Hyperspectral Image Classification Using Mixed Convolutions and Covariance Pooling

Jianwei Zheng, Yuchao Feng, Cong Bai, and Jinglin Zhang

***Abstract*— Recently, convolution neural network (CNN)-based hyperspectral image (HSI) classification has enjoyed high popularity due to its appealing performance. However, using 2-D or 3-D convolution in a standalone mode may be suboptimal in real applications. On the one hand, the 2-D convolution overlooks the spectral information in extracting feature maps. On the other hand, the 3-D convolution suffers from heavy computation in practice and seems to perform poorly in scenarios having analogous textures along with consecutive spectral bands. To solve these problems, we propose a mixed CNN with covariance pooling for HSI classification. Specifically, our network architecture starts with spectral–spatial 3-D convolutions that followed by a spatial 2-D convolution. Through this mixture operation, we fuse the feature maps generated by 3-D convolutions along the spectral bands for providing complementary information and reducing the dimension of channels. In addition, the covariance pooling technique is adopted to fully extract the second-order information from spectral–spatial feature maps. Motivated by the channel-wise attention mechanism, we further propose two principal component analysis (PCA)-involved strategies, channel-wise shift and channel-wise weighting, to highlight the importance of different spectral bands and recalibrate channel-wise feature response, which can effectively improve the classification accuracy and stability, especially in the case of limited sample size. To verify the effectiveness of the proposed model, we conduct classification experiments on three well-known HSI data sets, Indian Pines, University of Pavia, and Salinas Scene. The experimental results show that our proposal, although with less parameters, achieves better accuracy than other state-of-the-art methods.**

***Index Terms*— Channel-wise shift, channel-wise weighting, convolutional neural network (CNN), covariance pooling, hyperspectral image (HSI) classification, principal component analysis (PCA).**

I. Introduction

HYPERSPECTRAL image (HSI) classification, as an important part of earth observation, is widely used in fine agriculture [1]–[3], military [4]–[6], environmental monitoring [7]–[9], and other aspects. HSI can obtain spectral information from hundreds of continuous spectrum segments of surface objects. With the rapid development of remote-sensing technology, the spatial resolution has also been greatly improved, which vastly enhances the ability for HSI data sets to properly express different objects.

In HSI classification tasks, there exist three main challenges: first, the spectral dimension of hyperspectral data has hundreds of band values and the information between the spectral bands is frequently redundant, which results in high data dimension and requires a lot of computing consumption. Second, the existence of mixed pixels brings a lot of interference to the classification of HSIs, since that one pixel often corresponds to multiple object categories and is mostly easy to cause misclassification. Third, HSI samples are expensive to label manually, resulting in relatively small amount of off-the-shelf labeled samples. To remedy all these problems, many related methods have been presented in the past decade.

Early machine learning algorithms, such as support vector machine (SVM) [10], k-nearest neighbor (K-NN) [11], decision trees [12], and extreme learning machine (ELM) [13], rely only on spectral features of HSIs without considering spatial information, and often lead to unsatisfactory results. In addition, some classification methods that based on designing effective feature extraction or dimensionality reduction techniques have also been proposed, such as principal component analysis (PCA) [14], independent component analysis (ICA) [15], and linear discriminant analysis (LDA) [16]. However, the feature maps generated without incorporating spatial contextual information are not reasonable either. Hence, more and more spectral–spatial feature extraction methods [13], [17]–[22] have been proposed to improve the representation of hyperspectral data and increase the classification accuracy, for example, Markov random filed [19], sparse representation [20], metric learning [21], and composite kernels [13], [22], to name just a few.

Recently, the study of HSI classification is seeing a paradigm shift, as deep-learning-based methods push aside the traditional models. For instance, stacked autoencoder (SAE) and deep belief network (DBN) have been widely used in this field. In [23], SAE is introduced for the first time for classification of hyperspectral satellite images. In [24], an efficient stacked discriminative sparse autoencoder is developed for the land-use classification task. Li *et al.* [25] and Peng *et al.* [26] investigated the strength of DBN-based model in obtaining deep spectral feature maps, which allows unsupervised pretraining over unlabeled samples in the first step and followed by a supervised fine-tuning over labeled samples. In the meanwhile, Ping *et al.* [27] diversified the DBN model

Manuscript received November 28, 2019; revised April 10, 2020; accepted May 15, 2020. This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFE0126100, in part by the Natural Science Foundation of China under Grant 41775008, Grant 61602413, Grant 61702275, and Grant 61976192, and in part by the Zhejiang Provincial Natural Science Foundation of China under Grant LY19F030016. *(Corresponding author: Cong Bai.)*

Jianwei Zheng, Yuchao Feng, and Cong Bai are with the College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China (e-mail: zjw@zjut.edu.cn; yuchao_eason@163.com; congbai@zjut.edu.cn).

Jinglin Zhang is with the College of Atmospheric Sciences, Nanjing University of Information Science and Technology, Nanjing 210044, China (e-mail: jinglin.zhang@nuist.edu.cn).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TGRS.2020.2995575

through regularizing the pretraining and fine-tuning steps, which guarantees the optimal classification results and gains better performances when compared with the original DBNs. However, both SAEs and DBNs belong to the fully connected networks, which not only contain larger number of parameters to train, but also suffer from spatial information loss due to their requirement of one-dimensional input form.

Inspired by the intrinsic structure of the visual system, the introduction of convolution neural network (CNN) [28]–[31] has extensively promoted the development of deep learning, and the CNN-based networks have made a major breakthrough in classification accuracy. Compared with a fully connected network, CNNs take advantage of local connections and shared parameters to extract the contextual 2-D spatial features. Li *et al.* [32] fed paired samples with new labels into deep CNN to extract pixel-pair features, leading to rich discriminative spectral features. In addition, Yang *et al.* [33] applied two channels of CNN to separately extract spatial and spectral features for HIS classification. Due to the fact that HSI is a cubic data, it can be naturally considered as a 3-D tensor, thus using 3-D-kernels to extract spectral–spatial features is an intuitive scheme. For example, in [30], a multiscale 3-D deep CNN is proposed, which could jointly learn both 2-D multi-scale spatial feature and 1-D spectral feature without any hand-crafted features or pre/post-processing steps. Zhong *et al.* [31] exploited each 3-D convolutional layer following residual blocks to learn more discriminative spectral and spatial representations of HSIs separately, and used batch normalization to regularize the learning process. In terms of both efficacy and efficiency, it is well known that a shallow 3-D-CNN often performs poorly for targets having very similar textures along successive spectral bands and a deep 3-D-CNN may be computationally unaffordable.

Motivated by the aforementioned works, in this article, we tackle the HSI classification problem by taking the characteristics of mixed convolution architecture and covariance pooling into account, which leads to a well-balanced result between computational cost and classification accuracy. Moreover, we propose two PCA-involved attention mechanisms for better performance, especially in the case of limited sample size. The main contributions of our work are listed as follows.

1) We propose an end-to-end network model that embeds the consecutive 3-D–2-D convolutions and covariance pooling into the traditional CNN architecture for HSI classification. The mixed use of 2-D and 3-D convolutions can extract more discriminative spectral–spatial features, in which the 2-D operation are added to fuse the spectral feature maps obtained by 3-D convolutions and reduce the dimension of channels. In addition, covariance pooling can extract more significant second-order information from spectral–spatial features than ordinary pooling methods.
2) Motivated by channel-wise attention mechanism, we propose two schemes, channel-wise shift and channel-wise weighting, to recalibrate channel-wise feature response by emphasizing more informative spectral bands and curbing the effects of useless ones. Specifically, these two schemes are used between PCA-involved layer and the first 3-D convolution layer.

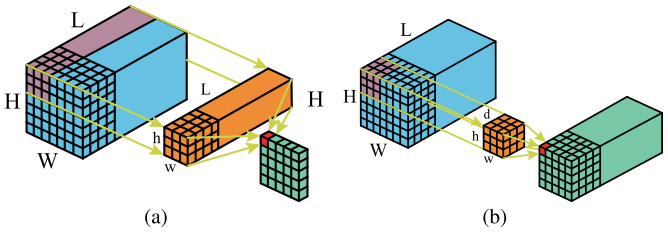


Fig. 1. Models of 2-D and 3-D convolutions. (a) 2-D output is generated by applying 2-D convolution on the HSI. (b) 3-D convolution obtains a 3-D output that remains spectral information.

The rest of this article is organized as follows. Section II gives a brief review of related work. Section III describes the details of our proposed classification framework. The comprehensive experiments are conducted in Section IV. Finally, the conclusion is drawn in Section V.

II. Related Work

A. 2-D and 3-D Convolutions

There are generally two patterns, 2-D-CNN and 3-D-CNN, for common convolution neural networks. An illustration of 2-D and 3-D convolution for hyperspectral data is shown in Fig. 1 and the formulations for 2-D and 3-D convolution are given in the following equations:

$$f_{ij}^{xy} = \Phi\left(\sum_{m}\sum_{p=1}^{h}\sum_{q=1}^{w} w_{ijm}^{pq} f_{(i-1)m}^{(x+p)(y+q)} + b_{ij}\right) \quad (1)$$

$$f_{ij}^{xyz} = \Phi\left(\sum_{m}\sum_{r=1}^{d}\sum_{p=1}^{h}\sum_{q=1}^{w} w_{ijm}^{pqr} f_{(i-1)m}^{(x+p)(y+q)(z+r)} + b_{ij}\right) \quad (2)$$

where f_{ij}^{xyz} means the output variable at position (x, y, z) in the jth feature map in the ith layer, Φ is the activation function, $(h \times w \times d)$ are the size of kernel, (p, q, r) represent the indexes of kernel, and m is the index of feature maps. Two parameters, the kernel weight w and the bias b, need to be given beforehand for training the network.

From Fig. 1, we can observe that the 2-D convolution operation focuses on extracting hyperspectral data by considering only the spatial correlation of each channel in the given image. As for the 3-D convolution operation, the correlation between different channels is also used to improve the ability of feature representation by obtaining the spectral–spatial feature maps. In other words, the 2-D convolution can extract the spatial features but fails to obtain the significant features in successive spectral bands, while the 3-D convolution is able to extract spectral–spatial features with sacrifice of more computational cost. Therefore, using 2-D convolution or 3-D convolution in a standalone mode is not the best option.

B. Feature Extraction and Pooling Strategy

In practice, various atmospheric scattering conditions and intraclass differences make the feature extraction of hyperspectral data very difficult. To solve this issue, deeper architecture

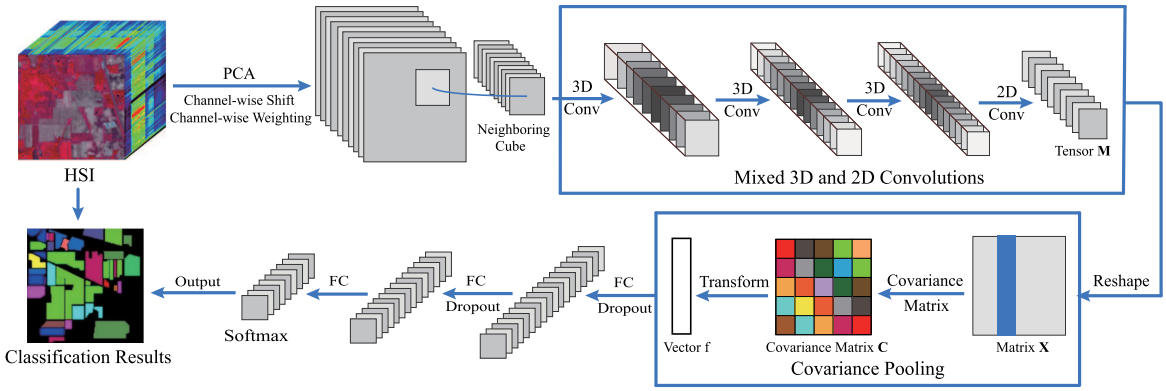


Fig. 2. Architecture of the proposed MCNN-CP Model. PCA involved channel-wise shift and channel-wise weighting are firstly used to reduce the dimension of the original HSI. Then, mixed use of 3-D and 2-D convolutions with covariance pooling is proposed for HSI classification.

is considered as a promising option since it may learn more abstract features at higher levels. However, a deep 3-D-CNN is often computationally unaffordable. Hence, the PCA algorithm is widely used for highlighting the main features of input data and also for better efficiency. In [34]–[36], before using CNN to gain the discriminative features, PCA is first adopted to reduce the dimensionality of the original data, whose experimental results verify that this scheme can improve the accuracy and efficiency of classification.

For common CNNs, typical max or average pooling strategies are always added to capture the first-order statistics. Recently, some high-order statistics were proved to achieve impressive improvement by obtaining a more significant and compact representation [37]–[39]. Lin *et al.* [40] developed a bilinear pooling network that consists of two feature extractors. The bilinear form simplifies the gradient computation and allows end-to-end training, which is particularly useful for fine-grained categorization. In addition, second-order pooling such as covariance pooling based on 2-D statistics were proposed and widely used. He *et al.* [41], [42] used this covariance matrix obtained by covariance pooling to generate several stacked features from different convolutional layers. Each element of this matrix stands for the covariance between two contributed feature maps and further provides more complementary information.

C. Effective Receptive Fields

It is well known that receptive fields are very important for convolutional neural networks, especially in object detection and instance segmentation [43], [44]. Luo *et al.* [45] proved that during the feature extraction of 2-D or higher dimensional data, only a small proportion of pixels in the theoretical receptive fields contribute significantly to an output unit's response, and thus the concept of effective receptive fields was proposed.

In many cases, the effective receptive fields always show a Gaussian distribution. For example, in 2-D space, the effective receptive fields of each convolutional layer are a small area outward from the center point, occupying only a fraction of the theoretical receptive fields. Moreover, its effect on the output appears to rapidly decay from center to edge. By visualizing the effective receptive fields and calculating the gradient of the output units with respect to the sampling locations, Zhu *et al.* [44] proved that during forward propagation, the pixels closer to the center of the receptive field are convolved with more times, which deliver information through many different paths to the output. On the contrary, the pixels near the edge may be convolved with fewer times, which have very few paths to propagate their impact. In the process of back propagation, since the gradients from an output unit are propagated across all paths, the center pixels will obtain larger gradients than the edge points. To remedy these issues, padding with zeros is an effective method to assure that all pixels can be treated equally.

However, in the process of tensor data, due to the limitation of the great amount of parameters and the high computational cost, the padding with zeros operation is seldom used for promoting equal attention to each pixel. Instead, we focus more on discarding the dross and selecting the essence. For example, in [46], a squeeze-and-excitation block is built to highlight the importance of informative channels and suppress channels with little information, which recalibrates feature maps and improves the final categorization accuracy. Therefore, how to use the characteristics of the effective receptive fields becomes very important to obtain more valid information during the convolution process.

III. Methodology

In this section, we present our classification framework termed as mixed use of 3-D and 2-D CNN with covariance pooling (MCNN-CP). As shown in Fig. 2, our method mainly consists of four steps. For any given data, we first use PCA to remove the spectral redundancy along spectral bands and apply one of the two operations, channel-wise shift and channel-wise weighting, to highlight the spectral bands containing more information. Then, for each neighboring cube, we exploit mixed use of 3-D and 2-D convolution to extract spectral–spatial features followed by reshaping tensor to matrix form along with the spectral bands. Consequently, a covariance pooling layer is appended to aggregate the obtained spectral–spatial feature maps. Finally, the output is

sent to three fully connected layers to derive the classification result.

A. PCA and Its Ranking Essence

Similar to [21], [47], and [48], we adopt the PCA step in our framework. The role of PCA lies in reducing the dimension along the spectral bands and holding the spatial information intact. Specifically, PCA obtains a more compact matrix by first forming the covariance matrix B of the input data X and then selecting k eigenvectors corresponding to the largest k eigenvalues, and the feature vectors are used as the column vectors to obtain the optimal projection matrix P. For the input HSI data with size $W \times H \times L$, we should first reshape it into $X \in R^{L\times N}$ with $N = W \times H$ and then introduce the cost function of PCA as follows:

$$\min_{P} \operatorname{tr}\left(P^T B P\right), \quad \text{s.t. } P^T P = I, B = \frac{1}{L} X X^T \tag{3}$$

where I is an identity matrix with appropriate size, tr and T represent the trace and transpose operations, respectively. When $P \in R^{L\times k}$ is achieved, the output of PCA would be $Y = P^T \times X \in R^{k\times N}$, whose spectral dimension is reduced from the original L to k.

As stated previously, PCA pursues to maintain the most intrinsic information with reduced dimensional space. In fact, it measures the importance of each direction by comparing the magnitude of the data variance in the projection space. As it is well known, the larger the data variance, the greater the amount of information contained. Hence, we can assert that the carried information of spectral bands is in descending order after the PCA step, in which the more informative bands would contribute more to the subsequent feature extraction process. That is to say, we can determine the relatively more important spectral bands by relying on the characteristics of ranking essence of PCA.

B. 3-D–2-D Convolutions and Their Margin Effect

In practice, a shallow 3-D-CNN is hardly to achieve satisfactory spectral–spatial features and a deep 3-D-CNN is often computationally unaffordable. Hence, it is necessary to study certain sound ways for a well balance between these two operations.

To extract more expressive and discriminative feature maps, we propose a mixed use of 3-D and 2-D convolutions with small kernels. As shown in Fig. 2, we first employ three 3-D convolution layers to gain the spectral–spatial feature maps. The 3-D convolution kernels are set to $8 \times 3 \times 3 \times 7$, $16 \times 3 \times 3 \times 5$, and $32 \times 3 \times 3 \times 3$, respectively, where $8 \times 3 \times 3 \times 7$ means eight 3-D convolution kernels with dimension $3 \times 3 \times 7$. Then, we reshape the output (four dimensions tensor) into three dimensions by concatenating the third and fourth dimensions. Finally, we use one 2-D convolution layer with a 3×3 convolution kernel to fuse spectral bands and reduce the number of spectral dimension. As is known to all, the feature maps generated by 3-D convolution have different characteristics, so the fusion by 2-D convolution operation would help improve the accuracy of HSIs classification by involving more complementary information. Evidently, the mixed use of 3-D and 2-D convolution can fully utilize both the spectral as well as spatial feature information to obtain more discriminative features.

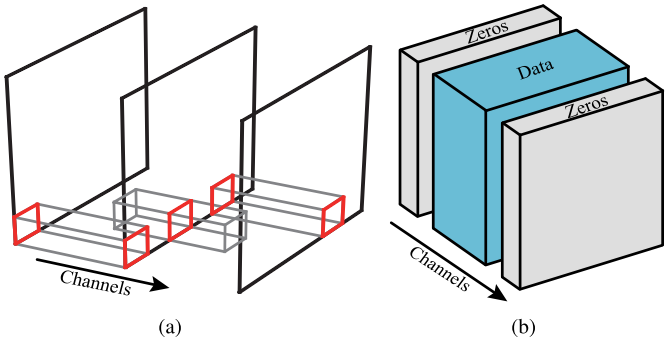


Fig. 3. (a) Model of 3-D convolution along the spectral dimension. The black square frames stand for the spectral bands, the gray frames represent the filters, and the red frames are the filters across the spectral bands. (b) Scheme of padding channels with zeros.

It is noted that in the traditional 2-D convolution process, there are edge effects where the intermediate pixels play a more considerable role than the edge pixels, and this problem also exists on the 3-D convolution along channels. Fig. 3(a) shows the typical 3-D convolution process, we can observe that for the most left and right marginal spectral bands, less 2-D convolution operations will be covered when only one 3-D-kernel is moving along their surface. Whereas, the relatively middle spectral bands would receive as many 2-D convolutions as the size of 3-D-kernel's third dimension. Concretely, the first spectral band has only one 2-D-kernel to extract the spatial feature in 3-D convolution, the second has two 2-D-kernels, the third has three 2-D-kernels, and so on, until the dth spectral band that will involve the maximum kernels. It is noted that in the traditional 2-D convolution process, for the sake of keeping the output image size unchanged while retaining more useful information, we usually pad the input image with zeros enlarging its all-round dimensions. Similarly, the third dimension of HSI data can also be enlarged during 3-D convolution, as shown in Fig. 3(b), so that more 2-D operations can be performed and each band can be situated in effective receptive field to preserve more information along the spectral bands. For this margin effect of 3-D convolution, the informative channels should be migrated to the middle of effective receptive fields.

C. Channel-Wise Shift and Channel-Wise Weighting

Based on the ranking essence of PCA and the margin effect of 3-D convolution, we propose two schemes, namely channel-wise shift and channel-wise weighting, which are used between the PCA step and the first 3-D convolution layer to enhance feature extraction capabilities and improve classification accuracy by emphasizing more informative spectral bands and suppressing less useful ones.

The channel-wise shift scheme is illustrated in Fig. 4, our core idea is to move the relatively more important spectral bands to the more central position for the most sufficient

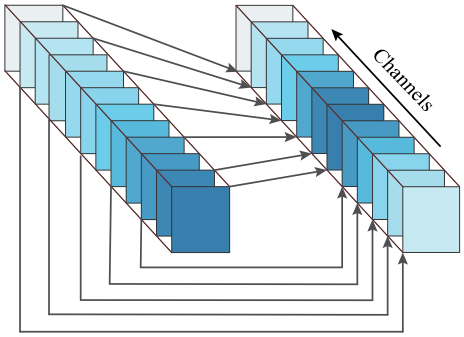


Fig. 4. Scheme of channel-wise shift. The deeper color denotes the more information in the spectral band.

3-D convolutions. To the contrary, the relatively less important spectral bands should be placed on the marginal place for slightly information retrieval and computational saving. This strategy can increase the number of spatial feature extraction times for informative channels and keep them in the center of the effective receptive fields. By this operation, we can assure that the important spectral bands would be kept in the middle of all the channels to involve more 3-D convolution operations.

The channel-wise weighting scheme is to penalize a predefined weight, namely 1 + Ratio, for each channel, where Ratio represents the proportion of the variance value of each principal component to the total variance value. Compared with the channel-wise attention mechanism, this scheme will not increase any cost since that the Ratio is directly generated by the PCA operation. It is intuitive that the larger the weight is, the more important the corresponding component would be, which can further reveal more spectral information in that band. It is noted that directly weighting a ratio between 0 and 1 results in shrunken features, thus we simply use 1 + Ratio to recalibrate channel-wise feature responses. Although other functions, for example, Sigmoid Function, may make the ratio smoother, our weight would avoid the feature shrink but still highlight the difference. Through back propagation, the informative channels will obtain a larger gradient amplitude than others. It should be also noted that the amplified difference would probably destroy the original correlation between different spectral bands. Therefore, the channel-wise weighting scheme is recommended only when the number of samples is relatively small.

D. Covariance Pooling

As shown in Fig. 2, after all the feature extraction steps, the output is a tensor $M \in \mathbb{R}^{H\times W\times D}$. We first reshape M into matrix $Z \in \mathbb{R}^{D\times N}$ with $N = H \times W$. Then, we can formulate the covariance matrix as follows:

$$C = \frac{1}{N-1}\sum_{i=1}^{N}(z_i - \overline{z})(z_i - \overline{z})^T \in \mathbb{R}^{D\times D} \tag{4}$$

where $z_i \in [z_1, z_2, \ldots, z_N]$ is the ith column vector of Z along the second dimension and $\overline{z} = (1/N)\sum_{i=1}^{N} z_i$.

Equation (4) is the second-order pooling for the stacked matrix X. Following [49]–[51], we can achieve three distinctive advantages by adopting covariance pooling to our network framework. First, each off-diagonal entry of C can make full use of the channel relationship to fuse complementary information coming from different feature maps. Second, we use the average operation in the process of covariance computation, which will make it better to filter the noise of a single sample. Third, since calculating the covariance between all feature maps is independent of the order of the feature maps, it is assumed that the covariance pooling is robust to rotation. However, we can also know from the formula that the computational load of the covariance pooling method is proportional to the scale of the input. Therefore, to ensure that the method is usable and significant, the appropriate input data size should be set. This is another motivation for adding a 2-D convolution layer to reduce the dimension of the feature maps.

In implementation, it has been proven in [52] that the original covariance matrix lies on the Riemannian manifold space, which may not suit for the subsequent Euclidean classification. Fortunately, with a simple logarithm operation, the covariance matrix would be transformed into the Euclidean space without losing any geometric relationships. Specifically, the pooled feature F can be formulated as

$$F = U\log(\Sigma)U^T \in \mathbb{R}^{D\times D} \tag{5}$$

where $C = U\Sigma U^T$, U and Σ are the eigenvector matrix and eigenvalue matrix of C, respectively. Due to the fact that matrix F is a symmetric matrix, we only need to vectorize the upper triangle matrix of F for the final vector f with dimension $D(D+1)/2$.

IV. Experiments

A. Data Description

To verify the practical performance of our MCNN-CP method, we conduct several experiments on three representative databases, including Indian Pines (IP), University of Pavia (UP), and Salinas Scene.

1) Indian Pines: The first data set is gathered by the AVIRIS in 1992 from Northwest Indian, containing 16 classes and having 145 × 145 spatial dimension with a resolution of 20 m by pixels. The original data set contains 224 spectral bands with the wavelength ranging from 400 to 2500 nm. The size of training and test samples for each subclass is shown in Table I.

2) University of Pavia: The second data set is collected by the ROSIS sensor in 2002 from Northern Italy, including nine urban land-cover types and having 610 × 340 spatial dimension with a resolution of 1.3 m by pixels. The original data set contains 103 spectral bands with the wavelength ranging from 430 to 860 nm. The size of training and test samples for each subclass is listed in Table II.

3) Salinas Scene (SA): The third data set is acquired by the AVIRIS sensor over Salinas Valley, California, and is characterized by 3.7-m pixels spatial resolution. The area covered comprises 512 lines by 217 samples and contains 16 classes. The original data set contains 224 spectral bands with the wavelength ranging from 360 to 2500 nm. The size of training and test samples for each subclass is shown in Table III.

TABLE I
SIZE OF TRAINING AND TEST SAMPLES FOR EACH SUBCLASS IN THE IP DATA

No.	Class Name	Training	Test
1	Alfalfa	14	32
2	Corn-no till	428	1000
3	Corn-min till	249	581
4	Corn	71	166
5	Grass-pasture	145	338
6	Grass-trees	219	511
7	Grass-pasture-mowed	8	20
8	Hay-windrowed	143	335
9	Oats	6	14
10	Soybean-no till	292	680
11	Soybean-min till	736	1719
12	Soybean-clean	178	415
13	Wheat	62	143
14	Woods	379	886
15	Buildings-Grass-Trees-Drives	116	270
16	Stone-Steel-Towers	28	65
	Total	3074	7175

TABLE II
SIZE OF TRAINING AND TEST SAMPLES FOR EACH SUBCLASS IN THE UP DATA

No.	Class Name	Training	Test
1	Asphalt	1989	4642
2	Meadows	428	13055
3	Gravel	630	1469
4	Trees	919	2145
5	Painted metal sheets	403	942
6	Bare Soil	1509	3520
7	Bitumen	399	931
8	Self-Blocking Bricks	1105	2577
9	Shadows	284	663
	Total	12832	29944

TABLE III
SIZE OF TRAINING AND TEST SAMPLES FOR EACH SUBCLASS IN THE SALINAS SCENE DATA

No.	Class Name	Training	Test
1	Brocoli_green_weeds_1	603	1406
2	Brocoli_green_weeds_2	1118	2608
3	Fallow	593	1383
4	Fallow_rough_plow	418	976
5	Fallow_smooth	803	1875
6	Stubble	1188	2771
7	Celery	1074	2505
8	Grapes_untrained	3381	7890
9	Soil_vinyard_develop	1861	4342
10	Corn_senesced_green_weeds	983	2295
11	Lettuce_romaine_4wk	320	748
12	Lettuce_romaine_5wk	578	1349
13	Lettuce_romaine_6wk	275	641
14	Lettuce_romaine_7wk	321	749
15	Vinyard_untrained	2180	5088
16	Vinyard_vertical_trellis	542	1265
	Total	16238	37891

The three selected data sets have their own characteristics: IP has a large spectral length with a small spatial size, and UP has a large spatial size with a small spectral length. For SA, both of these two sizes are large. Such a choice covers three different application scenarios and would be beneficial to the verification of our model.

TABLE IV
OA (%) OF DIFFERENT SPATIAL SIZES OF THE HSI SUBCUBE UNDER 5% AND 30% TRAINING DATA

Spatial Size	5%			30%		
	IP	UP	SA	IP	UP	SA
21×21	95.95	98.90	99.77	99.61	**99.95**	99.98
23×23	96.10	98.87	99.70	99.62	99.92	99.95
25×25	96.29	**98.98**	**99.86**	**99.69**	99.93	**100**
27×27	95.97	98.74	99.78	99.55	99.56	99.99
29×29	**96.79**	98.56	99.79	99.68	99.71	99.98

B. Experimental Setup

To demonstrate the efficacy and the efficiency of our proposed classification model, we choose several widely used supervised methods as the competitors, including SVM [10], 2-D-CNN [34], 3-D-CNN [53], M3D-CNN [30], and SSRN [31]. For SVM, the penalty parameter and the spread of the Gaussian kernel are, respectively, chosen from two candidate sets $\{10^{-1}, 10^{-2}, 10^{-3}\}$ and $\{1, 10, 10^2, 10^3\}$ using a Grid Search method.

For the sake of fairness, we choose to use the same spatial sizes of 25×25 pixels as inputs. The batch size, the learning rate, and the number of training epochs are set to 256, 0.001, and 100, respectively. For the last three layers of the fully connected layers in our model, we use 256, 128, and 16 (categories) neural units, respectively. Moreover, we set the dropout of the first and second layers to 0.4. All of the experiments are conducted on a personal laptop with Intel i7-9750H 2.6-GHz processor, 16-GB RAM, and an NVIDIA GTX1650 graphic card. The used coding tool is Python 3.6 with Keras-2.2.4.

To reduce the influence of random initialization, we repeatedly run all the algorithms five times and then compute the average results for the final report. Besides, to evaluate the performance of different classification methods, the three well-known numerical indexes, overall accuracy (OA), average accuracy (AA), and Kappa Coefficient (Kappa), are adopted to assess the classification results. OA represents the ratio between the number of correctly classified samples to the total test samples, AA represents the average of accuracies in all classes, and Kappa is an available measure of agreement between the ground truth map and classification map.

C. Parameter Analysis

In the proposed method, there are three important hyperparameters: the spatial size of each neighboring cube, the reduced spectral dimension after PCA, and the dropout proportions of the fully connected layers. To demonstrate their individual role to the final accuracy, we perform HSI classification with regarding to the variation of each hyperparameter in this section.

First, by fixing the reduced spectral length and the dropout ratio, we traverse the spatial size of input sub-cubes from five candidate values $\{21 \times 21, 23 \times 23, 25 \times 25, 27 \times 27$, and $29 \times 29\}$ to run our method. For each data set, we randomly select 5% and 30% of data from each class as the training set, and take the remaining samples as test groups. Table IV show the OA results of the proposed method on three HSIs, we can

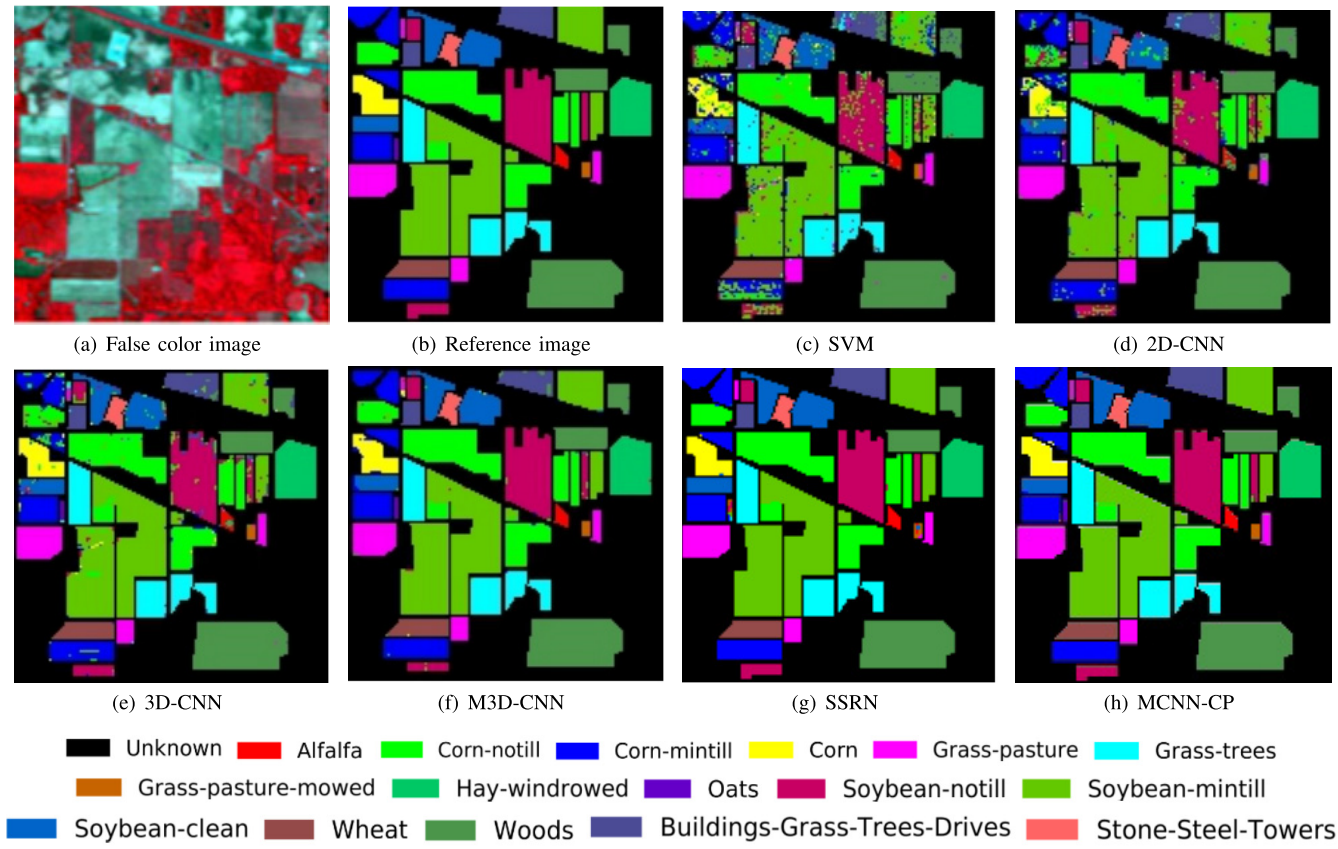


Fig. 5. Classification maps of the IP data using different models. (a) False color image. (b) Reference image. (c) SVM. (d) 2-D-CNN. (e) 3-D-CNN. (f) M3D-CNN. (g) SSRN. (h) MCNN-CP.

observe that the proposed method achieves the best OA when the spatial size is set to be 25×25 in large probability.

Second, to determine the best subspace dimension in PCA under the condition of fixed spatial size and dropout proportion, we search for different k from a given set $\{15, 20, 25, 30, 35, L\}$, where L means that the corresponding experiments are conducted without using PCA. Table V lists the OA results versus different spectral dimensions on three data sets. From this table, we can see that among all the subspace spectral bands, spectral length 35 is more suitable for the IP and SA data sets, and spectral length 20 is more appropriate for the UP data set. It is noted that the results of our method without using PCA only occupy one best value in all the cells, which verifies that PCA can reduce the redundancy of spectral bands and obtain better performance on average.

Third, we fix the other two hyperparameters and investigate different dropout proportions from $\{0, 0.2, 0.3, 0.4, 0.5\}$ to check the capacity of information capture. Dropout is used to solve the overfitting phenomenon caused by insufficient samples. The experimental results are listed in Table VI. Taking into account all the classification results from different data sets, we finally set the ratio to be 0.4 as the final parameter.

D. Performance Comparison

In this section, we use four numerical indexes to report the quantitative results acquired by all the competing methods on different data sets, including class-wise accuracy, OA, AA, and Kappa. Table VII shows the accuracy on the IP data and Table VIII are the results on the Salinas Scene data, while Table IX reports the results on the UP data. From these tables, we can observe that SVM achieves the lowest accuracy and our proposed model occupies the first place among all the competing approaches. From Table VIII, we can see that 2-D-CNN has better classification results than 3-D-CNN in the Salinas Scene data. The reason is attributed to the redundancy among many sub-classes, for example, Grapes-untrained and Vinyard-untrained. In other words, the texture of these sub-classes are very similar over most spectral bands. In addition, the performance of SSRN is always higher than M3D-CNN. The superiority of our model lies in the addition of a 2-D convolution layer after all the three 3-D convolution layers, which can better extract spectral–spatial features from the HIS data sets.

Figs. 5–7 illustrate the classification maps using SVM, 2-D-CNN, 3-D-CNN, M3D-CNN, SSRN, and MCNN-CP. The visual results in these figures are consistent with the numerical values listed in Tables VII–IX. Compared with the reference image, we can draw the conclusion that the quality of classification map of SSRN and MCNN-CP is far better than other methods. Furthermore, we can easily find that the average of class-wise classification accuracy of our model is higher than SSRN. By carefully comparing our method with SSRN, we can observer that more incorrectly classified pixels exist in the SSRN generated results. Benefiting from

TABLE V

OA (%) OF DIFFERENT SUBSPACE SPECTRAL DIMENSIONS UNDER 5% AND 30% TRAINING DATA

Spectral Size		5%			30%		
		IP	UP	SA	IP	UP	SA
With PCA (k)	15	95.90	98.72	99.75	99.60	99.87	99.99
	20	95.49	**98.98**	99.75	99.52	**99.93**	99.99
	25	95.45	98.60	99.77	99.58	99.85	**100**
	30	95.92	98.74	99.81	99.64	99.90	**100**
	35	**96.29**	98.67	**99.86**	99.69	99.92	**100**
Without PCA	L	94.21	97.85	99.52	**99.78**	99.87	99.98

TABLE VI

OA (%) OF DIFFERENT DROPOUT RATIOS UNDER 5% AND 30% TRAINING DATA

Dropout	5%			30%		
	IP	UP	SA	IP	UP	SA
without	95.32	98.42	99.58	99.62	99.90	99.98
0.2	95.59	98.69	99.83	99.65	99.88	**100**
0.3	95.77	98.64	**99.87**	99.65	99.92	99.99
0.4	96.29	**98.98**	99.86	**99.68**	**99.94**	**100**
0.5	**96.76**	98.78	99.69	99.57	99.88	99.99

TABLE VII

CLASSIFICATION RESULTS (%) OF DIFFERENT MODELS ON THE IP DATA

Class.	SVM	2D-CNN	3D-CNN	M3D-CNN	SSRN	MCNN-CP
1	82.20	75.00	79.23	97.03	97.82	100
2	73.82	81.40	88.60	97.90	99.17	99.45
3	82.15	87.60	85.81	92.41	99.53	99.88
4	77.12	62.04	90.53	93.25	97.79	100
5	73.66	92.30	96.11	95.00	99.24	100
6	93.40	99.21	98.43	99.74	99.51	100
7	96.21	75.00	92.36	100	98.70	100
8	85.72	100	98.51	99.99	99.85	100
9	97.38	64.28	88.90	96.61	98.50	100
10	71.01	82.79	87.72	96.32	98.74	100
11	76.50	91.27	91.42	97.13	99.30	99.45
12	83.90	82.89	90.04	97.16	98.43	99.94
13	83.56	99.30	99.00	99.60	100	98.62
14	98.63	98.87	97.95	98.42	99.31	100
15	94.21	86.29	82.57	83.31	99.20	100
16	69.63	100	98.51	100	97.82	100
OA	85.29 ±2.82	89.49 ±0.16	91.09 ±0.41	95.33 ±0.12	99.18 ±0.25	**99.72 ±0.18**
AA	79.03 ±2.66	86.13 ±0.81	91.59 ±0.16	96.40 ±0.73	98.93 ±0.59	**99.81 ±0.54**
Kappa	83.11 ±3.16	87.95 ±0.50	89.98 ±0.50	94.71 ±0.21	99.06 ±0.29	**99.68 ±0.13**

TABLE VIII

CLASSIFICATION RESULTS (%) OF DIFFERENT MODELS ON THE SALINAS SCENE DATA

Class.	SVM	2D-CNN	3D-CNN	M3D-CNN	SSRN	MCNN-CP
1	99.60	100	98.41	97.50	100	100
2	99.82	99.96	100	100	100	100
3	99.26	99.63	99.23	99.43	100	100
4	99.40	99.28	99.90	99.51	99.89	100
5	99.42	99.20	99.43	99.72	100	100
6	100	100	99.55	99.23	100	100
7	99.83	100	99.72	99.45	100	100
8	85.25	93.62	89.75	82.63	100	100
9	99.71	100	99.81	99.70	100	100
10	97.03	98.82	98.36	97.31	99.91	100
11	98.24	99.73	98.12	98.05	100	100
12	99.46	100	98.96	98.50	100	100
13	98.77	100	98.93	98.70	100	100
14	97.30	99.86	98.60	98.42	100	100
15	92.71	91.52	79.31	87.18	99.96	100
16	99.41	99.92	94.51	91.11	100	100
OA	92.94 ±0.33	97.39 ±0.02	93.95 ±0.15	94.79 ±0.30	99.97 ±0.05	**100 ±0.01**
AA	94.61 ±2.29	98.85 ±0.07	97.02 ±0.64	96.26 ±0.56	99.97 ±0.04	**100 ±0.02**
Kappa	92.12 ±0.19	97.07 ±0.11	93.31 ±0.51	94.22 ±0.22	99.97 ±0.06	**100 ±0.01**

TABLE IX

CLASSIFICATION RESULTS (%) OF DIFFERENT MODELS ON THE UP DATA

Class.	SVM	2D-CNN	3D-CNN	M3D-CNN	SSRN	MCNN-CP
1	94.72	98.51	98.40	98.31	100	100
2	97.15	99.54	96.91	96.10	99.87	100
3	82.73	84.62	97.05	96.34	100	100
4	96.82	98.04	98.84	98.82	100	99.51
5	99.71	100	100	99.97	100	99.89
6	90.48	97.10	99.32	99.83	100	100
7	87.73	95.05	98.92	99.66	100	100
8	88.29	96.39	98.33	99.23	99.34	99.86
9	99.90	99.69	99.90	99.92	100	99.95
OA	94.33 ±0.19	97.84 ±0.21	96.52 ±0.08	95.77 ±0.21	99.89 ±0.02	**99.94 ±0.03**
AA	92.97 ±0.42	96.56 ±0.03	97.47 ±1.32	95.09 ±1.31	99.89 ±0.05	**99.92 ±0.12**
Kappa	92.51 ±0.71	97.19 ±0.51	95.50 ±0.22	94.51 ±0.14	99.87 ±0.02	**99.92 ±0.03**

TABLE X

COMPARISON OF PADDING CHANNELS WITH ZEROS(A) AND CHANNEL-WISE SHIFT(B) ON 30% TRAINING SAMPLES OF THE IP

Data	Without A and B	With A	With B	With A and B
OA(%)	99.46	99.54	99.72	99.64
AA(%)	99.40	99.69	99.81	99.72
Kappa(%)	99.38	99.47	99.68	99.59
Train(m)	11.6	14.5	11.7	14.4
Test(s)	5.8	7.5	5.8	7.1
Parameters	1,011,584	1,122,176	1,011,584	1,122,176

the introduction of the covariance pooling strategy, our model does not cause such problems and performs better than SSRN.

For all the competitors, Fig. 8 further shows their performance along with the increasing number of training samples. We can see that our model again outperforms the others in all these cases. SSRN ranks second among all these competing methods, which achieves comparable results of OA and Kappa in the case of sufficient training samples. However, its AA is very low, especially when the number of training samples is small. In contrast, our model can achieve very high AA results while holding slightly better OA and Kappa values than SSRN, which demonstrates that the AA of our MCNN-CP in all classes is surely higher.

E. Component Analysis

To further investigate the underlying contributions of different tricks used in our model, we demonstrate their individual roles in this section. The experimental results are shown in Fig. 9. From this figure, we can see that different combinations of the proposed mechanisms, covariance pooling, channel-wise shift, and channel-wise weighting, can all improve the accuracy to some extent, which empirically support our theoretical analysis. When the covariance pooling is absent, using channel-wise shift and channel-wise weighting individually can lead to slightly higher classification accuracy, and combing these two schemes together can get even better results. When the covariance pooling layer is added, channel-wise shift can still improve the accuracy of classification, while channel-wise weighting does not perform well.

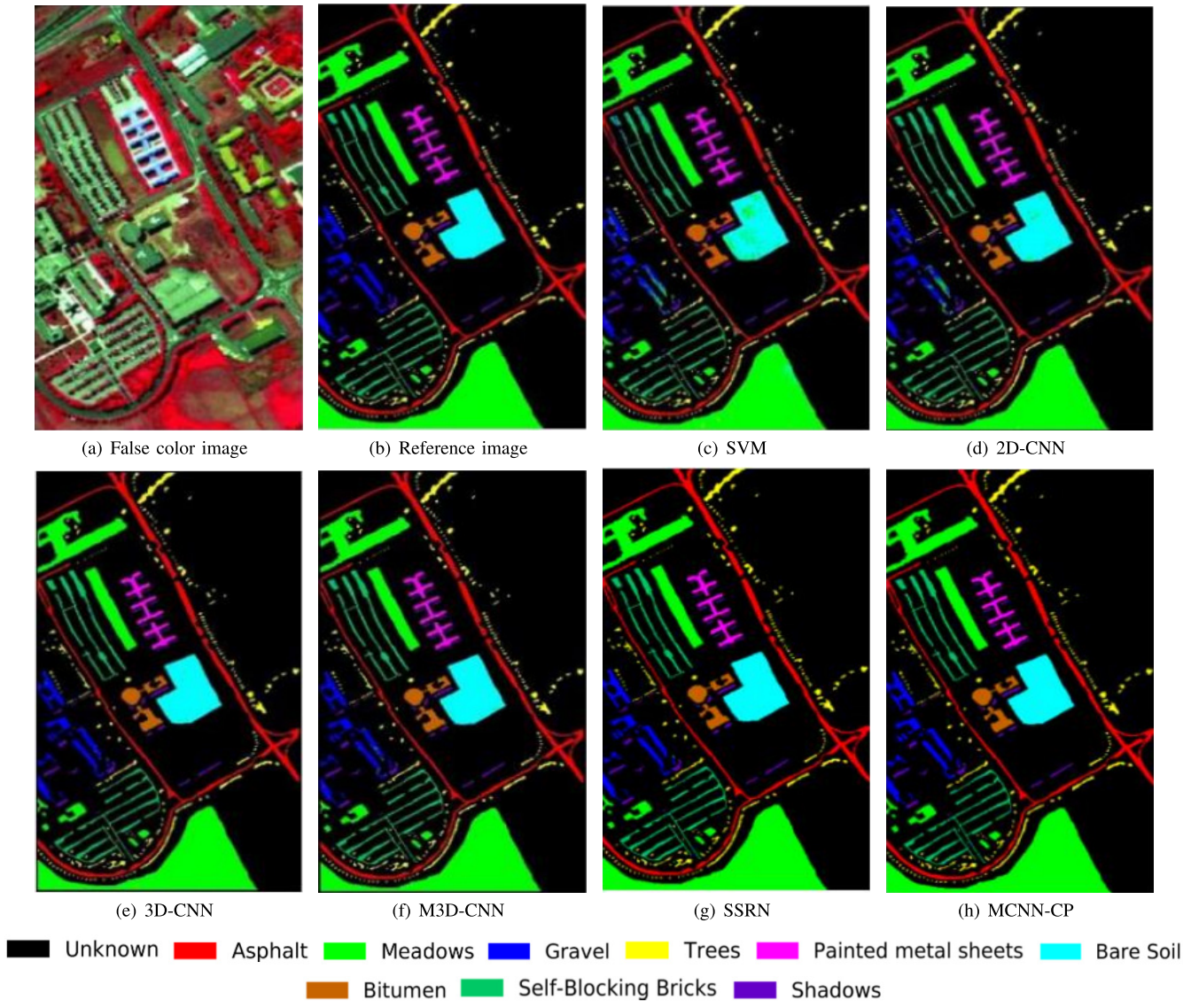


Fig. 6. Classification maps of the UP data using different models. (a) False color image. (b) Reference image. (c) SVM. (d) 2-D-CNN. (e) 3-D-CNN. (f) M3D-CNN. (g) SSRN. (h) MCNN-CP.

We discuss the reason for this result as follows. Although both the schemes of channel-wise shift and channel-wise weighting emphasize the importance of different spectral bands and try to reveal more structural information, excessive variance between weights will reversely hinder our model from learning the trustful spectral–spatial features. In general, in practical applications, once we use the covariance pooling, it is suggested that the mechanism of channel-wise weighting should not be adopted. However, we can still find from Fig. 9 that in the case of 1% training samples, using channel-wise weighting and covariance pooling together can effectively emphasize the information of the main spectral bands and improve the classification accuracy. Hence, we can still use channel-wise weighting together with the covariance pooling when the training samples are very small.

Recall that one can achieve the similar function by bilaterally padding the spectrum channel with zeros. To directly compare this scheme with our channel-wise shift, Table X shows the results using our method with or without these two schemes. From this table, our first observation is that channel-wise shift achieves better classification accuracy over the other options. Moreover, our scheme increases few training and running time compared with the very original method. Encouragingly, all these results are achieved without requirement of any additional parameters. We can also observe that when using the two schemes together, the classification accuracy ranks between their individual results. From this, we deduce that compared with the method of expanding channels to obtain a fair number of convolution times for each spectral band, our proposed channel-wise shift will emphasize informative spectral bands and suppress useless ones, so as to obtain more discriminative features. In summary, our strategy is simpler, faster and also more accurate.

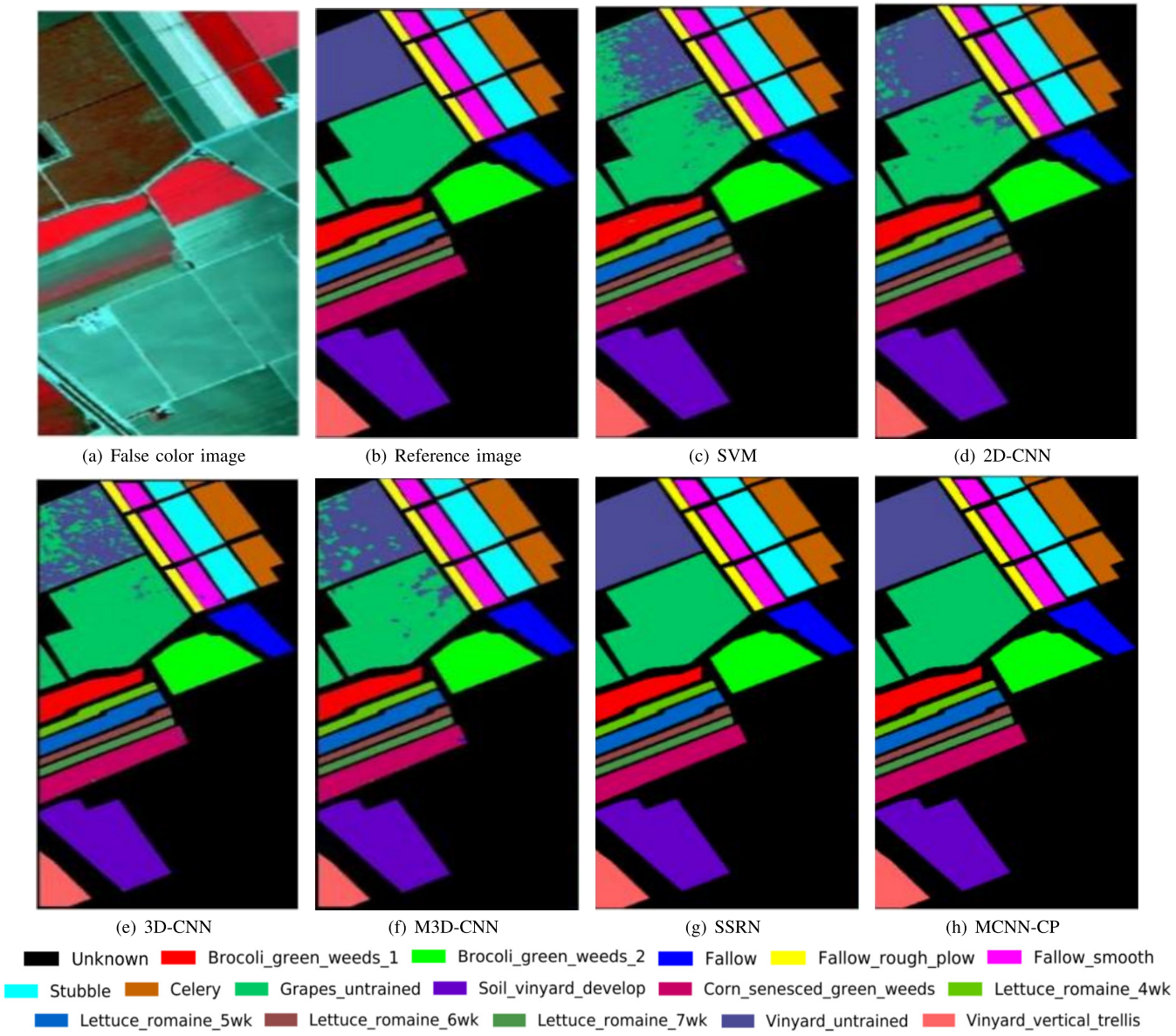


Fig. 7. Classification maps of the Salinas Scene data using different models. (a) False color image. (b) Reference image. (c) SVM. (d) 2-D-CNN. (e) 3-D-CNN. (f) M3D-CNN. (g) SSRN. (h) MCNN-CP.

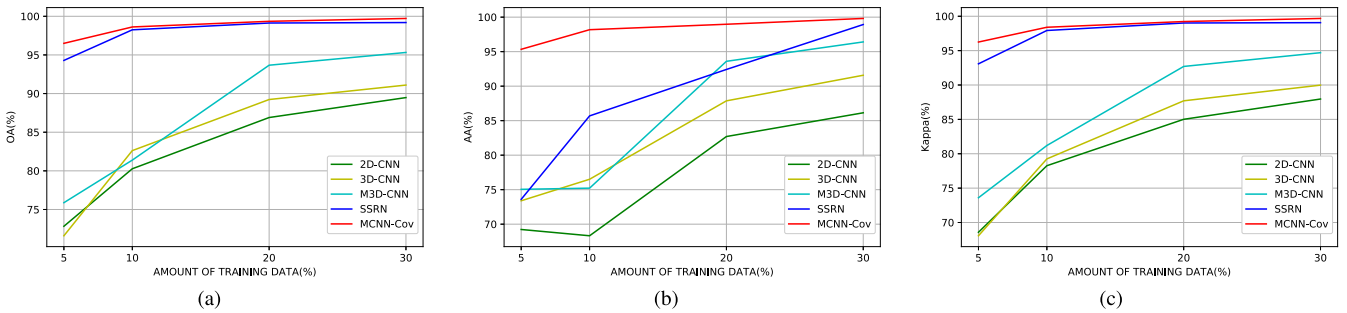


Fig. 8. Classification results (%) using different models on the IP data under different amount of training size. (a) OA. (b) AA. (c) Kappa.

Fig. 10 further shows the performance of our method with or without using the mixed convolution mechanism. From the results of using only 2-D or 3-D convolution layers, the mixed usage of 2-D-3-D operation achieves a well balance between excellent classification results and fast running time. This result verifies that the introduction of 2-D convolution

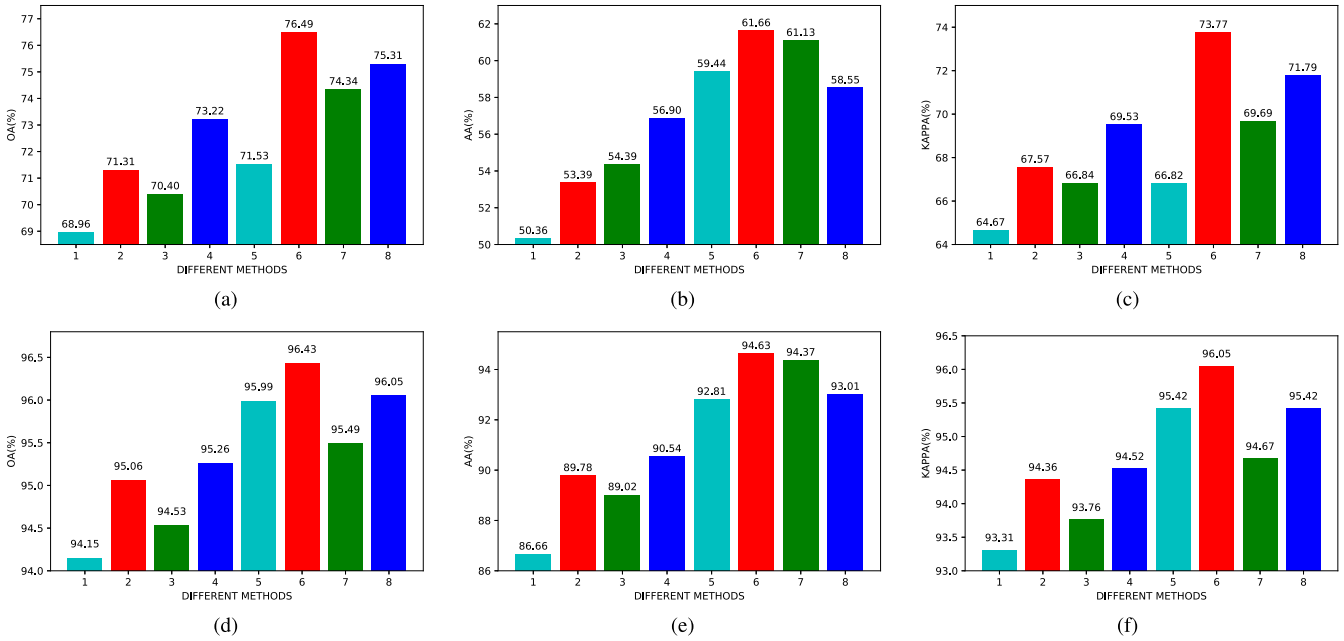


Fig. 9. Classification results (%) using different methods on the IP data. Among all the methods, (5)–(8) adopt the covariance pooling scheme, while (1)–(4) not. (2) and (6) adopt the channel-wise shift scheme. (3) and (7) adopt the channel-wise weighting scheme. (4) and (8) adopt both the channel-wise shift and weighting schemes. (a) OA with 1% traing data. (b) AA with 1% traing data. (c) Kappa with 1% traing data. (d) OA with 5% traing data. (e) AA with 5% traing data. (f) Kappa with 5% traing data.

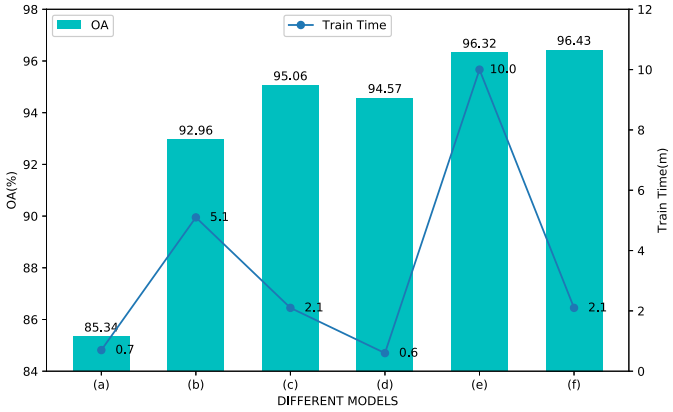


Fig. 10. Results of OA(%) and training time(s) using different models on 5% training samples of the IP. (d)–(f) adopt the covariance pooling scheme, while (a)–(c) not. (a) and (d) adopt four 2-D convolution layers. (b) and (e) adopt three 3-D convolution layers. (c) and (f) adopt three 3-D and one 2-D convolutions.

TABLE XI
OUTPUT FORM AND PARAMETERS OF THE PROPOSED MODEL ON THE IP DATA

Layer	Output Shape	Parameters
Input Data	(145,145,200,1)	-
Preprocessing Layer	(25,25,35,1)	0
3D Convolution (8,3,3,7)	(23,23,29,8)	512
3D Convolution (16,3,3,5)	(21,21,25,16)	5776
3D Convolution (32,3,3,3)	(19,19,23,32)	13856
Reshape Layer	(19,19,736)	0
2D Convolution (64,3,3)	(17,17,64)	424000
Reshape Layer	(289,64)	0
Covariance Pooling	(64,64)	0
Feature Vector	(2080)	0
Fully Connected Layer	(256)	532480
Fully Connected Layer	(128)	32896
Fully Connected Layer	(16)	2064
Total Trainable Parameters With PCA: (Spectral Size 35)		1,011,584
Total Trainable Parameters Without PCA: (Spectral Size 200)		4,052,864

can not only effectively fuse the numerous spectral bands generated by 3-D convolutions, but also reduce the dimension of channels and release many occupied resources for speed-up. From Fig. 10, we also can conclude that covariance pooling is beneficial for overcoming the defect of overlooking spectral information in a full 2-D convolutional network, thereby significantly improving model performance.

From Figs. 8–10, we can summarize the following three conclusions: first, adding a 2-D convolution layer after three 3-D convolution layers not only reduces the spectral dimension for computational saving, but also extracts discriminative spectral–spatial features for better classification results. Second, the covariance pooling can fully exploit the second-order information contained in spectral–spatial feature maps and achieve appealing results while reducing hyperparameters in subsequent processing. Third, the channel-wise shift and channel-wise weighting are useful tricks to highlight the information of important spectral bands. However, it is not recommended to use channel-wise weighting together with covariance pooling when the number of training samples is already enough.

We finally show the structure and the parameter number of our model in Table XI. We can find that the experimental results of MCNN-CP are generated with relatively

small amount of training parameters. It is noted that without using PCA to reduce the dimensionality of the channels, the amount of model parameters will be four times more, as listed in Table XI, which is undoubtedly fatal for training efficiency and equipment requirements.

V. Conclusion

In this article, a mixed use of 3-D and 2-D CNN with covariance pooling is proposed for HSI classification. In contrast to using 3-D or 2-D convolution alone, the mixed usage can combine the ability of 3-D convolution that extracts spectral–spatial features and the advantage of 2-D convolution that fuses spectral bands for better discrimination and fewer model parameters. Besides, the covariance pooling strategy based on second-order statistics can make full use of spectral information and spatial information. In addition, two schemes, channel-wise shift and channel-wise weighting, are proposed to further reveal richer information and recalibrate channel-wise feature responses by giving prominence to informative channels and suppressing the effects of useless spectral bands. The experimental results demonstrate that our proposed network architecture can effectively enhance the accuracy of HSI classification. However, the current two schemes of channel-wise weighting and covariance pooling cannot be used together to guarantee a better performance. Besides, the weight obtained as (1+Ratio) may be not the best choice, either. In the near future, we will make effort to find better weights for more universal applications of our model.

References

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

[2] C. Camino, V. González-Dugo, P. Hernández, J. C. Sillero, and P. J. Zarco-Tejada, "Improved nitrogen retrievals with airborne-derived fluorescence and plant traits quantified from VNIR-SWIR hyperspectral imagery in the context of precision agriculture," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 70, pp. 105–117, Aug. 2018.

[3] R. J. Murphy, B. Whelan, A. Chlingaryan, and S. Sukkarieh, "Quantifying leaf-scale variations in water absorption in lettuce from hyperspectral imagery: A laboratory study with implications for measuring leaf water content in the context of precision agriculture," *Precis. Agricult.*, vol. 20, no. 4, pp. 767–787, Aug. 2019.

[4] I. Makki, R. Younes, C. Francis, T. Bianchi, and M. Zucchetti, "A survey of landmine detection using hyperspectral imaging," *ISPRS J. Photogramm. Remote Sens.*, vol. 124, pp. 40–53, Feb. 2017.

[5] M. Shimoni, R. Haelterman, and C. Perneel, "Hypersectral imaging for military and security applications: Combining myriad processing and sensing techniques," *IEEE Geosci. Remote Sens. Mag.*, vol. 7, no. 2, pp. 101–117, Jun. 2019.

[6] R. Nigam, B. K. Bhattacharya, R. Kot, and C. Chattopadhyay, "Wheat blast detection and assessment combining ground-based hyperspectral and satellite based multispectral data," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 42, pp. 473–475, Jul. 2019.

[7] R. D. M. Scafutto, C. R. de Souza Filho, and W. J. de Oliveira, "Hyperspectral remote sensing detection of petroleum hydrocarbons in mixtures with mineral substrates: Implications for onshore exploration and monitoring," *ISPRS J. Photogramm. Remote Sens.*, vol. 128, pp. 146–157, Jun. 2017.

[8] T. Bajjouk *et al.*, "Detection of changes in shallow coral reefs status: Towards a spatial approach using hyperspectral and multispectral data," *Ecol. Indicators*, vol. 96, pp. 174–191, Jan. 2019.

[9] X. Chen, H. Lee, and M. Lee, "Feasibility of using hyperspectral remote sensing for environmental heavy metal monitoring," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 42, pp. 1–4, Mar. 2019.

[10] F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 8, pp. 1778–1790, Aug. 2004.

[11] L. Ma, M. M. Crawford, and J. Tian, "Local manifold learning-based k-nearest-neighbor for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 11, pp. 4099–4109, Nov. 2010.

[12] J. Ham, Y. Chen, M. M. Crawford, and J. Ghosh, "Investigation of the random forest framework for classification of hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 3, pp. 492–501, Mar. 2005.

[13] Y. Zhou, J. Peng, and C. L. P. Chen, "Extreme learning machine with composite kernels for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 6, pp. 2351–2360, Jun. 2015.

[14] X. Kang, X. Xiang, S. Li, and J. A. Benediktsson, "PCA-based edge-preserving features for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 12, pp. 7140–7151, Dec. 2017.

[15] J. Wang and C.-I. Chang, "Independent component analysis-based dimensionality reduction with applications in hyperspectral image analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 6, pp. 1586–1600, Jun. 2006.

[16] T. V. Bandos, L. Bruzzone, and G. Camps-Valls, "Classification of hyperspectral images with regularized linear discriminant analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 3, pp. 862–873, Mar. 2009.

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

[18] L. He, J. Li, C. Liu, and S. Li, "Recent advances on spectral–spatial hyperspectral image classification: An overview and new guidelines," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 3, pp. 1579–1597, Mar. 2018.

[19] Z. Wu *et al.*, "GPU parallel implementation of spatially adaptive hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 4, pp. 1131–1143, Apr. 2018.

[20] Y. Chen, N. M. Nasrabadi, and T. D. Tran, "Hyperspectral image classification using dictionary-based sparse representation," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 10, pp. 3973–3985, Oct. 2011.

[21] G. Cheng, Z. Li, J. Han, X. Yao, and L. Guo, "Exploring hierarchical convolutional features for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 11, pp. 6712–6722, Nov. 2018.

[22] J. Li, P. R. Marpu, A. Plaza, J. M. Bioucas-Dias, and J. A. Benediktsson, "Generalized composite kernel framework for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 9, pp. 4816–4829, Sep. 2013.

[23] A. O. B. Ozdemir, B. E. Gedik, and C. Y. Y. Cetin, "Hyperspectral classification using stacked autoencoders with deep learning," in *Proc. 6th Workshop Hyperspectral Image Signal Process., Evol. Remote Sens. (WHISPERS)*, Jun. 2014, pp. 1–4.

[24] X. Yao, J. Han, G. Cheng, X. Qian, and L. Guo, "Semantic annotation of high-resolution satellite images via weakly supervised learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 6, pp. 3660–3671, Jun. 2016.

[25] T. Li, J. Zhang, and Y. Zhang, "Classification of hyperspectral image based on deep belief networks," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 5132–5136.

[26] P. Liu, H. Zhang, and K. B. Eom, "Active deep learning for classification of hyperspectral images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 2, pp. 712–724, Feb. 2017.

[27] Z. Ping, Z. Gong, S. 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, Mar. 2017.

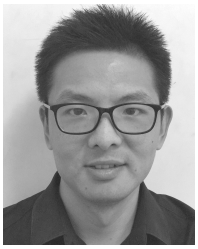
[28] J. Zhang, P. Liu, F. Zhang, and Q. Song, "CloudNet: Ground-based cloud classification with deep convolutional neural network," *Geophys. Res. Lett.*, vol. 45, no. 16, pp. 8665–8672, Aug. 2018.

[29] C. Bai, L. Huang, X. Pan, J. Zheng, and S. Chen, "Optimization of deep convolutional neural network for large scale image retrieval," *Neurocomputing*, vol. 303, pp. 60–67, Aug. 2018.

[30] M. He, B. Li, and H. Chen, "Multi-scale 3D deep convolutional neural network for hyperspectral image classification," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 3904–3908.

[31] 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.

[32] W. Li, G. Wu, F. Zhang, and Q. Du, "Hyperspectral image classification using deep pixel-pair features," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 2, pp. 844–853, Feb. 2017.
[33] J. Yang, Y. Zhao, J. C.-W. Chan, and C. Yi, "Hyperspectral image classification using two-channel deep convolutional neural network," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2016, pp. 5079–5082.
[34] 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.
[35] J. M. Haut, M. E. Paoletti, J. Plaza, J. Li, and A. Plaza, "Active learning with convolutional neural networks for hyperspectral image classification using a new Bayesian approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 11, pp. 6440–6461, Nov. 2018.
[36] 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. (IGARSS)*, Jul. 2015, pp. 4959–4962.
[37] J. Carreira, R. Caseiro, J. Batista, and C. Sminchisescu, "Free-form region description with second-order pooling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 6, pp. 1177–1189, Jun. 2015.
[38] D. Acharya, Z. Huang, D. P. Paudel, and L. Van Gool, "Covariance pooling for facial expression recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 480–4807.
[39] P. Li, J. Xie, Q. Wang, and Z. Gao, "Towards faster training of global covariance pooling networks by iterative matrix square root normalization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 947–955.
[40] T.-Y. Lin, A. RoyChowdhury, and S. Maji, "Bilinear CNN models for fine-grained visual recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1449–1457.
[41] N. He, L. Fang, S. Li, A. Plaza, and J. Plaza, "Remote sensing scene classification using multilayer stacked covariance pooling," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 12, pp. 6899–6910, Dec. 2018.
[42] N. He, L. Fang, S. Li, J. Plaza, and A. Plaza, "Skip-connected covariance network for remote sensing scene classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 5, pp. 1461–1474, May 2020.
[43] Y. Li, Y. Chen, N. Wang, and Z.-X. Zhang, "Scale-aware trident networks for object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6054–6063.
[44] X. Zhu, H. Hu, S. Lin, and J. Dai, "Deformable ConvNets v2: More deformable, better results," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9308–9316.
[45] W. Luo, Y. Li, R. Urtasun, and R. Zemel, "Understanding the effective receptive field in deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 4898–4906.
[46] J. Han, X. Yao, G. Cheng, X. Feng, and D. Xu, "P-CNN: Part-based convolutional neural networks for fine-grained visual categorization," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Aug. 6, 2019, doi: 10.1109/TPAMI.2019.2933510.
[47] G. Chen and S.-E. Qian, "Denoising of hyperspectral imagery using principal component analysis and wavelet shrinkage," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 3, pp. 973–980, Mar. 2011.
[48] W. Sun and Q. Du, "Graph-regularized fast and robust principal component analysis for hyperspectral band selection," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 6, pp. 3185–3195, Jun. 2018.
[49] X. Yu, S. Xiong, Y. Gao, and X. Yuan, "Contour covariance: A fast descriptor for classification," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*. Cham, Switzerland: Springer, Sep. 2019, pp. 589–600.
[50] P. Li, J. Xie, Q. Wang, and W. Zuo, "Is second-order information helpful for large-scale visual recognition?" in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 2089–2097.
[51] L. Fang, N. He, S. Li, A. J. Plaza, and J. Plaza, "A new spatial–spectral feature extraction method for hyperspectral images using local covariance matrix representation," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 6, pp. 3534–3546, Jun. 2018.
[52] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache, "Geometric means in a novel vector space structure on symmetric positive-definite matrices," *SIAM J. Matrix Anal. Appl.*, vol. 29, no. 1, pp. 328–347, Jan. 2007.
[53] A. B. Hamida, A. Benoit, P. Lambert, and C. B. Amar, "3-D deep learning approach for remote sensing image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 8, pp. 4420–4434, Aug. 2018.

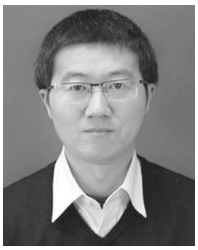


Jianwei Zheng received the B.S. degree in electronic and computer engineering and the Ph.D. degree in control theory and control engineering from the Zhejiang University of Technology, Hangzhou, China, in 2005 and 2010, respectively.

He is an Associate Professor with the College of Computer Science and Technology, Zhejiang University of Technology. His research interests include machine learning and compressive sensing. He has authored over 60 journal articles and conference papers in these areas.

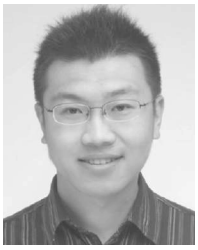
Yuchao Feng received the B.E. degree from Wenzhou University, Wenzhou, China, in 2019. He is pursuing the B.S. degree with the College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, China.

His current research interests include remote-sensing image process and pattern recognition.



Cong Bai received the B.E. degree from Shandong University, Jinan, China, in 2003, the M.E. degree from Shanghai University, Shanghai, China, in 2009, and the Ph.D. degree from the National Institute of Applied Sciences, Rennes, France, in 2013.

He is an Associate Professor with the College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, China. His research interests include computer vision and multimedia processing.



Jinglin Zhang received the B.E. degree from South Central University for Nationalities, Wuhan, China, in 2007, the M.E. degree from Shanghai University, Shanghai, China, in 2010, and the Ph.D. degree from the National Institute of Applied Sciences, Rennes, France, in 2013.

He is a Professor with the College of Atmospheric Sciences, Nanjing University of Information Science and Technology, Nanjing, China. His research interests include computer vision, high-performance computing, and interdisciplinary research with pattern recognition and atmospheric science.