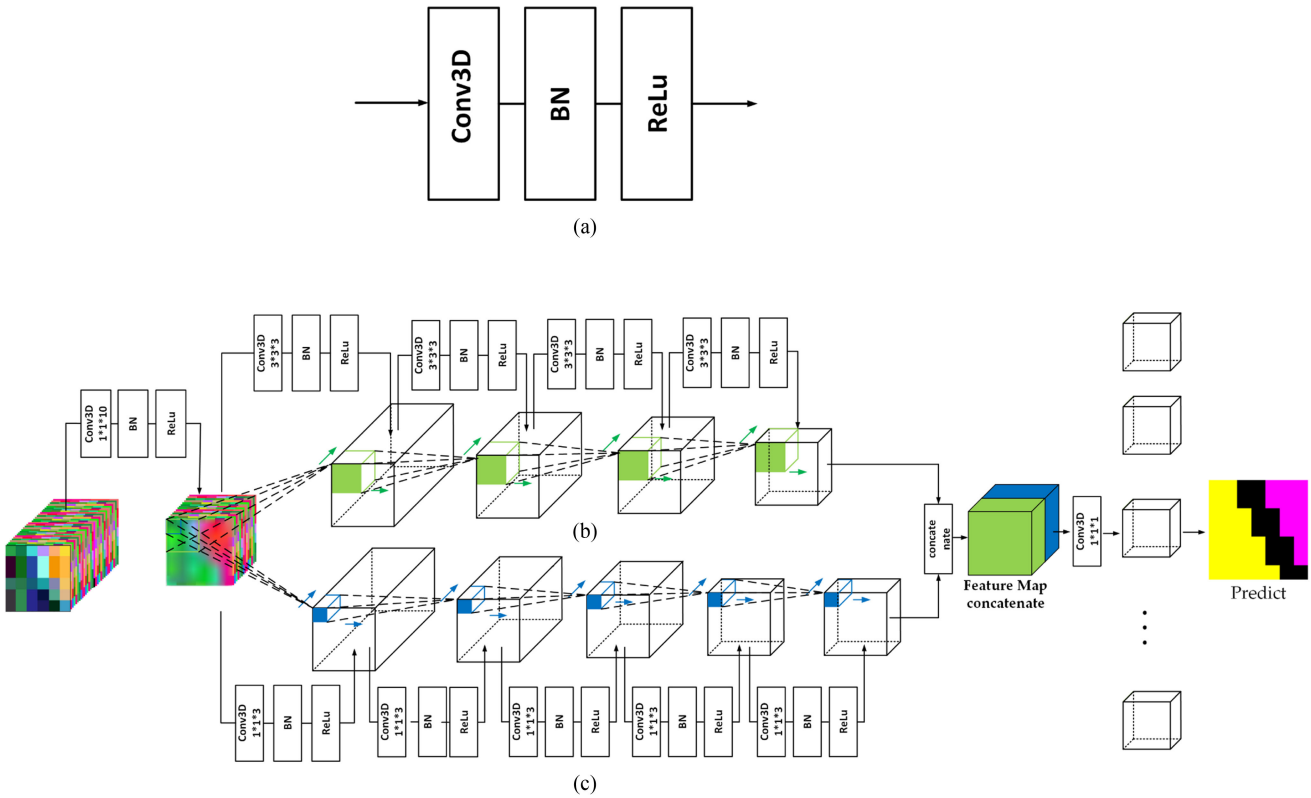


Fig. 5. SS3FCN architecture used for HSI classification. (a) Basic unit of SS3FCN. (b) and (c) Structure of the proposed SS3FCN. Prior to the fusion of two branches, we employ a convolution layer with large stride (1*1*10 for Salinas Valley and Indian Pines, 1*1*6 for Pavia University and Houston University). The features from 3-D branch and 1-D branch are concatenated into 512 feature maps, and forwarded to the last convolution layer.

basic units to extract the spectral–spatial features. Owing to the abundant spectral information within HSIs and its importance for HSI classification, we adopt one specialized 1-D branch with five basic units to extract additional spectral features. Then, the output feature maps from these two branches are concatenated and fed into the subsequent convolution layer to predict each pixels label. Take the Salinas Velley dataset as an example, there are 256 feature maps with the size of $6 \times 6 \times 3$ from the 3-D branch and 256 features maps with the same size from the 1-D branch. Then, these features are concatenated together and forwarded to the final convolution layer. The related code is freely available[1].

IV. Datasets and Experimental Setup

A. Experimental Dataset

As the sizes of these four HSIs are different, we employ different parameters in term of the size of blocks, the number of folds K to split the whole image.

Salinas Valley: The data, captured by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor, is of 512×217 pixels with a spatial resolution of 3.7 m per pixel. It has 224 bands and 16 classes. We split the whole image of Salinas Valley into nine folds, and therefore we repeat the experiment nine times with different training samples. Fig. 6(a) 1–9 are the visualized demonstrations of all the nine folds with white patches for training, and the Fig. 6(a) 0 is the ground-truth for Salinas Valley.

Pavia University: The data was captured by the reflective optics system imaging spectrometer sensor with 103 bands over the Pavia University in northern Italy. The spatial size of the image is 610×340 with high resolution of 1.3 m per pixel. We split the image into ten different folds, as shown in Fig. 6(b) 1–10. Fig. 6(b) 0 is the ground truth for Pavia University.

Indian Pines: This image was captured by AVIRIS sensor over the Indian Pines test site in Northwestern Indiana. It has 145×145 pixels (with a spatial resolution of 20 m per pixel) and 200 bands after removing 20 water absorption bands. It has 16 labeled classes. We set the parameter K as four in the split stage. Fig. 6(c) is the visualization of this dataset, with Fig. 6(c) 0 as the ground truth and Fig. 6(c) 1–4 as four different folds. The white represents the patches used for training in each fold.

Houston University: This image was captured by ITERS-CASI over the University of Houston and the neighboring urban area. The spatial size of this image is 349×1905 with a spatial resolution of 2.5 m per pixel. It has 15 labeled classes. We split the whole image of Houston University Dataset into six folds. Fig. 6(d) 0 shows the ground truth, and the white in Fig. 6(d) 1–6 are the six different folds used for training in each fold.

In the Salinas Valley dataset, 3.76%, 3.76%, and 92.56% of the labeled pixels are utilized for training, validation, and test. The ratio is 6.64%:6.64%:86.72%, 11.02%:11.02%:77.96%, and 18.60%:18.60%:62.80 for Pavia University dataset, Indian Pines dataset, and Houston University dataset, respectively. For

[1][Online]. Available: https://github.com/leonzx7/SS3FCN

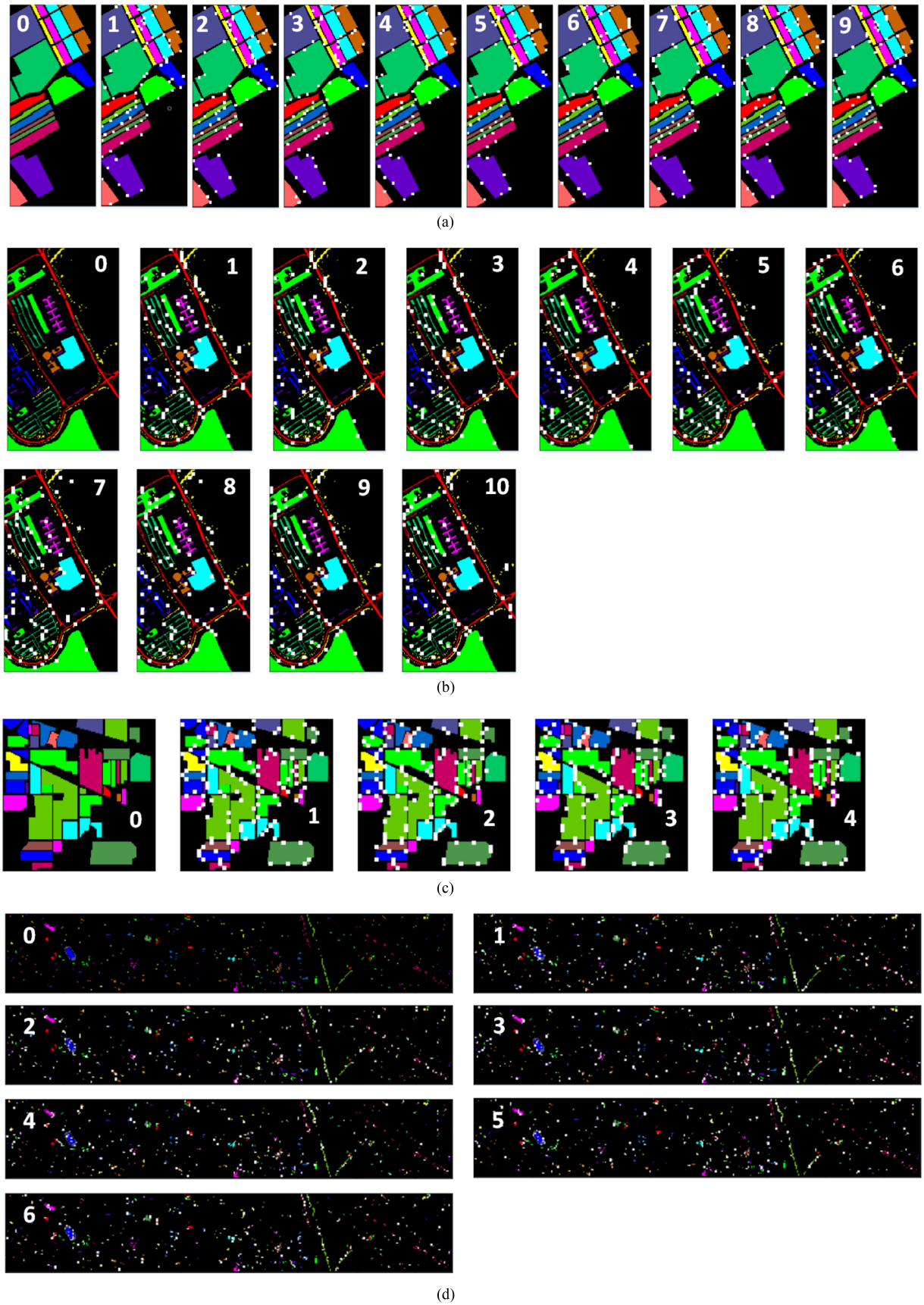


Fig. 6. Training-test splits over (a) Salinas Valley, (b) Pavia University, (c) Indian Pines, and (d) Houston University. Subfigure (a) 0, (b) 0, (c) 0, and (d) 0 are the ground truth for the corresponding datasets. The other subfigures denote the data splits for each dataset, and the white patches are the training samples.

comparison, we also list the number of training and test pixels utilized in one recent paper based on nonoverlap split [30] in Table I. We significantly reduce the ratio for training of Salinas Valley dataset and Indian Pines dataset. As to the Pavia University dataset, this article can provide a more balanced split with a ratio comparable to that in VHIS [30]. For instance, we utilize 121 of 1330 pixels with C7 class for training in this article whereas none of them was selected in [30], as shown in Table I(b). For the Houston University dataset, we used a similar number of pixels to those used in [28].

B. Parameters and Configuration

Considering the variance of the spatial size and the number of spectral bands, the corresponding structures for dealing with these four datasets are slightly different, as shown in Table II. More specially, the kernel size of layer 0 is set as $1 \times 1 \times 6$ with stride of $1 \times 1 \times 3$ for Pavia University and Houston University datasets, smaller than that for the other two datasets (i.e., $1 \times 1 \times 10$ with stride of $1 \times 1 \times 5$). The main reason is that the Pavia University and Houston University datasets only contain 103 bands and 144 bands, approximately 50% less than that in the other two datasets, i.e., 224 and 200 bands.

We employ the focal loss as the loss function and Adam as the optimizer. The focal loss adds a modulating factor to the cross entropy [44]. For binary classification, it is defined as

$$FL(p_t) = -(1-p_t)^r \log(p_t)$$

$$\text{where } p_t = \begin{cases} p & \text{if } y = 1 \\ 1-p & \text{otherwise.} \end{cases}$$

Here, p is the models estimated probability for the class with label $y = 1$, $r \geq 0$ is the focusing parameter. It reduces the easy samples contributions to the loss and focus on hard samples. The Adam is a computationally efficient and well-performing optimization algorithm. The parameters for Adam are set as follows, beta_1 $= 0.9$, beta_2 $= 0.999$, epsilon $= 1 \times e^{-8}$. We set the initial learning rate as 0.01 and shrank it to 1/10 of the previous one after every 35 epochs. Considering the impact of randomness, we repeat the experiments for five times and get the average performance. We compare the proposed 2-branch SS3FCN with the one with only one branch for HSI pixels-to-pixels classification. In addition, we also compare it with state-of-the-art classification methods without information leakage, and show the benefit of our proposed data split method. We report the performance in term of AA and OA across these four datasets. The experiments are implemented using Keras framework on computers with NVIDIA GEFORCE GTX1080 GPU, Intel i7-8700 K processor, and 32 GB RAM.

V. Results and Discussion

In this article, we evaluate the performance of the proposed method on four of the most well-known hyperspectral datasets [45], including Salinas Valley, Pavia University, Indian Pines, and Houston University, and compare it with the recently proposed methods.

TABLE I
AVERAGE NUMBER OF TRAINING/TESTING PIXELS IN FOUR DATASETS

(a)Salinas Valley

No.	Total	SS3FCN				VHIS [30]		
		Train	Val	Test	Ratio(%)	Train	Test	Ratio(%)
C1	2009	82	82	1845	4.08	232	1777	11.55
C2	3726	115	115	3496	3.09	359	3367	9.63
C3	1976	89	89	1798	4.50	185	1791	9.36
C4	1394	155	155	1084	11.12	152	1242	10.90
C5	2678	156	156	2366	5.83	115	2563	4.29
C6	3959	182	182	3595	4.60	296	3663	7.48
C7	3579	169	169	3241	4.72	262	3317	7.32
C8	11271	229	229	10813	2.03	1020	10251	9.05
C9	6203	139	139	5925	2.24	360	5843	5.80
C10	3278	130	130	3018	3.97	424	2854	12.93
C11	1068	108	108	852	10.11	122	946	11.42
C12	1927	116	116	1695	6.02	188	1739	9.76
C13	916	102	102	712	11.14	79	837	8.62
C14	1070	113	113	844	10.56	88	982	8.22
C15	7268	111	111	7046	1.53	842	6426	11.59
C16	1807	43	43	1721	2.38	97	1710	5.37
Total	54129	2037	2037	50051	3.76	4821	49308	8.91

(b)Pavia University

No.	Total	SS3FCN				VHIS [30]		
		Train	Val	Test	Ratio(%)	Train	Test	Ratio(%)
C1	6631	641	641	5349	9.67	531	6100	8.01
C2	18649	791	791	17067	4.24	1250	1250	6.70
C3	2099	210	210	1679	10.00	111	1988	5.29
C4	3064	302	302	2460	9.86	190	2874	6.20
C5	1345	135	135	1075	10.04	127	1218	9.44
C6	5029	176	176	4677	3.50	176	4853	3.50
C7	1330	121	121	1088	9.10	0	1330	0.00
C8	3682	368	368	2946	9.99	240	3442	6.52
C9	947	95	95	757	10.03	74	873	7.81
Total	42776	2839	2839	37098	6.64	2699	23928	6.31

(c)Indian Pines

No.	Total	SS3FCN				VHIS [30]		
		Train	Val	Test	Ratio(%)	Train	Test	Ratio(%)
C1	46	8	8	30	17.39	9	37	19.57
C2	1428	193	193	1042	13.52	352	1076	23.82
C3	830	104	104	622	12.53	207	623	24.94
C4	237	19	19	199	8.02	59	178	24.89
C5	483	61	61	361	12.63	121	362	25.05
C6	730	99	99	532	13.56	182	548	24.93
C7	28	3	3	22	10.71	7	21	25.00
C8	478	24	24	430	5.02	120	358	25.10
C9	20	5	5	10	25.00	5	15	25.00
C10	972	115	115	742	11.83	224	748	23.05
C11	2455	218	218	2019	8.88	591	1864	24.07
C12	593	64	64	465	10.79	148	445	24.96
C13	205	31	31	143	15.12	51	154	24.88
C14	1265	128	128	1009	10.12	316	949	24.98
C15	386	41	41	304	10.62	97	289	25.13
C16	93	16	16	61	17.20	23	70	24.73
Total	10249	1129	1129	7991	11.02	2512	7737	24.51

(d)Houston University

No.	Total	SS3FCN				Others [30]		
		Train	Val	Test	Ratio(%)	Train	Test	Ratio(%)
C1	1251	205	205	842	16.30	198	1053	16.00
C2	1254	242	242	769	19.30	190	1064	15.00
C3	697	116	116	465	16.70	192	505	28.00
C4	1244	211	211	823	16.90	188	1056	15.00
C5	1242	196	196	849	15.80	186	1056	15.00
C6	325	57	57	212	17.40	182	143	56.00
C7	1268	246	246	776	19.40	196	1072	15.00
C8	1244	222	222	800	17.90	191	1053	15.00
C9	1252	259	259	734	20.70	193	1059	15.00
C10	1227	237	237	752	19.30	191	1036	16.00
C11	1235	258	258	720	20.90	181	1054	15.00
C12	1234	238	238	758	19.30	192	1042	16.00
C13	469	105	105	258	22.50	184	285	39.00
C14	428	86	86	257	20.00	181	247	42.00
C15	660	117	117	427	17.70	187	473	28.00
Total	15030	2794	2794	9442	18.60	2832	12198	19.00

TABLE II
ARCHITECTURE OVERVIEW (SHOWN IN COLUMNS)

Layer	Branches	Salinas Valley		Pavia University		Indian Pines		Houston University	
		Spatial	Spectral	Spatial	Spectral	Spatial	Spectral	Spatial	Spectral
Layer0	Kernel size	$1 \times 1 \times 10$		$1 \times 1 \times 6$		$1 \times 1 \times 10$		$1 \times 1 \times 6$	
	Strides	$1 \times 1 \times 5$		$1 \times 1 \times 3$		$1 \times 1 \times 5$		$1 \times 1 \times 3$	
	Filters	64		64		64		64	
Layer1	Kernel size	$3 \times 3 \times 3$	$1 \times 1 \times 3$	$3 \times 3 \times 3$	$1 \times 1 \times 3$	$3 \times 3 \times 3$	$1 \times 1 \times 3$	$3 \times 3 \times 3$	$1 \times 1 \times 3$
	Strides	$1 \times 1 \times 2$	$1 \times 1 \times 2$	$1 \times 1 \times 2$	$1 \times 1 \times 2$	$1 \times 1 \times 2$	$1 \times 1 \times 2$	$1 \times 1 \times 2$	$1 \times 1 \times 2$
	Filters	64	64	64	64	64	64	64	64
Layer2	Kernel size	$3 \times 3 \times 3$	$1 \times 1 \times 3$	$3 \times 3 \times 3$	$1 \times 1 \times 3$	$3 \times 3 \times 3$	$1 \times 1 \times 3$	$3 \times 3 \times 3$	$1 \times 1 \times 3$
	Strides	$1 \times 1 \times 2$	$1 \times 1 \times 2$	$1 \times 1 \times 2$	$1 \times 1 \times 2$	$1 \times 1 \times 2$	$1 \times 1 \times 2$	$1 \times 1 \times 2$	$1 \times 1 \times 2$
	Filters	128	128	128	128	128	128	128	128
Layer3	Kernel size	$3 \times 3 \times 3$	$1 \times 1 \times 3$	$3 \times 3 \times 3$	$1 \times 1 \times 3$	$3 \times 3 \times 3$	$1 \times 1 \times 3$	$3 \times 3 \times 3$	$1 \times 1 \times 3$
	Strides	$1 \times 1 \times 2$	$1 \times 1 \times 2$	$1 \times 1 \times 2$	$1 \times 1 \times 2$	$1 \times 1 \times 2$	$1 \times 1 \times 2$	$1 \times 1 \times 2$	$1 \times 1 \times 2$
	Filters	256	256	256	256	256	256	256	256
Layer4	Kernel size	$3 \times 3 \times 3$	$1 \times 1 \times 3$	$3 \times 3 \times 3$	$1 \times 1 \times 3$	$3 \times 3 \times 3$	$1 \times 1 \times 3$	$3 \times 3 \times 3$	$1 \times 1 \times 3$
	Strides	$1 \times 1 \times 2$	$1 \times 1 \times 2$	$1 \times 1 \times 2$	$1 \times 1 \times 2$	$1 \times 1 \times 2$	$1 \times 1 \times 2$	$1 \times 1 \times 2$	$1 \times 1 \times 2$
	Filters	256	512	256	512	128	256	256	512
Layer5	Kernel size		$1 \times 1 \times 3$		$1 \times 1 \times 3$		$1 \times 1 \times 3$		$1 \times 1 \times 3$
	Strides		$1 \times 1 \times 1$		$1 \times 1 \times 1$		$1 \times 1 \times 1$		$1 \times 1 \times 1$
	Filters		256		256		256		512
Layer_c		concatenate		concatenate		concatenate		concatenate	
Layer6	Kernel size	$1 \times 1 \times 1$		$1 \times 1 \times 1$		$1 \times 1 \times 1$		$1 \times 1 \times 1$	
	Strides	$1 \times 1 \times 1$		$1 \times 1 \times 1$		$1 \times 1 \times 1$		$1 \times 1 \times 1$	
	Filters	17		10		17		16	

A. Pixels-to-Pixels Classification Results

We compare the proposed SS3FCN with state-of-the-art methods based on 3-D CNN without information leakage. Furthermore, to demonstrate the discriminative power of features from the proposed two-branch framework, we also evaluate the performance of networks with single-branch, i.e., the 3-D branch using spectral–spatial information and 1-D branch focusing on spectral information. Both pixels-to-pixels classification and patch-wise classification can be used to assign a categorical label to each pixel in the HSI. We compare the performance corresponding to these two strategies. The abbreviations of the networks to compare are as follows.

VHIS [30]: The recently proposed method utilizing 1-D network based on patch-based data split without training-test overlap.

3-D CNN [6]: The patch-wise classification exploring the spectral–spatial information based on the patch-based data split without training-test overlap.

PCA/PCA-ON [35]: The data augmentation strategy including PCA-based offline setting and PCA online setting.

GAN/PCA-ON [35]: The data augmentation strategy including GAN-based offline setting and PCA online setting.

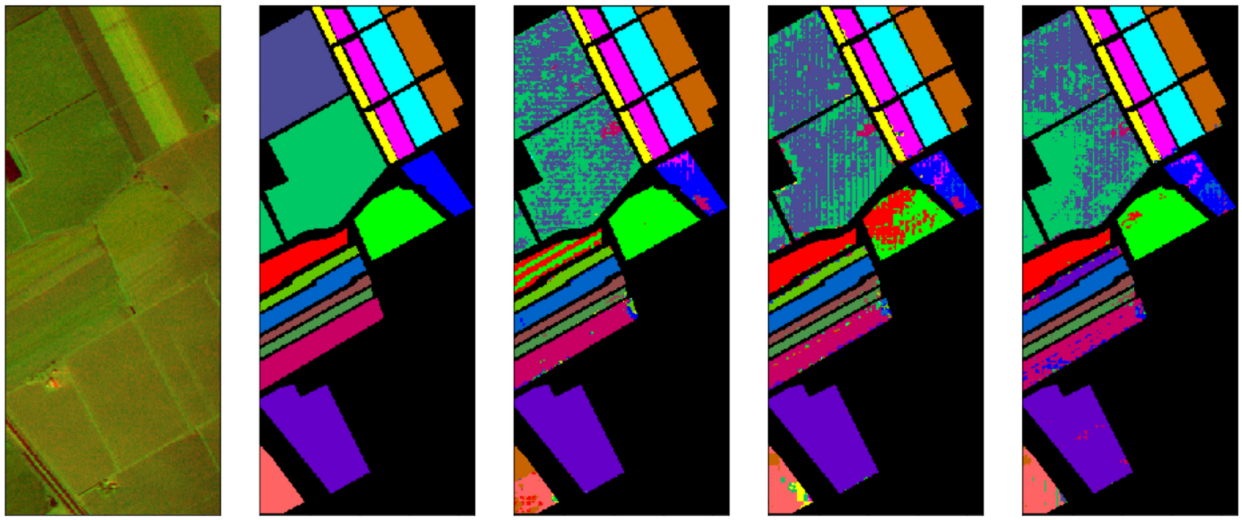
3-D Branch: The 3-D branch exploring the spectral–spatial information, as shown in Fig. 5(b).

1-D Branch: The 1-D branch focusing on spectral information, as shown in Fig. 5(c).

1) Salinas Valley: In the first experiment, we conduct our study on the Salinas Valley dataset. In the dataset split stage, we set the size of blocks (i.e., $W \times H$) as 7×7 and the size of patches (i.e.,$D \times D$) as 6×6 for this dataset. The pixels-to-pixels classification performance provided by different methods are shown in Table III. Overall, even with less training samples, the proposed SS3FCN achieves the best performance (with OA $= 81.32\%$ and AA $= 86.13\%$), followed by SS3FCN 1-D branch, SS3FCN 3-D branch, PCA/PCA-ON augmentation, and GAN/PCA-ON augmentation, 3-D CNN and VHIS 1-D CNN without data augmentation. Although both SS3FCN and the methods for patch-based classification are based on 3-D CNN, the SS3FCN interpret the problem for assigning label to each pixel as a pixels-to-pixels classification problem. Therefore, the proposed method can efficiently exploit the spectral–spatial information from all the pixel with labels, whereas the patch-based classification cannot for the information leakage issue. The comparison among the classification strategies demonstrates that the proposed two-branch-based method outperform the networks with only one branch. The OA&AA are increased from 77.31% & 82.32% to 81.32% & 86.13%, respectively, via introducing the additional 3-D branch. The learned spatial representations provide complementary information to the spectral features. It also highlights that the spectral features may be more discriminative than spatial features for small HSI analysis. In addition, the detailed accuracies corresponding to 16 classes averaged over five runs are also listed in Table III. Since the descriptions about the network setting in [30] are not detailed enough to produce the classification maps corresponding the performance shown in Table III, we visually report the classification results of the methods based on pixels-to-pixels classification

TABLE III
PERFORMANCE OF DIFFERENT METHODS FOR THE SALINAS VALLEY DATASET(%)

No.	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	OA	AA
1	99.64	84.60	84.26	98.21	94.17	98.66	99.73	74.51	99.83	91.25	93.94	99.48	95.63	96.95	50.21	72.21	83.71±0.55	89.58±0.55
2	99.74	93.13	74.00	95.67	96.37	98.99	99.78	94.79	99.66	86.68	95.16	99.61	99.44	93.89	4.28	79.25	81.79±0.35	88.15±0.38
3	36.89	99.63	72.64	99.30	99.19	99.54	99.54	46.47	99.67	90.59	97.27	99.62	99.56	85.39	77.04	41.37	78.83±0.745	83.98±0.88
4	99.55	87.05	41.19	99.42	97.43	99.43	98.78	81.65	99.86	92.68	98.41	97.16	98.62	83.37	34.78	74.00	81.75±0.25	86.46±0.38
5	99.19	99.83	48.83	98.60	99.00	99.63	99.81	61.96	99.80	55.67	99.56	89.82	98.06	94.58	73.56	0.05	79.36±0.90	82.37±0.11
6	97.75	99.41	56.90	97.19	97.96	99.43	99.67	94.18	99.91	84.02	96.02	97.67	97.03	94.79	24.36	98.77	84.92±0.24	89.69±0.40
7	99.92	71.63	62.13	97.71	83.98	98.44	99.05	76.99	99.49	87.77	97.49	99.96	99.74	84.67	53.94	19.24	80.31±0.45	83.26±1.41
8	99.30	98.74	87.53	98.81	97.23	99.82	99.16	77.22	98.84	82.93	2.82	99.17	98.39	83.42	59.85	90.12	85.03±0.90	85.83±2.46
9	99.22	99.23	69.65	98.28	95.35	99.80	99.34	15.68	99.98	85.07	87.13	99.30	99.56	68.85	92.81	64.76	76.14±0.75	85.87±0.80
SS3FCN	92.36	**92.58**	**66.35**	**98.13**	95.63	99.30	99.43	69.27	**99.67**	84.07	85.31	**97.98**	98.45	87.32	52.31	59.97	**81.32±2.80**	**86.13±2.31**
1D Branch	90.01	90.88	64.32	97.02	**96.4**	99.66	**99.73**	65.01	97.39	79.61	**84.65**	91.28	**99.36**	87.68	49.32	53.35	78.80±2.62	84.10±1.73
3D Branch	76.08	80.73	60.56	96.19	92.80	99.34	99.16	64.00	96.79	75.85	83.31	93.59	97.24	88.40	51.88	61.27	77.31±3.32	82.32±3.56
SS3FCN-VHIS	62.67	84.87	42.53	93.15	93.78	**99.74**	96.79	**86.87**	92.22	74.57	58.99	65.16	90.12	**92.31**	18.86	**63.36**	74.60±2.30	76.00±3.11
VHIS [30]	85.91	73.88	33.72	65.92	46.42	79.63	73.59	72.16	71.87	73.11	72.51	71.06	75.80	72.04	45.03	22.54	64.20	64.70
PCA/PCA-ON [35]	95.88	90.06	47.1	79.71	66.76	79.84	79.62	78.14	94.24	**87.65**	73.19	90.75	98.35	87.57	**57.01**	46.11	76.67	78.25
GAN/PCA-ON [35]	96.02	86.1	50.76	79.73	68.45	79.73	79.67	78.02	94.03	85.72	70.95	89.08	96.93	85.54	54.66	43.87	75.87	77.45
3DCNN [6]	**96.49**	75.15	39.89	61.61	52	79.21	76.81	74.84	78.14	85.69	71.56	76.49	80.86	62.15	61.8	33	69.72	69.09



(a) (b) (c) (d) (e)

Fig. 7. Predicted maps of different models for Salinas Valley via one fold of the cross validation. (a) False color image. (b) Ground truth. (c) 1-D branch network (OA: 79.82%, AA: 85.16%). (d) 3-D branch network (OA: 78.49%, AA: 83.04%). (e) SS3FCN (OA: 81.75%, AA: 86.46%).

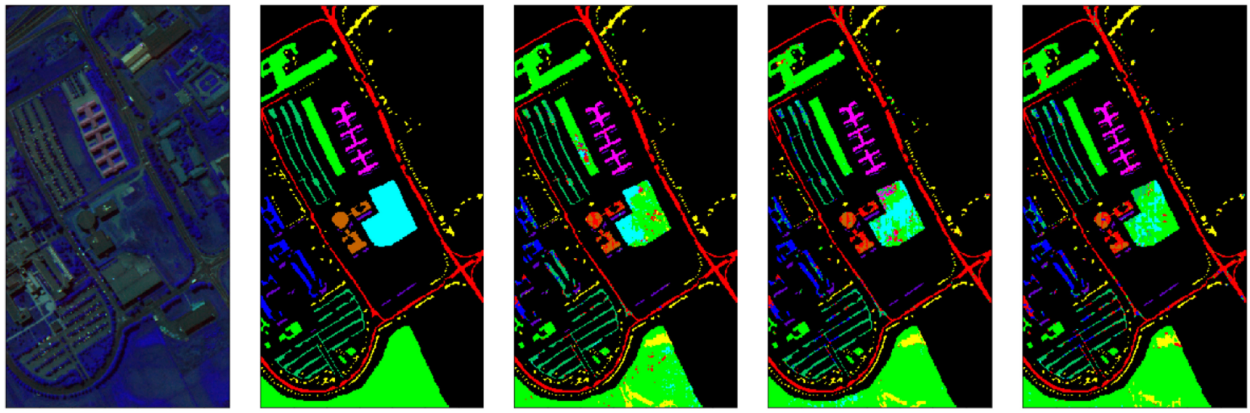
only in Fig. 7. The qualitative comparisons among different classification methods are in line with quantitative comparison in Table III.

2) Pavia University: In the second experiment, we conduct our study on the Pavia University dataset. Considering the size of this HSI is relatively large, we select the block size as 11×11 and the patch size as 10×10 to avoid ignoring part of the spatial information. The accuracies acquired by different classification methods are reported in Table IV. Similar to the results for Salinas Valley dataset, the proposed two-branch SS3FCN achieves the best performance. The OA/AA of the proposed two-branch SS3FCN is 79.89%/76.60%, significantly better than the performance based on patch-wise classification. In addition, the proposed strategy for training-test splits provides a more balanced dataset. As shown in Table I(b), 9.10% of pixels in class C7 are selected for training based on our proposed data partition method, whereas that ratio of patch-based method VHIS is 0. This may be the main reason for failing to predict the class C7 in VHIS, 3-D CNN, PCA/PCA-ON, and GAN/PCA-ON. In addition, we report the detailed accuracies corresponding to 16 classes averaged over five runs in Table IV, and show the resulted maps corresponding to one of these five runs in Fig. 8.

3) Indian Pines: Table V shows the overall and class-specific performance acquired by different methods for Indian Pines dataset, and Fig. 9 visually demonstrates the resulted maps. If the input patches are too small, some of the spatial information may be ignored. Therefore, considering the tradeoff between the spatial size of the input patches and the number of training samples in each class, in this experiment, we set the block size as 4×4 and the patch size as 3×3. Overall, 11.02% of pixels are selected to train the SS3FCN, less than the pixels used in VHIS and the data augmentation methods (i.e., 24.5%). However, the proposed method achieves 4.36%/5.54% better performance than VHIS in term of the OA and AA. In addition,

TABLE IV
PERFORMANCE OF DIFFERENT METHODS FOR THE PAVIA UNIVERSITY DATASET(%)

No.	C1	C2	C3	C4	C5	C6	C7	C8	C9	OA	AA
1	96.57	89.31	64.63	90.02	95.22	31.31	50.12	92.28	93.04	81.31±1.50	78.06±1.95
2	97.40	82.70	34.91	80.97	96.95	11.45	74.34	92.03	96.38	74.07±4.35	74.13±6.61
3	98.60	93.98	33.41	84.02	99.37	12.20	73.08	89.76	88.38	78.82±1.75	74.76±2.64
4	96.62	92.45	72.12	86.33	98.82	34.32	33.64	81.77	91.76	81.47±1.40	76.42±2.25
5	99.06	89.19	51.60	80.50	93.73	38.31	65.30	89.39	91.00	80.83±1.83	77.56±3.20
6	94.78	88.78	35.95	92.07	93.69	43.00	27.63	94.11	61.40	79.55±5.69	70.16±5.27
7	96.92	92.62	66.64	88.00	84.96	25.39	72.12	90.68	92.86	81.68±1.73	78.91±1.29
8	98.83	96.31	72.34	78.58	97.73	11.99	74.80	83.14	88.35	80.90±6.32	78.01±4.57
9	98.06	90.25	82.22	84.06	95.14	11.60	72.05	87.99	92.42	79.53±2.60	79.31±3.51
10	97.93	93.06	73.73	83.55	92.61	16.33	72.99	87.27	91.21	80.70±1.36	78.74±3.40
SS3FCN	**97.48**	90.86	**58.75**	84.81	94.82	23.59	**61.61**	88.84	88.68	**79.89±2.13**	**76.60±2.71**
1D Branch	97.16	89.06	58.69	87.14	97.93	20.07	60.01	89.17	87.22	78.79±3.87	76.27±6.27
3D Branch	97.44	**91.07**	55.15	80.67	92.45	**31.34**	52.76	81.96	73.01	79.29±2.58	72.87±5.38
SS3FCN-VHIS	85.19	90.40	14.83	**92.98**	**99.37**	17.63	46.35	**89.59**	**99.06**	76.23±1.63	70.60±3.53
VHIS [30]	93.40	86.20	47.58	86.89	59.81	27.14	0.00	78.46	79.27	73.26	62.08
PCA/PCA-ON [35]	93.42	86.52	46.88	92.21	59.74	27.68	0.00	78.32	79.60	73.84	62.71
GAN/PCA-ON [35]	92.87	81.83	47.51	91.53	59.83	28.90	0.00	74.86	79.53	71.46	62.83
3DCNN [6]	90.66	81.85	41.92	93.02	59.79	25.20	0.00	70.18	79.03	70.07	60.18



(a) (b) (c) (d) (e)

Fig. 8. Predicted maps of different models for Pavia University via one fold of the cross validation. (a) False color image. (b) Ground truth. (c) 1-D branch network (OA: 79.20%, AA: 76.31%). (d) 3-D branch network (OA: 78.14%, AA: 74.15%). (e) SS3FCN (OA: 81.31%, AA: 78.06%).

TABLE V
PERFORMANCE OF DIFFERENT METHODS FOR THE INDIAN PINES DATASET(%)

No.	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	OA	AA
1	98.17	71.98	55.42	11.81	88.39	92.95	80.56	62.46	0.00	85.36	72.66	34.32	97.35	83.00	54.44	82.63	71.39±1.24	66.97±4.29
2	34.04	75.75	50.43	3.89	71.10	87.17	0.00	81.36	100.00	67.91	75.35	41.11	84.50	98.84	34.04	82.14	71.35±2.37	61.73±4.10
3	29.39	77.01	69.22	17.39	54.65	93.71	0.00	97.03	26.67	74.18	66.86	39.68	95.31	97.96	34.53	86.74	72.08±2.03	60.02±3.45
4	0.00	86.83	67.90	14.11	55.85	93.96	0.00	85.96	0.00	85.16	62.39	48.12	96.56	87.26	28.72	49.25	71.05±2.28	53.88±2.10
SS3FCN	40.40	77.89	**60.74**	11.80	67.50	91.95	20.14	81.71	31.67	**78.15**	69.32	40.81	**93.43**	91.77	37.93	75.19	**71.47±0.41**	**60.65±4.54**
1D Branch	16.81	**82.83**	52.19	26.93	50.33	84.79	23.29	**92.78**	**53.13**	56.62	67.92	35.70	91.23	**92.74**	37.55	79.93	69.44±1.60	59.05±3.82
3D Branch	38.38	55.78	49.29	7.89	53.11	78.85	**24.54**	52.07	21.26	43.54	68.94	27.48	90.02	87.98	31.80	67.65	59.56±3.52	49.91±5.97
SS3FCN-VHIS	**50.34**	45.96	17.84	8.35	58.65	89.38	17.19	52.49	1.52	41.26	59.14	8.77	91.81	82.96	15.90	**88.03**	52.26±2.83	45.60±1.09
VHIS [30]	17.68	56.89	51.55	36.27	69.02	92.35	0	86.95	19.55	60.05	**74.05**	43.71	94.15	91.18	**43.39**	45.04	67.11	55.11
PCA/PCA-ON [35]	16.07	67.14	55.85	**43**	**70.2**	**93.1**	23.94	70.63	33.18	64.66	68.9	**52.15**	91.96	79.32	34.43	44.45	68.57	56.81
GAN/PCA-ON [35]	14.64	62.21	52.04	36.53	69.02	87.8	23.8	68.12	27.44	62.67	67.44	48.21	89.1	76.19	35.2	44.49	65.97	54.06
3DCNN [6]	5	33.7	28.3	17.88	51.32	60.18	0	65.99	1.67	53.06	54.27	23.2	65.87	77.01	37.95	37.94	48.89	38.33

(a) (b) (c) (d) (e)

Fig. 9. Predicted maps of different models for Indian Pines via one fold of the cross validation. (a) False color image. (b) Ground-truth map. (c) 1-D branch network (OA: 70.09%, AA: 62.74%). (d) 3-D branch network (OA: 60.68%, AA: 59.71%); (e) SS3FCN (OA: 71.01%, AA: 66.66%).

TABLE VI
PERFORMANCE OF DIFFERENT METHODS FOR THE HOUSTON UNIVERSITY DATASET(%)

No.	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	OA	AA
1	97.15	78.53	95.92	94.88	98.99	88.47	84.01	49.74	75.66	92.99	85.55	71.57	70.83	98.58	94.92	83.95±0.84	85.19±0.93
2	84.80	97.40	99.06	95.62	98.67	74.36	93.93	76.41	71.63	83.06	79.84	65.95	68.13	96.30	96.23	85.36±2.84	85.43±2.81
3	98.90	80.86	99.28	93.01	99.78	66.82	84.94	67.98	79.21	88.06	74.76	56.99	76.31	95.92	95.28	83.44±0.38	83.87±0.48
4	96.78	90.22	98.52	91.84	99.36	71.91	86.14	66.07	79.64	76.69	73.04	62.86	52.00	85.01	68.61	80.66±1.6	79.91±2.50
5	90.91	85.90	99.59	89.57	86.67	79.03	80.72	66.62	82.21	70.51	71.46	79.18	66.32	98.14	92.66	81.93±2.60	82.63±2.00
6	99.50	79.81	99.21	92.45	94.11	75.22	87.75	71.44	80.17	79.40	73.10	74.57	78.85	92.25	98.55	84.46±2.7	85.09±2.20
SS3FCN	94.67	85.45	98.60	92.90	96.26	75.97	**86.25**	**66.37**	**78.09**	**81.78**	76.29	68.52	68.74	**94.37**	91.04	**83.30±1.57**	**83.69±1.94**
1D Branch	**96.73**	82.10	**99.44**	**93.02**	94.03	**82.10**	82.32	56.57	75.55	77.82	**77.29**	**70.82**	54.95	93.57	90.73	81.25±2.25	81.80±2.05
3D Branch	91.76	**87.04**	92.14	92.91	**97.44**	66.09	79.24	58.18	74.34	75.80	66.53	55.75	**77.67**	89.31	87.50	78.78±4.75	79.45±2.94
SS3FCN-VHIS	90.18	87.78	98.39	93.87	94.60	89.63	77.21	58.71	73.48	65.01	65.15	41.39	41.85	91.01	**97.26**	76.28±2.97	77.71±2.82

compared with the two-branch framework, the SS3FCN 3-D branch fails in the prediction of many more classes. A major reason might be that the input patches are too small to reserve the spatial information.

4) Houston University: As to the Houston University dataset, although the image is large, the pixels with label information are rather limited. We select the block size as 7×7 and patch size 6×6 to extract the edge of small pieces. Table VI shows classification accuracies acquired by different methods, and Fig. 10 visually demonstrates the resulted maps. Similar to the results of the above four datasets, the 2-branch SS3FCN get the best result. With the same training and test pixels, the performance of the proposed 2-branch SS3FCN network is higher by 7.02%/8.98% better performance than the 1-D network proposed in VHIS in term of the OA and AA.

5) Overall Comparison: In Table VII, we show the overall difference between the proposed 2-branch SS3FCN with the other pixel labeling methods based on nonoverlapped training/test set, including 1-branch SS3FCN, VHIS, 3-D CNN, PCA/PCA-ON augmentation and GAN/PCA-ON augmentation. Theoretically, the 3-D branch can provide better results than the 1-D branch. However, the 1-D branch outperforms the 3-D one in three of these four datasets. We suspect that the main reason is that the number of training samples is limited and the 3-D branch cannot extract enough amount of spatial information. Comparing with single branch SS3FCN, the two-branch strategy model obtained better performance for its ingenious design of the network architecture. It aims to fully exploit the contribution from the spectral and spatial information. It also demonstrates that the abundant spectral information should be given more attentions in analyzing HSIs with low spatial resolution.

In addition, comparing with VHIS and the other VHIS-based networks, the proposed two-branch SS3FCN achieves a big improvement. For instance, as in Salinas Valley, OA is increased by up to 17.12% and AA is increased by up to 21.43%. Even without data augmentation via advanced ways, the two-branch SS3FCN still can get better performance.

B. Analysis and Discussion

1) Training Process: The Fig. 11 demonstrates the training loss and validation loss with respect to the number of training epochs on Salinas Valley, Pavia University, and Indiana Pines, respectively. The initial learning rate is set as 0.01 and decayed to 1/10 of the previous learning rate after every 35 epochs. The networks are kept for training until the validation loss does not decrease in the next 20 epochs and get the model with the best performance on the validation dataset. As can see from Fig. 11, the models for Salinas Valley dataset and Pavia University get converged faster than that for Indian Pines dataset.

2) Impact of the Size of Input Patches: The Fig. 12 demonstrates the influence of the spatial size of selected patches to the performance on four datasets. The network structures are same as that in Table II except for the patch size smaller than 3×3 where we set the spectral–spatial kernel to be $1 \times 1 \times 3$, rather than $3 \times 3 \times 3$. We evaluate the performance when the width (i.e., D, equal to the height) of patches is set to be [4, 6, 8, 10], [6, 8, 10, 12], and [1, 3, 5, 7] for Salinas Valley, Pavia University, and Indiana Pines dataset, respectively. The obtained optimal patch size is 6×6, 10×10, 3×3 accordingly. Theoretically speaking, the larger patches, the better performance will be achieved. However, the size of the original HSI is limited. Given

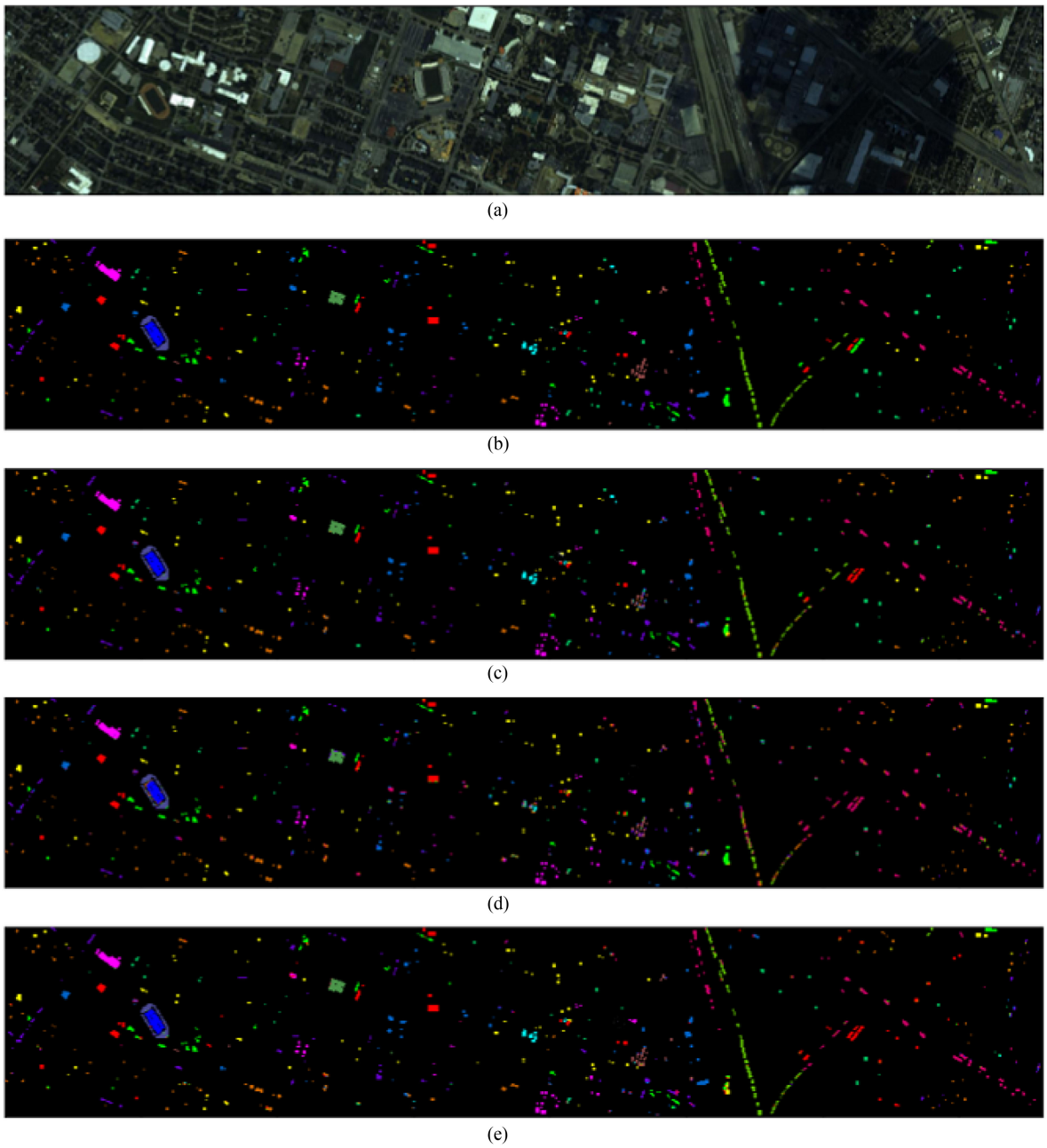


Fig. 10. Predicted maps of different models for Houston University dataset via one fold of the cross validation. (a) False color image. (b) Ground truth. (c) 1-D branch network (OA: 81.93%, AA: 83.17%). (d) 3-D branch network (OA: 79.52%, AA: 79.47%). (e) SS3FCN (OA: 83.44%, AA: 83.87%).

larger patches, the number of samples might be too less to obtain satisfying network. For instance, the number of patches will be reduced by 31% when the patch size is changed from 10×10 to 12×12.

3) Impact of the Block Size: We evaluate the performance corresponding to the blocks with different sizes. Taking the Salinas Valley for example (see Fig. 13), the proposed method achieve the best performance with OA/AA of 81.32%/86.13% when the block size is 7×7. We suspect that, given smaller blocks, the network cannot fully exploit the spatial information. If we further increase the block size (larger than 7×7), in order to keep the ratio of training pixels unchanged (to make a fair comparison with other works), we have to increase the value of K and get less blocks input to the network. Given less training samples (patches), the performance of the network deteriorates. Similar results are obtained in analyzing the other three datasets.

4) Classification Performance With Different Initializers: Weight initialization aims to prevent layer activation outputs from exploding or vanishing in calculating the output values of each layer. If either occurs, the network has to take longer time to converge, or even does not converge at all. In this article, we evaluate the performance with three initialization strategies on Salinas Valley dataset and the performance is shown in Table VIII.

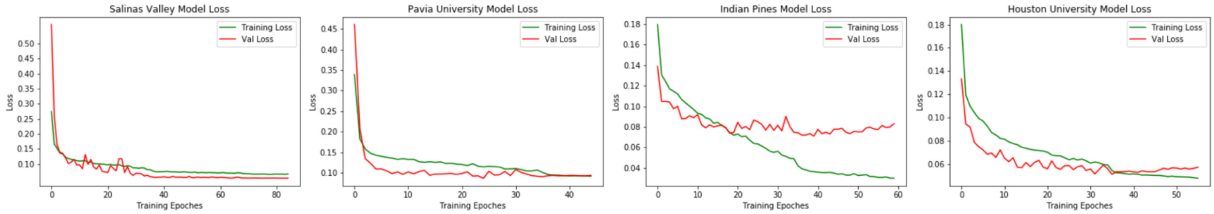


Fig. 11. Loss curves of training and validation loss over epochs for four datasets. (a) Salinas Valley. (b) Pavia University. (c) Indian Pines. (d) Houston University.

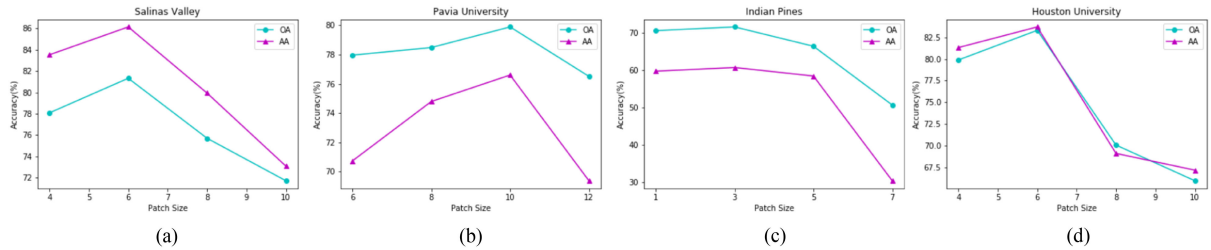


Fig. 12. Influence of the patch sizes on OA and AA. (a) Salinas Valley. (b) Pavia University. (c) Indian Pines. (d) Houston University.

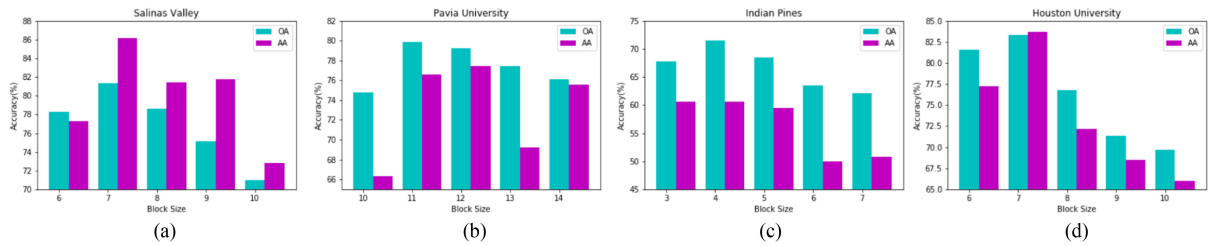


Fig. 13. Influence of the block sizes on OA and AA. (a) Salinas Valley. (b) Pavia University. (c) Indian Pines. (d) Houston University.

TABLE VII
AVERAGE DIFFERENCE IN OA AND AA BETWEEN DIFFERENT MODELS(%)

	SalinasV		PaviaU		IndianP		HoustonU	
	OA	AA	OA	AA	OA	AA	OA	AA
SS3FCN	81.32	86.13	79.89	76.60	71.47	60.65	83.3	83.69
1D Branch	78.80	84.10	78.79	76.27	69.44	59.05	81.25	81.80
	2.52↑	2.03↑	1.10↑	0.33↑	2.03↑	1.60↑	2.05↑	1.89↑
3D Branch	77.31	82.32	79.29	72.87	59.56	49.91	78.78	79.45
	4.01↑	3.81↑	0.61↑	3.73↑	11.91↑	10.74↑	4.52↑	4.24↑
SS3FCN-VHIS	74.60	76.00	76.23	70.60	51.90	44.00	76.28	77.71
	6.72↑	10.13↑	3.66↑	6.00↑	19.57↑	16.65↑	7.02↑	5.98↑
VHIS [30]	64.20	64.70	73.26	62.08	67.11	55.12	\	\
	17.12↑	21.43↑	6.63↑	14.52↑	4.36↑	5.53↑	\	\
PCA/PCA-ON [35]	76.67	78.25	73.84	62.71	68.57	56.81	\	\
	4.65↑	7.88↑	6.05↑	13.89↑	2.90↑	3.84↑	\	\
GAN/PCA-ON [35]	75.87	77.45	71.46	62.83	65.97	54.06	\	\
	5.45↑	8.68↑	8.43↑	13.77↑	5.50↑	6.59↑	\	\
3DCNN [6]	69.72	69.09	70.07	60.18	48.89	38.33	\	\
	11.60↑	17.04↑	9.82↑	16.42↑	22.58↑	22.32↑	\	\

For the random strategy, we randomly initialize the convolution kernels. Glorot_uniform, also known as Xavier uniform, draws samples from a uniform distribution [46]. He_normal draws samples from a truncated normal distribution centered on 0

TABLE VIII
INFLUENCE OF INITIALIZERS FOR THE SALINAS VALLEY DATASET

	Random	Glorot_uniform	He_normal
OA(%)	74.98±2.10	80.33±1.33	81.85±1.14
AA(%)	75.51±2.99	83.83±1.36	84.45±0.84
Epochs	118	61	51

with $stddev = sqrt(2/fan\_in)$ where fan_in is the number of input units in the weight tensor [47]. As shown in Table VIII, the network with He_normal initializations converges after 51 epoches and achieves OA/AA of 81.85%/84.45%. The experiments on the other three datasets also suggest that He_normal initializations is much more effective and efficient than the other two ways.

5) Impact of BN: In this article, we introduce the BN to increase the network's robustness against potential bad initialization. We evaluate the performance without BN (denoted as No_BN), with only one BN layer after the first convolutional layer (denoted as 1_BN), with one BN layer after each convolutional layer (denoted as All_BN). The results are shown in Table IX. Taking the Salinas Valley dataset for example, All_BN

TABLE IX
INFLUENCE OF BN LAYER ON THE OA, AA AND TRAINING SPEED

			SalinasV	PaviaU	IndianP	HoustonU
All_BN	Accuracy	OA(%)	81.32±2.80	79.89±2.13	71.47±2.87	83.30±1.72
		AA(%)	86.13±2.31	76.60±2.71	60.65±5.40	86.69±2.13
	Epoches		27.4(65%)	31.2(55%)	22.75(55%)	20.5(74%)
1_BN	Accuracy	OA(%)	76.00±2.30	77.28±2.54	66.75±2.16	73.89±3.39
		AA(%)	81.08±2.02	71.63±3.46	56.44±3.83	74.39±3.84
	Epoches		30.5(65%)	35.2(55%)	30.25(55%)	28.75(74%)
No_BN	Accuracy	OA(%)	21.48	43.71	10.5	7.88
		AA(%)	6.25	10	10.55	6.65
	Epochs		\	\	\	\

TABLE X
PERFORMANCE WITH DIFFERENT NETWORK DEPTH FOR THE SALINAS VALLEY DATASET

	3D Branch	1D Branch	OA(%)	AA(%)
Network Depth	4	5	81.32±2.80	86.13±2.31
	5	5	79.99±3.34	85.11±1.87
	4	6	78.73±2.94	85.83±3.24
	5	6	77.01±0.53	82.86±2.99

TABLE XI
PERFORMANCE WITH DIFFERENT K FOR SALINAS VALLEY DATASET

	K=5	K=7	K=9	K=11
Train Pixels	3662	3016	2839	1665
Ratio(%)	6.77	4.83	3.76	3.07
OA(%)	84.46±2.34	82.05±1.96	81.32±2.80	78.29±3.21
AA(%)	92.39±1.21	89.99±1.09	86.13±2.31	85.78±2.22

network achieves an accuracy of 65% after 27.4 epochs on the validation dataset, whereas the 1_BN cannot get similar results until 30.5 epochs in average, and the NO_BN network cannot get the similar results. In addition, the OA/AA of All_BN network converges at 81.32%/86.13%, significantly higher than that of the 1_BN and No_BN network.

6) Impact of Network Depth: In this work, we also evaluate the performance of the network with different depth. For simplicity, we also take the Salinas Valley for example. We test the combination of 4/5 layers in the 3-D branch and 5/6 layers in the 1-D branch. As shown in Table X, the combination of four 3-D convolutional layers and five 1-D convolutional layers provides the best performance. The performance degrades if we further increase the network depth, given the training samples are limited.

7) Performance With Various Number of Pixels in the Training Set: In this work, we evaluate the performance with different K, corresponding to various number of pixels used for training, and the result on Salinas Valley dataset is shown in Table XI. With the decrease of the number of pixels for training from 6.77% to 3.07%, the performance in term of OA/AA degrades from 84.46%/92.39% to 78.29%/85.78%. Similar results are obtained for the other three datasets.

VI. CONCLUSION

With the great success of fully convolutional network in many vision-related problems, we are motivated to develop a novel pixels-to-pixels classification framework based on the fusion of FCNs to assign label to each pixel in HSI, referred to as SS3FCN. The literatures demonstrate that the emerging patch-wise HSI classification methods can easily lead to overoptimistic results for the information leakage issue. Inspired by this, we first introduce a simple method for splitting the training/test dataset, which provides a more balanced split. Considering the complementary characters, we employ the 3-D FCN to jointly explore the spectral-spatial information. In light of the importance of the spectral information in HSI classification and the relatively small input patches in many datasets, we introduce one additional branch based on 1-D CNN to get more discriminative and robust features. In addition, we interpret the task of assigning label for each pixel as a pixels-to-pixels classification problem where all the label information is fully exploited, different from the ways in traditional patch-wise classifications. The proposed method takes advantages of both 3-D FCN and the training/test split strategy, and achieves better performance with less training pixels when it is compared with state-of-the-art classification methods. Furthermore, the basic structure including 3-D FCN, BN, and ReLU layer can be easily generalized to other HSI datasets for their simple architectures and powerful learning abilities.

REFERENCES

[1] X. Li, Z. Yuan, and Q. Wang, "Unsupervised deep noise modeling for hyperspectral image change detection," *Remote Sens.*, vol. 11, no. 3, 2019, Art. no. 258.
[2] M. J. Khan, H. S. Khan, A. Yousaf, K. Khurshid, and A. Abbas, "Modern trends in hyperspectral image analysis: A review," *IEEE Access*, vol. 6, pp. 14 118–14 129, 2018.
[3] J. Transon, R. D'Andrimont, A. Maugnard, and P. Defourny, "Survey of hyperspectral earth observation applications from space in the sentinel-2 context," *Remote Sens.*, vol. 10, no. 2, 2018, Art. no. 157.
[4] Y. Xu, B. Du, F. Zhang, and L. Zhang, "Hyperspectral image classification via a random patches network," *ISPRS J. Photogrammetry Remote Sens.*, vol. 142, pp. 344–357, 2018.
[5] Y. Li, H. 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.
[6] Q. Gao, S. Lim, and X. Jia, "Hyperspectral image classification using convolutional neural networks and multiple feature learning," *Remote Sens.*, vol. 10, no. 2, 2018, Art. no. 299.
[7] F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 8, pp. 1778–1790, 2004.
[8] M. Belgiu and L. Drăguţ, "Random forest in remote sensing: A review of applications and future directions," *ISPRS J. Photogrammetry Remote Sens.*, vol. 114, pp. 24–31, 2016.
[9] Z. Zhong, J. Li, Z. Luo, and M. Chapman, "Spectral–spatial residual network for hyperspectral image classification: A 3-D deep learning framework," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 2, pp. 847–858, Feb. 2018.
[10] Y. Yuan, J. Lin, and Q. Wang, "Hyperspectral image classification via multitask joint sparse representation and stepwise MRF optimization," *IEEE Trans. Cybern.*, vol. 46, no. 12, pp. 2966–2977, Dec. 2016.
[11] Y. Tarabalka and A. Rana, "Graph-cut-based model for spectral-spatial classification of hyperspectral images," in *Proc. IEEE Geosci. Remote Sens. Symp.*, Jul. 2014, pp. 3418–3421.
[12] X. Kang, S. Li, and J. A. Benediktsson, "Spectral–spatial hyperspectral image classification with edge-preserving filtering," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 5, pp. 2666–2677, May 2014.

[13] R. Kemker, C. Salvaggio, and C. Kanan, "Algorithms for semantic segmentation of multispectral remote sensing imagery using deep learning," *ISPRS J. Photogrammetry Remote Sens.*, vol. 145, pp. 60–77, 2018.
[14] D. Hong, N. Yokoya, N. Ge, J. Chanussot, and X. X. Zhu, "Learnable manifold alignment (LEMA): A semi-supervised cross-modality learning framework for land cover and land use classification," *ISPRS J. Photogrammetry Remote Sens.*, vol. 147, pp. 193–205, 2019.
[15] D. Hong, N. Yokoya, J. Chanussot, and X. X. Zhu, "Cospace: Common subspace learning from hyperspectral-multispectral correspondences," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 7, pp. 4349–4359, Jul. 2019.
[16] J. Lin, L. Zhao, S. Li, R. Ward, and Z. J. Wang, "Active-learning-incorporated deep transfer learning for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 11, pp. 4048–4062, Nov. 2018.
[17] J. Lin, C. He, Z. J. Wang, and S. Li, "Structure preserving transfer learning for unsupervised hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 10, pp. 1656–1660, Oct. 2017.
[18] L. Zhang, L. Zhang, and B. Du, "Deep learning for remote sensing data: A technical tutorial on the state of the art," *IEEE Geosci. Remote Sens. Mag.*, vol. 4, no. 2, pp. 22–40, Jun. 2016.
[19] Y. Chen, X. Zhao, and X. Jia, "Spectral–spatial classification of hyperspectral data based on deep belief network," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 6, pp. 2381–2392, Jun. 2015.
[20] 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.
[21] L. Zhao, Z. Chen, Y. Yang, Z. J. Wang, and V. C. Leung, "Incomplete multi-view clustering via deep semantic mapping," *Neurocomputing*, vol. 275, pp. 1053–1062, 2018.
[22] Q. Wang, Z. Yuan, Q. Du, and X. Li, "Getnet: A general end-to-end 2-D CNN framework for hyperspectral image change detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 1, pp. 3–13, Jan. 2019.
[23] W. Zhao and S. Du, "Spectral–spatial feature extraction for hyperspectral image classification: A dimension reduction and deep learning approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 8, pp. 4544–4554, Aug. 2016.
[24] 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.
[25] B. Pan, Z. Shi, and X. Xu, "Mugnet: Deep learning for hyperspectral image classification using limited samples," *ISPRS J. Photogrammetry Remote Sens.*, vol. 145, pp. 108–119, 2018.
[26] X. Yang, Y. Ye, X. Li, R. Y. Lau, X. Zhang, and X. Huang, "Hyperspectral image classification with deep learning models," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 9, pp. 5408–5423, Sep. 2018.
[27] S. Li, W. Song, L. Fang, Y. 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.
[28] R. Hang, Q. Liu, D. Hong, and P. Ghamisi, "Cascaded recurrent neural networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 8, pp. 5384–5394, Aug. 2019.
[29] X. Li, Z. Yuan, and Q. Wang, "Unsupervised deep noise modeling for hyperspectral image change detection," *Remote Sens.*, vol. 11, no. 3, 2019, Art. no. 258.
[30] J. Nalepa, M. Myller, and M. Kawulok, "Validating hyperspectral image segmentation," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 8, pp. 1264–1268, Aug. 2019.
[31] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vision*, 2015, pp. 1520–1528.
[32] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2015, pp. 3431–3440.
[33] H. N. Mhaskar and T. Poggio, "Deep vs. shallow networks: An approximation theory perspective," *Anal. Appl.*, vol. 14, no. 6, pp. 829–848, 2016.
[34] R. Kemker, R. Luu, and C. Kanan, "Low-shot learning for the semantic segmentation of remote sensing imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 10, pp. 6214–6223, Oct. 2018.
[35] J. Nalepa, M. Myller, and M. Kawulok, "Hyperspectral data augmentation," 2019, *arXiv:1903.05580*.
[36] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*.
[37] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 1139–1147.
[38] L. Qu, C. Wu, and L. Zou, "3D dense separated convolution module for volumetric image analysis," *Applied Sciences*, vol. 10, no. 2, 2020, Art. no. 485.
[39] T. Cooijmans, N. Ballas, C. Laurent, Ç. Gülçehre, and A. Courville, "Recurrent batch normalization," 2016, *arXiv:1603.09025*.
[40] Y. Gao, X. Wang, Y. Cheng, and Z. J. Wang, "Dimensionality reduction for hyperspectral data based on class-aware tensor neighborhood graph and patch alignment," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 8, pp. 1582–1593, Aug. 2015.
[41] K. Makantasis, K. Karantzalos, A. Doulamis, and N. Doulamis, "Deep supervised learning for hyperspectral data classification through convolutional neural networks," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2015, pp. 4959–4962.
[42] E. Martel *et al.*, "Implementation of the principal component analysis onto high-performance computer facilities for hyperspectral dimensionality reduction: Results and comparisons," *Remote Sens.*, vol. 10, no. 6, 2018, Art. no. 864.
[43] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vision*, 2015, pp. 4489–4497.
[44] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vision*, 2017, pp. 2980–2988.
[45] C. Deng, Y. Xue, X. Liu, C. Li, and D. Tao, "Active transfer learning network: A unified deep joint spectral–spatial feature learning model for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 3, pp. 1741–1754, Mar. 2019.
[46] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, 2010, pp. 249–256.
[47] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proc. IEEE Int. Conf. Comput. Vision*, 2015, pp. 1026–1034.

Liang Zou (Member, IEEE) received the B.Sc. degree in microelectronics from Anhui University, Hefei, China, in 2010, the M.Sc. degree in biomedical engineering from the University of Science and Technology of China, Hefei, in 2013, and the Ph.D. degree in electrical and computer engineering from The University of British Columbia (UBC), Vancouver, BC, Canada, in 2017.

He had been a Postdoctoral Research Fellow with UBC since graduation. Between July 2017 and February 2018, he also worked as a Research Scientist with the Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly (LILY), Nanyang Technological University, Singapore. He has been an Associate Professor with China University of Mining and Technology since February 2019. His research interest includes Signal Processing and Machine Learning.

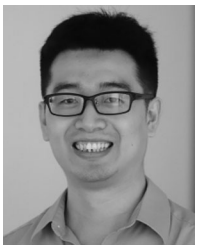
Dr. Zou has served as the organizing Co-Chair for five international conferences and reviewers for more than ten leading journals (e.g., IEEE TRANSACTIONS ON SIGNAL PROCESSING, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, IEEE JOURNAL OF BIOMEDICAL AND HEALTH INFORMATICS, IEEE SIGNAL PROCESSING LETTERS).

Xingliang Zhu received the B.Sc. degree in electronics and information engineering in 2018 from Anhui University, Hefei, China, where he is currently working toward the M.Sc. degree.

His current research interest includes machine learning and its application in remote sensing.

Changfeng Wu received the B.Sc. degree in electronics and information engineering in 2017 from Anhui University, Hefei, China, where he is currently working toward the M.Sc. degree.

His current research interest includes biomedical engineering and computer vision.



Yong Liu (Member, IEEE) received the B.S. degree from the Department of Electronic Science and Technology, University of Science and Technology of China, Hefei, China, in 2008, and the Ph.D. from the School of Computer Science and Engineering, Nanyang Technological University, Singapore, in 2016.

He is currently a Research Scientist Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly (LILY), Nanyang Technological University, Singapore. Prior to that, he was a Manager, Data Scientist at NTUC Enterprise, Singapore, from November 2017 to July 2018, and a Research Scientist at Data Analytics Department, Institute for Infocomm Research (I2R), A*STAR, Singapore, from November 2015 to October 2017. His research papers appear in leading international conferences and journals. His research areas include various topics in machine learning and data mining.

Dr. Liu has been invited as a PC Member of major conferences such as Knowledge Discovery in Databases, International Joint Conferences on Artificial Intelligence, Association for the Advancement of Artificial Intelligence, Conference on Information and Knowledge Management, International Conference on Data Mining, and reviewer for IEEE/ACM transactions. He is a member of ACM.

Lei Qu (Member, IEEE) received the Ph.D. degree in computer application techniques from Anhui University, Hefei, China, in 2008.

Between 2009 and 2011, he was a Postdoctoral Researcher with Howard Hughes Medical Institute—Janelia Farm Research Campus, Ashburn, VA, USA. He is currently a Full Professor with Anhui University. His research interests include computer vision, machine learning, and bioimage informatics.