A Hybrid Convolutional Neural Network with Anisotropic Diffusion for Hyperspectral Image Classification

Feng Lu, Qichao Liu, Mohsen Molaei, and Liang Xiao(✉)

School of Computer Science and Engineering,
Nanjing University of Science and Technology, Nanjing, China
xiaoliang@mail.njust.edu.cn

Abstract. Recent research has shown that methods based on deep convolutional neural networks (DCNN) can achieve high accuracy in the classification of hyperspectral image (HSI). However, convolution operations with different dimensions in deep neural networks usually perform in an isotropic structure, resulting in the loss of extracting deep feature of anisotropic neighborhood. Thus how to improve the ability of discriminative features learning is the key issue in DCNN. In this paper, we propose an anisotropic diffusion partial differential equation (PDE) driven hybrid CNN framework, named PM-HCNN. The proposed framework uses 2D convolution and 3D convolution layers to extract of spectral and spatial contexts in HSI. And a PDE based diffusion layer is cascaded as feature propagation layers after hybrid convolution layers to propagate the intrinsical discriminative features of various classes. Due to the anisotropic diffusion on the feature space, the classification mistakes of traditional CNNs with a small number of training data can be further eliminated while object boundaries can be preserved. Experimental results on several popular datasets show that the proposed PM-HCNN achieved state-of-the-art performance compared with the existing deep learning-based methods.

Keywords: Convolutional neural networks · Hyperspectral image classification · Anisotropic diffusion model · Feature propagation

1 Introduction

Hyperspectral remote sensors provide an effective method for human to observe the Earth's surface. Hyperspectral imagery (HSI) with high resolution in both spectral and spatial domains (i.e. derived from sensor systems) can be used in a wide range of specific applications, such as agriculture, physics, and surveillance. In this case, the acquired hyperspectral images can provide an almost continuous spectral curve for each pixel in the image. Capturing rich spectral information while acquiring spatial information, facilitates the generation of complex models to identify and classify different types of materials or plants in the images. However, the increase of the amount of hyperspectral image data and the redundancy in spectral information bring great challenges to the classification of hyperspectral images.

Y. Zhao et al. (Eds.): ICIG 2019, LNCS 11903, pp. 541–552, 2019.
https://doi.org/10.1007/978-3-030-34113-8_45

In past decades, many effective methods have been proposed for the classification of HSIs. For example, k-nearest neighbor (KNN) [1] has been used to construct a distance function to analyze the similarity between testing samples and training samples. In [2], support vector machine (SVM) has more potential to determine the decision boundary between classes by kernel methods. Also inspired by hybrid huberized SVM (HHSVMs), a random HHSVM algorithm [3] for HSI was proposed. Meanwhile, low-dimensional sparse representation-based classification (SRC) [4] and random forest-based classifiers [5] have also been proved to be effective in HSI classification. Nevertheless, the existing methods mentioned above paid less attention to the spatial relationship of the neighboring pixels, and the feature vectors fed into the classifier are represented only by the spectral features of the pixels.

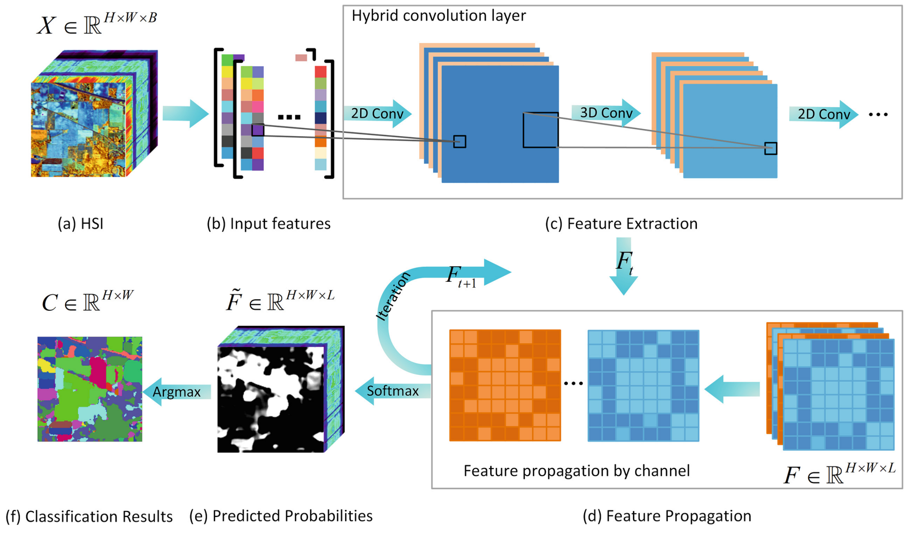
In recent years, spatial features have been considered in the classification process to resolve the issues raised by some pixels of different categories with similar spectrums. Consequently, some works have been conducted to introduce spatial information of HSI into classification, such as Markov Random Field (MRF) [6] and wavelet transform [7]. In the meantime, several traditional filters have also been adopted in the classification, such as bilateral filter, mean filter and Gabor filter. Furthermore, with considering spatial domains of HSI, for the first time in the [8], an edge retention filter [8] was proposed to construct the spectral and spatial information of HSIs. In [9], the Gabor filter was combined with the nearest regular subspace (RMS), and then the spatial features extracted by the filter were fed to the RMS classifier.

Recently, the deep learning models have been introduced into the field of computer vision as a powerful tool. Due to the powerful feature representation ability of deep networks, many typical deep learning methods have been applied into the HSI classification, such as deep belief networks (DBNs) [10] and convolutional neural networks (CNNs) [11, 12]. In order to enhance the utilization of HSI spectral-spatial structures, Li et al. [13] proposed a new hyperspectral classification framework based on a fully CNN, and integrated the optimized extreme learning machines (ELM). Cao et al. [14] combines Markov random fields and convolutional neural networks to classify images by formulating the problem from Bayesian. Additionally, Zhang et al. [15] uses a multi-scale summation approach based on regions, and then feeds all spectral and spatial information into a fully connected network.

Despite the significant improvements by deep networks in spectral-spatial joint classification, the standard convolution operations in deep networks usually perform in an isotropic structure, therefore, which fails to extract the features of anisotropic neighborhood of HSIs, a phenomenon which additionally challenges the accurate classification of pixels near the object boundaries. In order to eliminate the drawback of deep networks with isotropic convolution, we propose an anisotropic diffusion-driven hybrid CNN framework, named PM-HCNN. In this work, first we use 2D and 3D hybrid convolution layers to extract spectral-spatial features from original HSI, and then a diffusion layer is added after the hybrid convolution layers to propagate the intrinsical discriminative features of various classes. Lastly, to enable feature extraction layer and the diffusion layer work collaboratively, we integrated the diffusion layer into the whole network and trained together. The experimental results on two popular HSI datasets demonstrate that the proposed PM-HCNN achieved state-of-the-art performance compared with the existing deep learning-based methods.

2 Anisotropic Diffusion Driven Hybrid CNN

Figure 1 shows the whole deep learning framework of HSI classification with anisotropic diffusion. The whole deep learning framework is divided into two parts: a feature extraction layer based on 2D and 3D convolution operations and a feature propagation layer based on anisotropic diffusion. Since the network is an end-to-end framework, the input is $X \in \mathbb{R}^{H \times W \times B}$, i.e., indicates the original HSI, and the output is $\tilde{F} \in \mathbb{R}^{H \times W \times L}$, i.e., indicates the probabilities that the pixel belongs to each class.



$X \in \mathbb{R}^{H\times W\times B}$
Hybrid convolution layer
2D Conv
3D Conv
2D Conv
(a) HSI
(b) Input features
(c) Feature Extraction
F_t
F_{t+1}
Iteration
$C \in \mathbb{R}^{H\times W}$
$\tilde{F} \in \mathbb{R}^{H\times W\times L}$
Argmax
Softmax
Feature propagation by channel
$F \in \mathbb{R}^{H\times W\times L}$
(f) Classification Results
(e) Predicted Probabilities
(d) Feature Propagation

Fig. 1. The hybrid CNN with anisotropic diffusion (PM-HCNN) for hyperspectral image (HSI) classification. H, W, B, L denote the height, width, bands, and categories.

2.1 Hybrid 2D and 3D Convolution Layers

CNN automatically learns the features of images at various levels through convolution operations, which is consistence with our common sense of understanding images. Therefore, once CNN was proposed, its hierarchical design was gradually recognized as the most effective and successful technique in the field of computer vision.

2D-CNN can extract context representation features efficiently during the feature extraction. The value at position (x, y) on the ith feature map in the lth layer is given by Eq. (1):

$$F_{l,i}^{x,y} = \sigma\left(\sum_{m} \sum_{h=0}^{H_l-1} \sum_{w=0}^{W_l-1} K_{l,i,m}^{h,w} F_{(l-1),m}^{(x+h),(y+w)} + b_{l,i}\right) \tag{1}$$

where $F_{(l-1),m}^{(x+h),(y+w)}$ is the value at position $(x+h, y+w)$ on the mth feature map in the $(l-1)$th layer, m indexes over the set of the feature maps in the $(l-1)$th layer which is connected to the current feature map, $b_{l,i}$ is the bias of the ith feature map in the lth layer, H_l and W_l are the height and width of the kernels, $K_{l,i,m}^{h,w}$ stands for the value of the kernel connected to the ith feature map in the lth layer at the position (h, w), and $\sigma(\cdot)$ is the activation function.

Due to the large number of bands in the hyperspectral image, the 2D convolution operation will generate a large number of parameters, which leads to over-fitting. The 1×1 2D convolution kernel used in the experiment can not only solve the above problem, but also realize the cross-channel information combination and increase the nonlinear characteristics. What is more, in this experiment, we combine a 2D convolutional layer and a 3D convolutional layer into one hybrid convolutional layer. We use a 2D convolution layer to extract spectral features and obtain spectral features maps. The formula is changed as follows:

$$F_{l,i}^{x,y} = \sigma(\sum_{m} K_{l,i,m} F_{(l-1),m}^{x,y} + b_{l,i}) \tag{2}$$

where $K_{l,i,m}$ indicates the value of the ith spectral convolution kernel of the lth hybrid convolutional layer at the position m. Thereafter we use the 3D convolution layer to convolve the spectral feature maps and output the spatial-spectral feature maps. Normally, 3D convolution operations are used for 3D feature cubes in an effort to compute spatiotemporal features when the input data is 3D. The 3D convolution operation is formulated as follows:

$$F_{l,i}^{x,y,z} = \sigma(\sum_{m} \sum_{h=0}^{H_l-1} \sum_{w=0}^{W_l-1} \sum_{r=0}^{R_l-1} K_{l,i,m}^{h,w,r} F_{(l-1),m}^{(x+h),(y+w),(z+r)} + b_{l,i}) \tag{3}$$

where i is the number of kernels in this layer, $F_{l,i}^{x,y,z}$ is the value on the ith feature cube in the lth layer at position (x, y, z), R_l indicates the spectral depth of 3D kernel, $K_{l,i,m}^{h,w,r}$ is the (h, w, r)th value of the kernel linked to the mth feature cube in the previous layer. In our model, each feature cube is processed independently. So m in Eq. (3) need to be set 1, and the transformed formula for 3D convolution operation is as follows:

$$F_{l,i,j}^{x,y,z} = \sigma(\sum_{h=0}^{H_l-1} \sum_{w=0}^{W_l-1} \sum_{r=0}^{R_l-1} K_{l,i}^{h,w,r} F_{(l-1),j}^{(x+h),(y+w),(z+r)} + b_{l,i}) \tag{4}$$

where j is the number of feature cubes in the preceding layer, $K_{l,i}^{h,w,r}$ is the (h, w, r)th value of the kernel linked to the jth feature cube of the preceding layer, $F_{l,i,j}^{x,y,z}$ is the output that is calculated by convolving the jth feature cube of the preceding layer with the ith kernel of the lth layer at the position (x, y, z).

In summary, we use a 2D convolution, of which size is 1×1, to reduce the dimension of the channel without changing height and width of the input data, and then extract the spectral and space information by 3D convolution. In addition, a plurality of the above hybrid convolution layers are used to construct a deep network, and the characteristics of both 2D and 3D convolution can be fully utilized to facilitate the purpose of extracting deep spectral-spatial features.

2.2 Feature Propagation Layers with Anisotropic Diffusion

A hyperspectral image with B channels and $H \times W$ size will be put into the neural network as a whole. Take the Indian Pines dataset as an example, BN is added before each 2D convolution layer to speed up the convergence of gradients, save computational resources and shorten the training time in the experiment.

In the stage of feature extraction, the spectral dimension of the data is transformed into 128 when passing through the 2D convolution layer containing 128 $1 \times 1 \times B$ kernels. Moreover, for the 3D convolution layers, the kernel size is set to (3, 3, 7), the kernel number is set to 2, the method of padding is set to "same", and the stride is set to (1, 1, 1), so the output of this hybrid layer becomes 2 channel feature maps. The input of the second hybrid convolution layer is the output of the last hybrid convolution layer, and the output of this layer is the same type of feature maps after the same convolution operations. Finally, we use $1 \times 1 \times 256$ 2D convolution layer to perform a pixel-by-pixel convolution operation on the feature maps.

For the following detailed explanation, after passing through the last 2D convolution layer, the output is F, then the calculation formula of F is:

$$F_{x,y,m} = \sum_{m} O_{i,m} F'_{x,y,m} + b_i \tag{5}$$

where $F'_{x,y,m}$ is the value at the position (x, y) on the mth feature map which output by the last hybrid convolution layer, $O_{i,m}$ is the ith convolution kernel and b_i indicates bias.

In order to enhance the quality of the feature maps and propagate the intrinsical discriminative features of various classes, after the hybrid convolution layers, we cascade a feature propagation (FP) layers with anisotropic diffusion. The idea is borrowed from the nonlinear PDE which was proposed by Perona and Malik (also known as PM diffusion) in [16].

Let $F_t^{x,y}$ denotes the value at the position (x, y) on the final pth feature map $(p = 1, \ldots, L)$ in the tth iteration, the resulted anisotropic diffusion feature map can be defined as follows:

$$\begin{aligned} F_{t+1}^{x,y} = F_t^{x,y} + \lambda(\alpha_N \cdot \nabla_N(F_t^{x,y}) + \alpha_S \cdot \nabla_S(F_t^{x,y}) \\ + \alpha_E \cdot \nabla_E(F_t^{x,y}) + \alpha_W \cdot \nabla_W(F_t^{x,y})) \end{aligned} \tag{6}$$

where t represents the number of iterations, E, S, W, and N represent East, South, West, and North, respectively, and $\lambda \in [0, 1/4]$ is used for the stability of the numerical scheme. α_N represents the diffusion coefficient in the north which controls the rate of diffusion. $\nabla_N(F_t^{x,y})$ indicates the derivative in the north. They can be expressed in detail as follows:

$$\begin{aligned} \alpha_N = \exp(-\|\nabla_N F\|^2/k^2) \qquad \alpha_S = \exp(-\|\nabla_S F\|^2/k^2) \\ \alpha_E = \exp(-\|\nabla_E F\|^2/k^2) \qquad \alpha_W = \exp(-\|\nabla_W F\|^2/k^2) \end{aligned} \tag{7}$$

$$\begin{aligned} \nabla_N(F_t^{x,y}) = F_t^{x,y-1} - F_t^{x,y} \;\; \nabla_S(F_t^{x,y}) = F_t^{x,y+1} - F_t^{x,y} \\ \nabla_E(F_t^{x,y}) = F_t^{x-1,y} - F_t^{x,y} \;\; \nabla_W(F_t^{x,y}) = F_t^{x+1,y} - F_t^{x,y} \end{aligned} \tag{8}$$

In Eqs. (7) and (8), the constant term k is used to control the sensitivity to the edge. And the symbol ∇ needs more attention and should not be confused with the gradient operator ∇, as it is used to represent the nearest-neighbor differences. The entire formula requires three parameters to be set beforehand: the number of iterations t, parameter λ and thermal conductivity parameter k. Larger values of k and λ, corresponds to the smoother image, and makes it more difficult to preserve the marginal features of the image.

After the FP layer, the formula for converting F into the classification probability $\tilde{F}$ is defined as:

$$\tilde{F}_k = \frac{e^{F_k}}{\sum_{i=1}^{L} e^{F_i}} \tag{9}$$

where $\tilde{F}_k$ is the probability that a pixel at a certain position belonging to category $k(1 \leq k \leq L)$ in the hyperspectral image.

Finally, the classification results $C \in \mathbb{R}^{h \times w}$ is computed as follows:

$$C = \text{Argmax}(\tilde{F}) \tag{10}$$

In deep learning, the loss function can evaluate the quality of the model and provide the direction of optimization. PM-HCNN adopts cross entropy as the loss function. In the network, the training set consists of pixels with corresponding class labels in the hyperspectral image. If the pixel $X^{x,y}$ of the hyperspectral image at the position (x, y) is a training sample, then $X^{x,y} \in D_{train}$. $Q^{x,y}$ indicates the probability vectors at the position (x, y). When $X^{x,y}$ belongs to category $k(1 \leq k \leq L)$, the corresponding vector $Q^{x,y}$ at the kth position is 1, and the rest is 0. Let $V_{train} \in \mathbb{R}^{H \times W \times L}$ be the probability labels of the network output $\tilde{F}$. It is converted from the labels corresponding to the training sample. Then the element in V_{train} satisfies the following formula:

$$V_{train}^{x,y} = \begin{cases} Q^{x,y} & X^{x,y} \in D_{train} \\ 0 & otherwise \end{cases} \tag{11}$$

where 0 is a vector whose elements are all 0, $V_{train}^{x,y}$ indicates the probability vector at the (x, y) position. So the loss function of the network in the training phase is:

$$Loss(\tilde{F}, V_{train}) = -\sum_{x=1}^{H} \sum_{y=1}^{W} \sum_{k=1}^{L} V_{train}^{x,y,k} \log(\tilde{F}^{x,y,k}) \tag{12}$$

where $\tilde{F}^{x,y,k}$ and $V_{train}^{x,y,k}$ represent the specific value of $\tilde{F}$ and V_{train} at the position (x, y, k). The loss function of the verification set can also be obtained by the same reasoning.

3 Experimental Analysis

In order to verify the validity of the proposed model in the classification, this paper uses two different hyperspectral datasets for experiments: The Indian Pines (IN) and Kennedy Space Center (KSC) dataset. In addition, five methods based on deep learning are selected for the comparative experiments: 2D-CNN [17], DC-CNN [18], 3D-CNN [19], MC-CNN [20] and SSRN [21]. Meanwhile, the paper uses the overall accuracy (OA), average accuracy (AA) and kappa statistic to measure the classification result of each model. OA is the ratio of the number of class pixels of the correct classification to the total number of categories. AA is the average of the ratio between each type of prediction and the total number of each category. Kappa coefficient is a method based on confusion matrix to measure classification accuracy.

3.1 Experimental Environment and Parameters

In the experiment, the training and testing process were conducted on a same computer with the following configuration: CPU: i7-8700K, GPU: NVIDIA GeForce GTX 1080Ti and Memory: 32 GB.

For the IN dataset, in the training process, the optimizer is Adam, initial learning rate is 0.001, and the number of iterations is set to 500. The network consists of 4 hybrid convolution layers, and the number of output channels is set to 128. For the 3D convolution kernel, the kernel size is set to (3, 3, 7), the method of padding is set to "same", the stride is set to (1, 1, 1) and the activation function is Sigmoid. In addition, the parameters of the FP layer are set as follows: the number of iterations t is 7, the k of the thermal conductivity is 5 and λ is 1/7.

For the KSC dataset, in the training process, the optimizer is Adam, the initial learning rate is 0.0005, and the number of iterations is set to 300. The network consists of 4 hybrid convolution layers, and the number of output channels is set to 64. For the 3D convolution kernel, the kernel size is set to (5, 5, 7), the method of padding is set to "same", the stride is set to (1, 1, 1) and the activation function is Sigmoid. In addition, the parameters of the FP layer are set as follows: the number of iterations t is 3, the k of the thermal conductivity is 3 and λ is 1/8.

3.2 Datasets

The Indian Pine dataset was gathered by airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor in 1992 over the Indian Pines test site in Indiana, USA. It was the first benchmark dataset to be used for the study of hyperspectral image classification techniques with a cut size of 145 × 145. The Indian Pines scene consists of 21025 pixels and 224 spectral reflectance bands in the wavelength range 0.4–2.5 μm, but 24 bands that cannot be reflected by water need to be removed for this the study. Because of the tremendous unbalanced number of samples among different classes and mixed pixels in the image, so the Indian Pines dataset has been widely used to evaluate the performance of classification methods. The dataset covers sixteen categories. And the numbers of training, verification, and test samples which belong to different classes are shown in Table 1.

The KSC dataset was acquired by the NASA AVIRIS (Airborne Visible/Infrared Imaging Spectrometer) instrument over the Kennedy Space Center (KSC), Florida, on March 23, 1996. The spatial resolution of The KSC data is about 18 m. And after removing water absorption and low SNR bands, 176 bands remain. Due to the similarity of spectral signatures for certain vegetation types, it is difficult to distinguish the land cover for the environment. For classification purposes, 13 classes were defined for the site. The numbers of training, verification, and test samples which belong to different classes are shown in Table 2.

3.3 Experimental Results

In our experiment, we randomly selected 10 sets of training samples for each data set for repeated experiments. The final classification result is presented as "mean". We use the same proportion of training, validation, and test data for the same dataset.

For IN dataset, the split percentage of training, validation, and testing data is 10%, 1%, 89%, respectively. Figure 2 shows the classification maps for IN dataset obtained by different methods. Table 1 shows the exact value of OA, AA, Kappa for each class by different methods. Under the same conditions, PM-HCNN has improved on OA and Kappa when comparing to other methods. And it is only slightly lower than MC-CNN by 0.14% in AA. Compared with SSRN, the three indicators increased by 0.46%, 6.77% and 0.53% respectively.

For KSC dataset, we use 5% of the labeled pixels as the training set, 1% for validation and 94% for test datasets, respectively. The classification maps of different methods are shown in Fig. 3. And from Table 2, we can get the exact value of OA, AA, Kappa for each class by different methods. It is obvious that PM-HCNN achieves the best result with an overall accuracy of 98.57%, which is 0.69% higher than the second best (97.88%) obtained by SSRN, and is 5.55% higher than the result of 93.02% by DC-CNN. Also for AA and KAPPA, the proposed method achieves the best result.

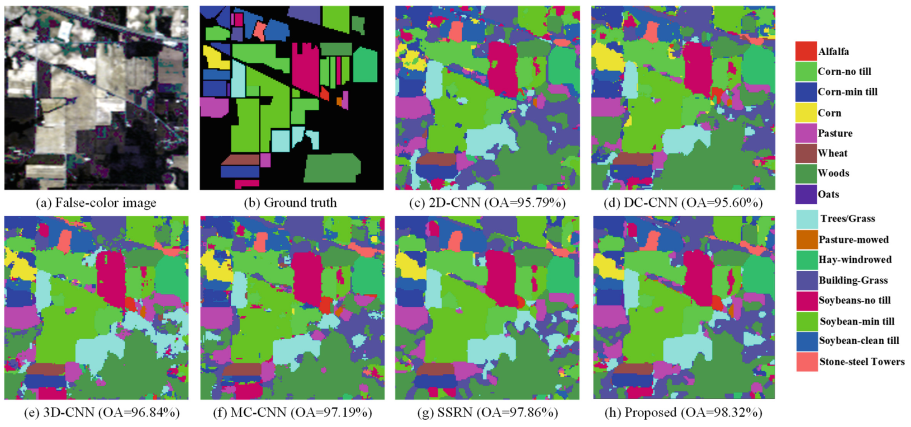


Fig. 2. The classification maps for IN dataset by different methods.

Table 1. Individual class, overall, average accuracy (%) and Kappa statistics of all methods on the Indian Pines dataset using 10% training samples and 1% validation samples.

Class	Samples			2D-CNN	DC-CNN	3D-CNN	MC-CNN	SSRN	Proposed
	Train	Val	Test						
Alfalfa	5	1	48	100	95.48	99.41	98.18	99.42	99.38
Corn-notill	143	14	1277	94.11	93.90	96.07	96.25	98.40	98.40
Corn-mintill	83	8	743	93.69	94.33	94.65	96.84	97.13	97.91
Corn	23	2	209	95.40	94.99	97.65	97.16	96.71	99.50
Pasture	49	4	444	96.87	98.17	98.76	99.03	97.20	99.80
Trees/Grass	74	7	666	98.35	98.54	98.00	98.61	99.01	99.05
Pasture-mowed	2	1	23	100	100	98.82	98.03	80	96.63
Hay-windrowed	48	4	437	96.58	98.19	99.09	99.45	99.45	99.86
Oats	2	1	17	100	93.84	95.00	97.64	0	93.86
Soybeans-notill	96	9	863	94.27	92.18	96.18	95.21	96.38	91.82
Soybean-mintill	246	24	2198	95.81	95.90	96.08	96.68	98.02	98.98
Soybean-cleantill	61	6	547	93.74	94.51	97.02	96.04	95.42	88.64
Wheat	21	2	189	99.68	99.37	99.78	99.58	98.95	98.27
Woods	129	12	1153	98.39	98.16	98.82	99.30	99.48	98.99
Building-Grass	38	3	339	95.67	92.18	94.72	95.71	96.76	98.02
Stone-steelTowers	9	1	85	94.48	93.15	95.48	95.48	97.05	99.35
OA				95.79	95.60	96.84	97.19	97.86	98.32
AA				96.69	95.81	97.22	97.45	90.59	97.36
Kappa				95.20	94.99	96.40	96.80	97.56	98.09

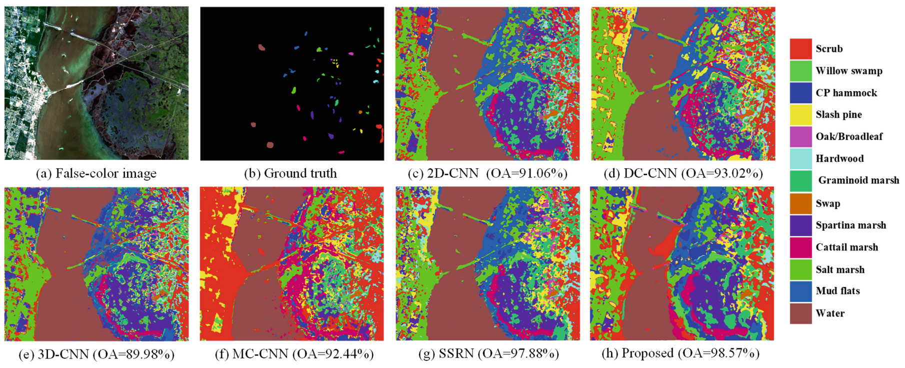


Fig. 3. The classification maps for KSC dataset by different methods

Table 2. Individual class, overall, average accuracy (%) and Kappa statistics of all methods on the Kennedy Space Center dataset using 5% training samples and 1% validation samples.

Class	Samples			2D-CNN	DC-CNN	3D-CNN	MC-CNN	SSRN	Proposed
	Train	Val	Test						
Scrub	18	4	325	97.65	98.54	98.58	98.97	98.84	99.23
Willow swamp	13	3	227	88.78	90.17	82.31	89.13	97.07	98.93
CP hammock	13	3	240	64.38	74.71	73.55	78.51	97.13	98.85
Slash pine	13	3	236	67.29	78.33	60.33	76.57	89.77	95.57
Oak/broadleaf	9	2	150	65.42	63.94	64.64	61.00	87.78	89.33
Hardwood	12	3	214	79.15	89.88	79.29	86.72	99.32	95.67
Swamp	6	2	97	80.14	87.94	77.98	82.79	93.55	84.91
Graminoid marsh	20	4	366	88.49	92.59	93.37	92.93	98.59	99.37
Spartina marsh	26	6	488	90.87	94.10	88.06	88.86	98.50	98.86
Cattail marsh	21	5	378	99.52	95.57	98.36	99.15	99.32	99.67
Salt marsh	21	5	393	99.64	99.00	99.69	99.84	99.74	100
Mud flats	26	6	471	97.98	96.34	89.44	95.47	98.16	97.96
Water	47	10	870	98.98	100	98.56	99.21	100	100
OA				91.06	93.02	89.98	92.44	97.88	98.57
AA				86.02	89.32	84.93	88.40	96.75	96.80
Kappa				90.04	92.23	88.84	91.58	97.64	98.41

In summary, for the two datasets, what is clear is that PM-HCNN can retain the little features at the boundary of regions belonging to different categories and make some different categories of features more visible. Furthermore, the proposed method can also eliminate the classification mistakes of traditional CNNs. So the classification accuracy has been improved when comparing to other methods.

4 Conclusion

In this paper, we propose a hybrid convolutional neural network with Perona and Malik diffusion (PM-HCNN) and apply it to hyperspectral image classification. The proposed PM-HCNN framework uses an end-to-end convolutional architecture to automatically extract spectral and spatial features through hybrid convolution layers which contains 2D and 3D convolution layers. For the loss of extracting deep feature of anisotropic neighborhood after using convolution operations, the framework contains a feature propagation layer based on anisotropic diffusion. The layer propagates the intrinsical discriminative features of various classes and makes each class more distinguishable from every other classes. These improvements result in preserving object boundaries and eliminate the classification mistakes of traditional CNNs with a small number of training data. Equipped with the analysis of experimental results, PM-HCNN framework can be used to get better classification accuracy than other methods. The future direction of our work is to established a unified diffusion driven CNN architecture, in which all the hyper-parameters both in the diffusion unit and CNN can be learnt from the training sets.

Acknowledgments. This work has been supported by the Jiangsu Provincial Social Developing Project (BE2018727), the Fundamental Research Funds for the Central Universities (Grant No.30918011104), the National Major Research Plan of China (Grant No. 2016YFF0103604), the Jiangsu Provincial Natural Science Foundation of China (Grant No. BK20161500).

References

1. Blanzieri, E., Melgani, F.: Nearest neighbor classification of remote sensing images with the maximal margin principle. IEEE Trans. Geosci. Remote Sens. **46**(6), 1804–1811 (2008)
2. Melgani, F., Bruzzone, L.: Classification of hyperspectral remote sensing images with support vector machines. IEEE Trans. Geosci. Remote Sens. **42**(8), 1778–1790 (2004)
3. Liu, W., Shen, X., Du, B., Tsang, I.W., Zhang, W., Lin, X.: Hyperspectral imagery classification via stochastic HHSVMs. IEEE Trans. Image Process. **28**(2), 577–588 (2019)
4. Liu, J., Wu, Z., Wei, Z., Xiao, L., Sun, L.: Spatial-spectral kernel sparse representation for hyperspectral image classification. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. **6**(6), 2462–2471 (2013)
5. Ham, J., Chen, Y., Crawford, M.M., Ghosh, J.: Investigation of the random forest framework for classification of hyperspectral data. IEEE Trans. Geosci. Remote Sens. **43**(3), 492–501 (2008)
6. Zhang, B., Li, S., Jia, X., Gao, L., Peng, M.: Adaptive Markov random field approach for classification of hyperspectral imagery. IEEE Geosci. Remote Sens. Lett. **8**(5), 973–977 (2011)
7. Tang, Y., Lu, Y., Yuan, H.: Hyperspectral image classification based on three-dimensional scattering wavelet transform. IEEE Trans. Geosci. Remote Sens. **53**(5), 2467–2480 (2015)
8. Kang, X., Li, S., Benediktsson, J.A.: Spectral-spatial hyperspectral image classification with edge-preserving filtering. IEEE Trans. Geosci. Remote Sens. **52**(5), 2666–2677 (2014)
9. Li, W., Du, Q.: Gabor-filtering-based nearest regularized subspace for hyperspectral image classification. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. **7**(4), 1012–1022 (2014)
10. Chen, Y., Zhao, X., Jia, X.: Spectral-spatial classification of hyperspectral data based on deep belief network. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. **8**(6), 2381–2392 (2015)
11. Hu, W., Huang, Y., Wei, L., Zhang, F., Li, H.: Deep convolutional neural networks for hyperspectral image classification. J. Sens. **2015**(2), 1–12 (2015)
12. Lee, H., Kwon, H.: Going deeper with contextual CNN for hyperspectral image classification. IEEE Trans. Image Process. **26**(10), 4843–4855 (2017)
13. Li, J., Zhao, X., Li, Y., Du, Q., Xi, B., Hu, J.: Classification of hyperspectral imagery using a new fully convolutional neural network. IEEE Geosci. Remote Sens. Lett. **26**(10), 4843–4855 (2017)
14. Cao, X., Zhou, F., Xu, L., Meng, D., Xu, Z., Paisley, J.: Hyperspectral image classification with Markov random fields and a convolutional neural network. IEEE Trans. Image Process. **27**(5), 2354–2367 (2018)
15. Zhang, M., Li, W., Du, Q.: Diverse region-based CNN for hyperspectral image classification. IEEE Trans. Image Process. **27**(6), 2623–2634 (2018)

16. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. IEEE Trans. Pattern Anal. Mach. Intell. **12**(7), 629–639 (1990)
17. Makantasis, K., Karantzalos, K., Doulamis, A., Doulamis, N.: Deep supervised learning for hyperspectral data classification through convolutional neural networks. In: IEEE International Geoscience and Remote Sensing Symposium (IGARSS), pp. 4959–4962 (2015)
18. Zhang, H., Li, Y., Zhang, Y., Shen, Q.: Spectral-spatial classification of hyperspectral imagery using a dual-channel convolutional neural network. Remote Sens. Lett. **8**(5), 438–447 (2017)
19. Li, Y., Zhang, H., Shen, Q.: Spectral–spatial classification of hyperspectral imagery with 3D convolutional neural network. Remote Sens. **9**(1), 67 (2017)
20. Chen, C., Zhang, J.-J., Zheng, C.-H., Yan, Q., Xun, L.-N.: Classification of hyperspectral data using a multi-channel convolutional neural network. In: Huang, D.-S., Gromiha, M., Han, K., Hussain, A. (eds.) ICIC 2018. LNCS (LNAI), vol. 10956, pp. 81–92. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-95957-3_10
21. Zhong, Z., Li, J., Luo, Z., Michael, C.: Spectral-spatial residual network for hyperspectral image classification: a 3-D deep learning framework. IEEE Trans. Geosci. Remote Sens. **56**(2), 847–858 (2018)