Hyperspectral Image Classification Method Based on CNN Architecture Embedding With Hashing Semantic Feature

Chunyan Yu, Meng Zhao, Meiping Song, Yulei Wang, Fang Li, Rui Han, and Chein-I Chang, *Life Fellow, IEEE*

***Abstract*—Deep convolutional neural networks (CNN) have led to a successful breakthrough for hyperspectral image (HSI) classification. In this paper, a CNN system embedded with an extracted hashing feature is proposed for HSI classification that utilizes the semantic information of the HSI. First, a series of hash functions are constructed to enhance the presentation of the locality and discriminability of classes. Then, the sparse binary hash codes calculated by the discriminative learning algorithm are combined into the original HSI. Next, we design a CNN framework with seven hidden layers to obtain the hierarchical feature maps with both spectral and spatial information for classification. A deconvolution layer aims to improve the robustness of the proposed CNN network and is used to enhance the expression of deep features. The proposed CNN classification architecture achieves powerful distinguishing ability from different classes. The extensive experiments on real hyperspectral images results demonstrate that the proposed CNN network can effectively improve the classification accuracy after the embedding of the extracted semantic features.**

***Index Terms*—Convolutional neural networks (CNN), hashing learning, hyperspectral image classification (HSIC), semantic feature extraction (SFE).**

I. Introduction

DUE to the continuous spectral channels with rich spectral information and high-resolution spatial structure, hyperspectral images are widely used in the fields of precision agriculture, forestry monitoring, absorption band mapping techniques on Mars and terrestrial drill core applications [1], [2]. Among these applications, hyperspectral image classification (HSIC) is the most fundamental research that has attracted more and more attention. In recent years, with the booming of artificial intelligence and big data theory [3], [4], a set of hyperspectral image classification methods based on manifold learning and sparse theory have achieved satisfactory performance [5]–[12]. However, these methods produce classification results using mechanism of shallow layer, cannot deal with the complex classification problem. Deep learning (DL) [13] allows the computer to automatically extract deep features and more abstract features to improve the accuracy of the classification and has been widely used in HSIC fields. As the most popular and successful DL framework, convolutional neural network (CNN) utilizes a series of hidden layers to extract hierarchical features that has proved to be effective in HSIC [14]–[20].

A hyperspectral image is originally a three-dimensional cube with the spectral and spatial continuity, recently which have integrated both spectral and spatial information have gained more popularity. Ma *et al*. [14] described a semi-supervised classification method using the local category labels of the samples and the global category labels obtained by the DL framework to perform HSIC with self-learning methods. In [15], an unsupervised representation learning method is proposed to investigate deconvolution networks for remote sensing scene classification. In [16], a deep feature fusion network with the residual learning to optimize several convolutional layers is proposed for HSI classification. In [17], a deeper CNN which directly learns end-to-end mapping between HSI and the labels was proposed by using a new spatial feature for selecting a band with spatial information enhancement.

The existing research works of the classification methods based on the CNN framework provide rich solutions for hyperspectral classification technology. However, there are two sides of problems of the CNN classification framework which should be mentioned at present. First, usually more than three convolution layers are presented to extract more features which caused the mathematical models getting more and more complex, and at the same time, the complexity of the models brings uninterpretable features in the CNN framework. Second, the CNN classification framework is a black box operation for deep features

Manuscript received February 22, 2019; accepted April 13, 2019. Date of publication May 21, 2019; date of current version July 17, 2019. This work was supported in part by the National Natural Science Foundation of Liaoning Province under Grant 20170540095, in part by the Fundamental Research Funds for Central Universities under Grants 3132016331, 3132019208, and 3132019218, in part by Recruitment Program of Global Experts for National Science and Technology Major Project, State Administration of Foreign Experts Affairs funded by ZD20180073, and in part by the National Natural Science Foundation of China under Grants 61601077, 61801075, and 41801231. *(Corresponding author: Meiping Song.)*

C. Yu, M. Zhao, M. Song, Y. Wang, F. Li, and R. Han are with the Center of Hyperspectral Imaging in Remote Sensing, Information and Technology College, Dalian Maritime University, Dalian 116026, China (e-mail: yucy@dlmu.edu.cn; 392958198@qq.com; smping@163.com; 7340912@qq.com; lifang0105@qq.com; 269899266@qq. com).

C.-I Chang is with the Center of Hyperspectral Imaging in Remote Sensing, Information and Technology College, Dalian Maritime University, Dalian 116026, China, with the National Yunlin University of Science and Technology, Yunlin 64002, Taiwan, with the Remote Sensing Signal and Image Processing Laboratory, Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore, MD 21250 USA, and also with the Department of Computer Science and Information Management, Providence University, Taichung 02912, Taiwan (e-mail: cchang@umbc.edu).

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

Digital Object Identifier 10.1109/JSTARS.2019.2911987

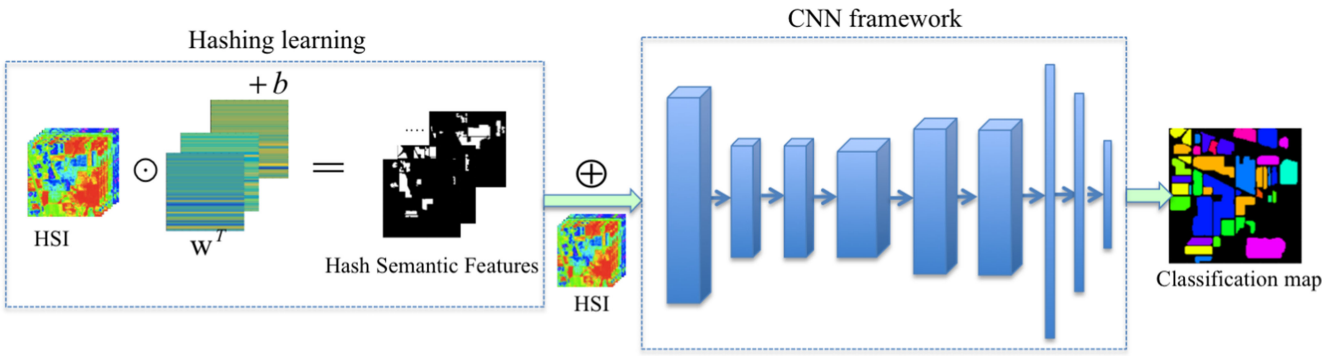


Fig. 1. CNN classification system with hashing feature extraction.

exploration and it is not sufficient for achieving better presentation of classification by employing only CNN. Nowadays, the fused profiles fed into the CNN network mainly obtained by the HSI data transformation or segmentation results [21]–[24]. Motivated by adding semantic features of HSI which can improve the classification efficiency and performance [25], in this paper, we presented a new classification method with hashing semantic feature fused into the CNN architecture. On one hand, hash learning is used to encode both spectral features and spatial neighborhood information simultaneously to improve the distinguishing ability of the different classes. On the other hand, we adopted a simpler CNN architecture with two convolution layers to explore the classification features.

Learning to hash approach aims to transform the original high-dimensional data to a low-dimensional representation, by learning a hash function, $y = h(x)$, in recent years, due to that the learned hash codes are able to preserve the proximity of neighboring data [26], hashing learning has attracted more attentions in designing efficient indexing and information retrieval field. The mechanism of hash learning is mainly divided into data independence and data dependency. Data-independent methods include locality-sensitive hashing [27] and its extended version. The main disadvantage of this type of function is random mapping, which is independent of the data itself, so it cannot reflect the characteristics of hyperspectral data. In this paper, the data-related hash function was used which maps data into binary codes to keep the similarities between the original feature space and the transformed space.

Our approach implemented the hyperspectral classification task by embedding a semantic feature map into the CNN network to promote classification and recognition. Fig. 1 shows the flowchart of the proposed CNN classification system. First, the semantic features present the similarities of the same class which are extracted by hashing learning, then the features are embedded in the original hyperspectral image for CNN classification. Next, the flow of the proposed CNN classification method based on the space-spectrum combination can be simply summarized as follows. In order to improve the accuracy and the convergence speed of the model, the input hyperspectral data needs to be normalized, then the deep feature maps are extracted through a convolution layer and the nonlinear features are obtained after an activation function. Next the deconvolution layer is used to enhance the feature downsampled by pooling layer, and finally the two fully connection layers finish the feature mapping and the deep features are converted to the softmax classifiers. The paper contributes to the literature containing three major aspects.

1) We present a new semantic feature extraction method by considering the locality and discriminative learning subspace between feature categories, and using compact codes semantically to encode the hyperspectral data. With the defined within-class and intraclass similarity constraints, the extracted features provide salient classification information for the followed CNN framework by maximizing the sample center distance.
2) Besides the spectral and spatial information, the proposed CNN classification architecture achieves powerful distinguishing ability from different classes by utilizing the extracted semantic features merged in the original HSI cube, and it explored the convolutional features and semantic contextual information simultaneously.
3) A simpler CNN network including two convolutional layers is adopted for HSI classification, and the deconvolution layer is designed to enhance the deep features that improve the robustness of the classification framework. Besides, to demonstrate the performance of the deconvolution layer and the pre-processed hashing procedure, we also designed other several CNN architectures to evaluate the classification performance correspondingly.

The remaining part of the paper is organized as follows. The proposed semantic feature extraction based on hashing learning is presented in Section II. Section III introduces the proposed CNN network for classification. Experimental result analysis is illustrated in Section IV and conclusions are drawn in Section V.

II. SFE Based on Hashing Learning

A. Hash Function Definition

Assume that a hyperspectral image is denoted by $\{\mathbf{r}_k\}_{k=1}^{N} \in \Omega$, where N is the number of the data sample, $\mathbf{r}_k = (r_{k1}, r_{k2}, ...r_{kL})^T$ is the kth data sample of the HSI cube, and L is the total number of spectral bands. We define categories of pixels denoted asy_{r_k}, where$y_{\mathrm{r}_k} \in \{0, 1\}$ depends on whether the pixel is the target pixel or not, value "0" indicates that it is

not the target of the *i*th class and "1" shows it is the target of the *i*th class. In this paper, a series of hash functions are defined as h the value of the hash function $h(r_k) = y_{r_k}{}' = \{0, 1\}$ represents whether the pixel belongs to the *class* or not. The specific hash function is defined as follows:

$$h(r_k) = \operatorname{sgn}\left(w^T r_k + b\right) \tag{1}$$

where w is the subspace projection vector and b is the offset [26], x denotes the original input signal.

The goal of hash learning is to minimize the difference between the value of the projection space and the target space. The within-class similarity in the same class should be small after the hash function, while the intraclass similarity between classes should be increased. Therefore, to distinguish two types of classes by maximizing the sample center distance, the loss function is defined as follows:

$$\operatorname{loss}(w, b) = \frac{CB}{CW} \tag{2}$$

where *CB* denotes the intraclass similarity, while the *CW* denotes the within-class similarity.

B. Similarity Preserving

The basic and important rule of the loss function designing is to preserve the similarities of the original HSI space. In what follows, the similarity definition of intraclass and within-class results and the learning algorithm of the loss function are discussed in details.

1) Intraclass Similarity: In order to make the final hash mapping codes have a better discriminability to distinguish the class and nonclass information, the similarities of intraclass include the similarity between the same class and the different classes. In this paper, we evaluated the similarities between classes with the Euclidean distance of the class center.

For the *i*th class C_i, the similarity level between the C_{i} and non-*i*th class center u_i is defined as follows:

$$cb1 = \phi(h, d_i, u_i) = \left\| y'_{d_i} - y'_{u_i} \right\|^2 \tag{3}$$

where d_i is the cluster center of C_i defined by the sample mean of the class of C_i, and u_i is not-*i*th class center defined as follows:

$$u_i = \frac{1}{p-1} \sum_{j=1, j\neq i}^{p} d_j.$$

Furthermore, the similarity level between the C_i and C_j is defined as follows:

$$cb2 = \sum_{j=1(j\neq i)}^{p} \phi(h, d_i, d_j) = \sum_{j=1(j\neq i)}^{p} \left\| y'_{d_i} - y'_{d_j} \right\|^2 \tag{4}$$

where d_i and d_j are the class center of C_i and C_j defined as same as above.

2) Within-Class Similarity: The similarity of with-in class is measured by calculating three types of distance. On one side, on account of the locality between the samples in the same class would have the same original feature, for the sample of the C_i, $r_k \in C_i$, the distance between the sample of r_k and the class center of C_i is designed to measure the within-class similarity of C_i

$$cw1 = \sum_{r_k \in C_i} \phi(h, r_k, d_i) = \sum_{r_k \in C_i} \left\| y'_{r_k} - y'_{d_i} \right\|^2. \tag{5}$$

On the other side, in order to emphasize the effect of the spatial features in local neighbor, the other within-class similarity is measured by calculating the distance between the pixel r_k and it's neighborhood pixels.

For the samples in class C_i, the degree of the neighbor hash value of C_i is defined by calculating the distance between the neighbors as follows:

$$\begin{aligned} cw2 &= \sum_{r_k \in C_i} \phi(h, r_k, r_{N_k}) \\ &= \sum_{r_k \in C_i} \sum_{N_k \in \text{Neighbor}} \left\| y'_{r_k} - y'_{r_{N_k}} \right\|^2 \end{aligned} \tag{6}$$

where N_k denotes the neighbor pixels of r_k.

For the samples in class u_i, the distance between the *l*th samples is designed to measure the within class similarity between the *l*th samples and u_i as follows:

$$cw3 = \sum_{r_l \notin C_i} \phi(h, r_i, u_i) = \sum_{r_l \notin C_i} \left\| y'_{r_l} - y'_{u_i} \right\|^2. \tag{7}$$

C. Learning the Orthogonal Projection Parameter

Followed by the definition of the similarities, the loss function needs to be optimized, in particular, the projection matrix *W* and the threshold *b* are optimized separately. After the substitution with the above formula (3)–(7), the loss function (2) converts to the following equation:

$$\operatorname{loss}(w, b) = \frac{cb1 + cb2}{cw1 + cw2 + cw3}. \tag{8}$$

To find the optimal hash function requires the use of the optimization of the following objective function:

$$(w*, b*) = \arg\max\left(\operatorname{loss}(w, b)\right). \tag{9}$$

According to the definition of the hash function, (5) can be transformed into the following form:

$$\begin{aligned} cw1 &= \sum_{r_k \in C_i} \phi(h, r_k, d_i) \\ &= \sum_{r_k \in C_i} \left\| \operatorname{sgn}(w^T r_k + b) - \operatorname{sgn}(w^T d_i + b) \right\|^2. \end{aligned} \tag{10}$$

It is difficult to directly optimize the above equation, to solve this problem, a spectral relaxation strategy was adopted to remove the symbolic function. Such an approximate optimization method was proved to have better results [24]. After removing the relaxation of the *Sgn* function, (10) converts to the equation

$$\begin{aligned} cw1 &= \sum_{r_k \in C_i} \left\| w^T r_k - w^T d_i \right\|^2 \\ &= \sum_{\mathrm{r_k} \in C_i} w^T (r_k - d_i)(r_k - d_i)^T w. \end{aligned} \tag{11}$$

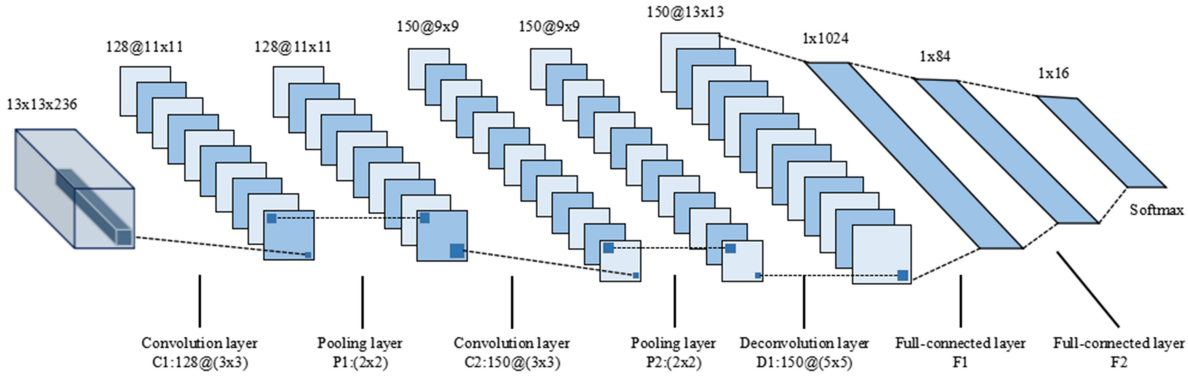


Fig. 2. Hierarchical structure diagram of the convolution neural network.

By letting

$$cw1' = \sum_{r_k \in C_i} (r_k - d_i)(r_k - d_i)^T. \tag{12}$$

The formula above is described as follows:

$$cw1 = w^T cw1' w. \tag{13}$$

Similarly, cw2, cw3, cb1, cb2 can also be represented by expressions as follows:

$$cw2 = w^T cw2' w \tag{14}$$

$$cw3 = w^T cw3' w \tag{15}$$

$$cb1 = w^T cb1' w \tag{16}$$

$$cb2 = w^T cb2' w. \tag{17}$$

In this way, the solution of the objective function (8) converts to the following formula:

$$(w, b) = \arg\max_{w,b} \left(\frac{w^T (cb1' + cb2') w}{w^T (cw1' + cw2' + cw3') w} \right). \tag{18}$$

In the following step, we introduce a Lagrangian multiplier λ to solve (18) that is formulated as follows:

$$\frac{cb1' + cb2'}{cw1' + cw2' + cw3'} w = \lambda w. \tag{19}$$

Because each eigenvector can represent the projection vector of the hash function, estimating an orthogonal matrix of w converts to solve an eigenvalue decomposition problem. In this paper, the optimal w can be computed as the eigenvector corresponding to the largest eigenvalue of the matrix as follows:

$$M = \frac{cb1' + cb2'}{cw1' + cw2' + cw3'}. \tag{20}$$

The value of offset b for each class is usually defined as same as [28]

$$b = \text{mean}(w^T x). \tag{21}$$

In summary, the whole procedure of the hashing semantic feature extraction method is outlined as follows.

Algorithm: Semantic Feature Extraction.

Initial conditions: Hyperspectral data Ω, the total number of classes p.
Preprocess Ω by using the zero mean method, $\Omega \to \Omega'$.
For every class i ($1 <= i <= $ p)
a. Calculate *cb1* and *cb2* according to Eqn. (13) and Eqn. (14).
b. Calculate *cw1*, *cw2*, and *cw3* according to Eqn. (15)-(17).
c. Construct matrix *M* according to Eqn. (20).
d. Calculate the eigenvalue denoted as $\{v\}$ and eigenvector denoted as $\{w\}$ of *M*.
e. Find the eigenvector from $\{w\}$ marked as $\mathrm{w_{max}(i)}$ corresponding to the largest eigenvalue of $\{v\}$.
f. Compute the offset b_i of the *i*th class by

$$\mathrm{b_i} = -\mathrm{mean}\Big(\underset{r_j \in C_i}{\mathrm{r}_j} \times \mathrm{w_{max}}(i)\Big)$$

g. For every pixel $r_j (r_j \in \Omega')$
Extract hash semantic features hf_{jh} by

$$f_{jh} = \mathrm{sgn}(\mathrm{r} \times \mathrm{w_{max}}(i) + b_i)$$

$$hf_{jh} = \begin{cases} (1 + f_{jh})/2 if f_{jh} \geq 0 \\ (1 - f_{jh})/2 if f_{jh} < 0 \end{cases}$$

End
Output: The semantic feature of class *i*, H_i.
End
Output: Feature maps of all the classes $\{H_i\}$ ($1 = i <= $ p).

III. Classification Method Based on the CNN Network

A. Network Structure of the CNN Model

A CNN includes a stack of convolutional layers and hidden layers that has been widely used in a range of computer vision tasks, such as image denoising [29]–[31], image detection, and classification [32]–[34]. In hyperspectral image processing,

CNN architecture makes use of spatial dependency and spectral information via sharing weights and bias of neurons in adjacent layers that can achieve better performance.

In this paper, the proposed CNN network is composed of seven layers, the hierarchical structure diagram of the CNN framework with specific parameters is shown in Fig. 2. Specifically, the model structure of the CNN consists of two convolution layers (C1, C2), two pooling layers (P1, P2), one deconvolution layer (D1), and two full connection layers (F1, F2). First, due to the limited training sample a preprocessing procedure is utilized to expand the selected sample sets, then the feature maps are obtained by a series of hidden layers, furthermore, to overcome the overfitting problem of the network, drop out operation. A deconvolution layer is used to maintain a good performance in this paper. For comparison, a global average pooling (GAP) layer and full connection layers are used separately in the experiment section.

B. Sample Data Augmentation

First, we gather data by augmentation to enrich the training sets of the samples in the preprocessing period and some samples by transformations with respect to the original samples that are generated in this paper. The three types of augmentation are listed as follows.

1) Reverse the sample data from up to down.
2) Reverse the sample data from left to right.
3) Adding random Gaussian noise to the sample.

Furthermore, to avoid the pixels belonging to the borders of the image which cannot be classified properly, in this paper, we adopted the mirroring strategy to preprocess the hyperspectral image. The specific approach implemented mirroring d pixels of border outward according to the size of sample data, for example, the size of sample data is set to 13×13, then we set $d = 5$ to keep the border pixel be the center of the modified sample.

C. CNN Architecture Details

Each convolution layer constantly updates the parameters of the convolution kernels to obtain the useful features. The size of input data is $d \times d \times l$, the number of channels is the same as the bands number which is l, the kernel size of the first convolution layer (c1) is 3×3, the number of kernels is k, then the size of the output feature map is $(d–2) \times (d–2) \times l$. The feature mapping structure adopts the ReLU function $y_{ij}^{xyz} = \max(0, u_{ij}^{xyz})$as the activation function of the convolutional network so that the feature map can be simply understood as putting negative values to "0," meanwhile it causes the sparsity of the CNN network. Next, the max-pooling function is utilized in the pooling layers (p1, p2), which use zero-padding policy to keep the size of each output feature the same as the previous layer. To compress the extracted features, two fully connected layers (FC1, FC2) are adopted in the network, the drop out rate in the first fully connection layer (FC1) is usually set to 0.5, the number of sigmoid neuron nodes is set to 1024 and 84 specifically.

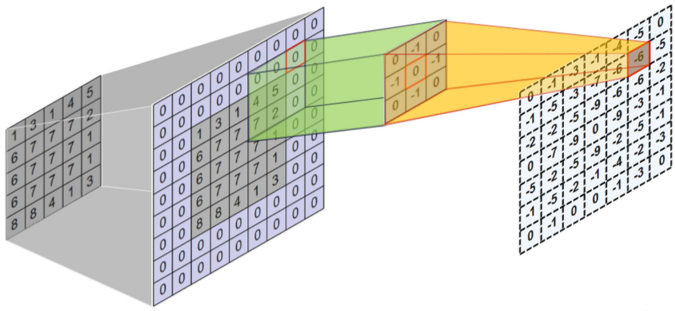


Fig. 3 Illustration of the deconvolution procedure.

D. Deconvolution Layer

In order to keep the size of the output map identical to the original data and enhance the expression of the extracted feature, we utilize deconvolution operation to expand the feature map. A deconvolution operation is also called transposition convolution, which is usually used to map low-dimensional input into a high-dimensional feature and is the inverse to convolution operations. The forward operation of the convolution layer can be expressed as a matrix multiplication in the tensorflow, the output $Y = CX$, where X is the original signal, C is convolution operator, Y is the processed signal after convolution, and the deconvolution layer can be expressed as $X = C^T Y$. When the input vector dimension is lower than the output vector dimension, the neural network is equivalent to a decoder, which realizes the reconstruction of the low-dimensional vector to the high-dimensional vector. Fig. 3 shows the implementation procedure of the deconvolution, it can be observed that deconvolution is an implementation of upsampling to enlarge the feature map with convolution kernel and padding, the size of input data is 5×5, the enlarged feature map is 7×7 after zero-padding and the convolution procedure with kernel size of 3×3. In our paper, the D1 layer has a kernel filter with the size of 5×5, and the length of the convolution filter is 150.

IV. Experiment and Result Analysis

A. Data Description

1) Purdue Indiana Indian Pines Scene: The first dataset is an AVIRIS image which was collected over north-western Indiana. The scene contains 145×145 pixels and 220 spectral bands in the range of 0.4–2.5 μm. It consists of 16 classes available in the ground truth image. Fig. 4(a) shows the ground truth image of Indian Pines and the processed image ($d = 5$) after mirroring projection policy described in Section III and is shown in Fig. 4(b).

2) Salinas Valley: The second dataset was captured by the AVIRIS sensor over the Salinas Valley in Southern California. The image has 224 bands, and the spatial resolution is 512×217. The ground truth contains 16 classes as shown in Fig. 5(a), and the image with mirroring projection processing ($d = 5$) is shown in Fig. 5(b).

3) University of Pavia: The third dataset used in the following experiment is an urban area surrounding the University of

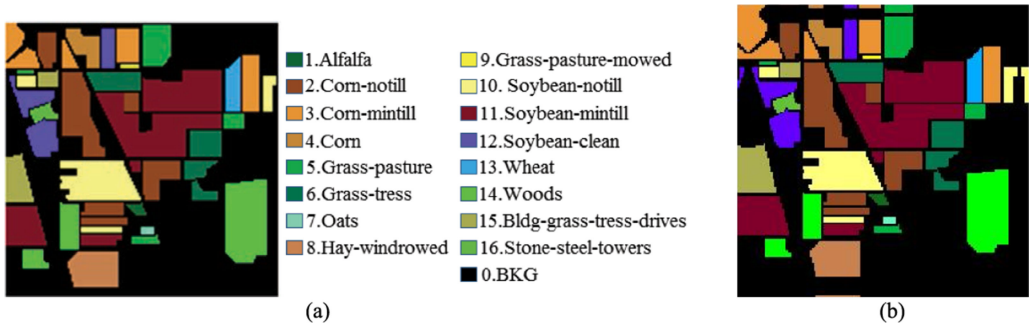


Fig. 4. Image of Purdue Indiana Indian Pines Scene. (a) Ground truth image. (b) Processed image after mirroring projection.

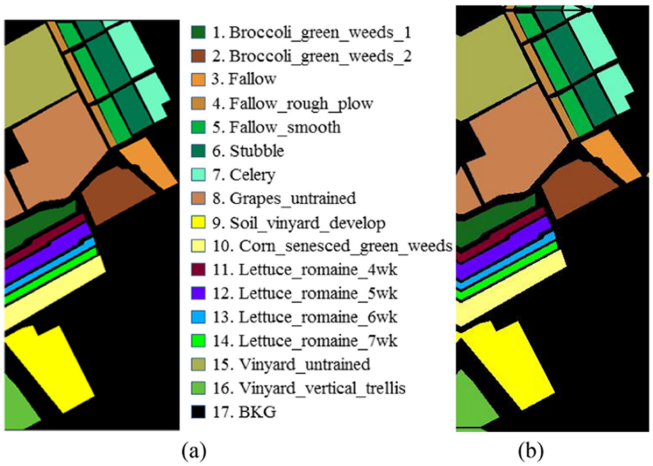


Fig. 5. Image of Salinas Valley. (a) Ground truth image. (b) Processed image after mirroring projection.

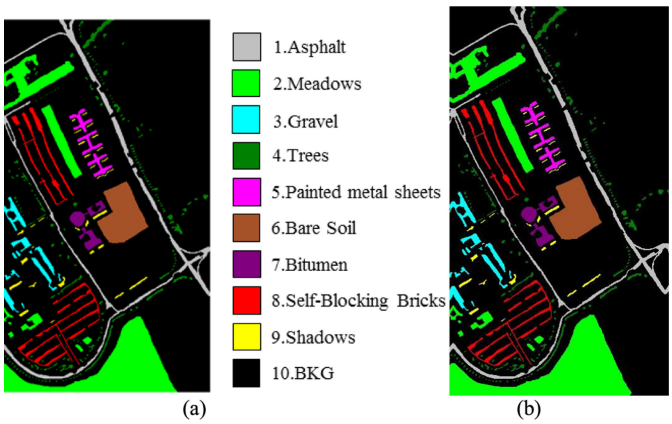


Fig. 6. Image of University of Pavia. (a) Ground truth image. (b) Processed image after mirroring projection.

Pavia, which was recorded by the ROSIS-03 satellite sensor. This image has 103 bands and the pixel resolution is 610×340. Nine classes of interest are presented in this image. Fig. 6(a) shows the ground truth image of University of Pavia and the processed image with mirroring policy ($d = 5$) for training is shown in Fig. 6(b).

4) Kennedy Space Center (KSC): The last dataset is called KSC data, which was acquired by AVIRIS in the range of 0.4–2.5 μm of KSC located in Florida. After removing water absorption and low SNR bands, the dataset has 176 bands used for the analysis. The resolution of the image is 512×614 and it containing 13 classes representing the various land cover types. Fig. 7(a) and (b) show the ground truth image and the processed image ($d = 5$) after mirroring projection policy of KSC, respectively.

B. Experimental Setting

In this section, the hyperspectral image classification methods based on SVM and CNN are utilized to verify the effectiveness of the proposed method. The SVM algorithm is implemented with the libsvm library with 10% of ground truth chosen randomly from the training set in our experiment. The SVM method with the embedded hash semantic feature is denoted as SVMH.

All experiments were run on a computer with Intel Xeon E5-2650 and single Quadro M2000 GPU, 64 GB memory, the OS is Windows 10, the platform is Python 3.6 in a tensorflow framework. In addition, in our experiment, the batch size is set to 100, the iteration number of trainings is 4000, the channel number of Purdue is set to 220 and 236 with hash semantic feature, the channel number of Salinas is set to 224 and 240 with hash semantic feature, the channel number of Pavia is set to 103 and 112 with hash semantic feature, and the channel number of KSC is set to 176 and 189 with hash semantic feature. Training sample numbers of the four datasets used by the CNN networks including samples selected randomly from every class and generated by the augmentation method mentioned in the previous section are shown in Tables I–IV, respectively.

C. Classifiers Descriptor for Comparison

To evaluate the performance of our proposed CNN method using the hashing extraction method, we completed a series of experiments for comparison with other the state-of-the-art classification methods. The network proposed in the paper is called as CNN with deconvolution and hashing method (CNNDH). The CNN methods in the comparison include the original CNN network with single architecture (CNNS) which has only one convolution layer, one pooling layer, and two

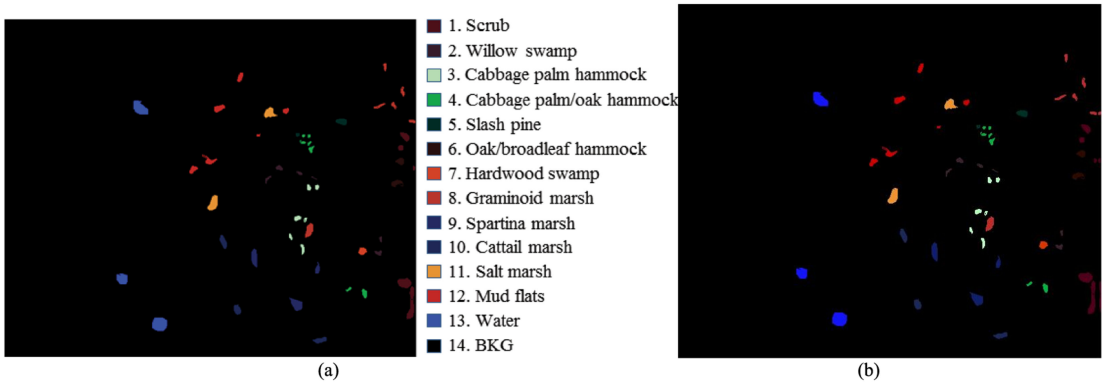


Fig. 7. Image of KSC. (a) Ground truth image. (b) Processed image after mirroring projection.

TABLE I
NUMBER OF SAMPLES IN THE TRAINING SET OF PURDUE INDIANA INDIAN PINES SCENE USED IN CNN METHODS

Class Name	Sample Number			
	Total	Training	Augmentation	Testing
Alfalfa	46	3	9	43
Corn-notill	1428	72	285	1356
Corn-mintill	830	42	166	788
Corn	237	12	47	225
Grass-pasture	483	25	96	458
Grass-trees	730	37	146	693
Grass-pasture-mowed	28	2	5	26
Hay-windrowed	478	24	95	454
Oats	20	1	4	19
Soybean-notill	972	49	194	923
Soybean-mintill	2455	123	491	2332
Soybean-clean	593	30	118	563
Wheat	205	11	41	194
Woods	1265	64	253	1201
Buildings-Grass-Trees-Drives	386	20	77	366
Stone-Steel-Towers	93	5	18	88

TABLE II
NUMBER OF SAMPLES IN THE TRAINING SET OF SALINAS VALLEY USED IN CNN METHODS

Class Name	Sample Number			
	Total	Training	Augmentation	Testing
Broccoli_green_weeds_1	2009	101	401	1908
Broccoli_green_weeds_2	3726	187	745	3539
Fallow	1976	99	395	1877
Fallow_rough_plow	1394	70	270	1324
Fallow_smooth	2678	134	535	2544
Stubble	3959	198	791	3761
Celery	3579	179	715	3400
Grapes_untrained	11271	564	2254	10707
Soil_vinyard_develop	6203	311	1240	5892
Corn_senesced_green_weeds	3278	164	655	3114
Lettuce_romaine_4wk	1068	54	213	1014
Lettuce_romaine_5wk	1927	97	385	1830
Lettuce_romaine_6wk	916	46	183	870
Lettuce_romaine_7wk	1070	54	214	1016
Vinyard_untrained	7268	364	1453	6904
Vinyard_vertical_trellis	1807	91	361	1716

TABLE III
NUMBER OF SAMPLES IN THE TRAINING SET OF UNIVERSITY OF PAVIA USED IN CNN METHODS

Class Name	Sample Number			
	Total	Training	Augmentation	Testing
Asphalt	6631	332	1326	6299
Meadows	18649	933	3729	17716
Gravel	2099	105	419	1994
Trees	3064	154	612	2910
Painted metal sheets	1345	68	269	1277
Bare Soil	5029	252	1005	4777
Bitumen	1330	67	266	1263
Self-Blocking Bricks	3682	185	736	3497
Shadows	947	48	189	899

TABLE IV
NUMBER OF SAMPLES IN THE TRAINING SET OF KENNEDY SPACE CENTER USED IN CNN METHODS

Class Name	Sample Number			
	Total	Training	Augmentation	Testing
Scrub	761	77	152	684
Willow swamp	243	25	48	218
Cabbage palm hammock	256	26	51	230
Cabbage palm/oak hammock	252	26	50	226
Slash pine	161	17	32	144
Oak/broadleaf hammock	229	23	45	206
Hardwood swamp	105	11	21	94
Graminoid marsh	431	44	86	387
Spartina marsh	520	52	104	468
Cattail marsh	404	41	80	363
Salt marsh	419	42	83	377
Mud flats	503	51	100	452
Water	927	93	185	834

fully connected layers, the CNNS network plus one deconvolution layer (CNNSD), the CNNS plus one convolution layer (CNNT), and the CNN network with a GAP layer instead of fully connection layer (CNNG). The CNNDH without hash extraction is noted as CNND. To show the better performance of the hashing semantic feature, the input data of the CNN network includes the original HIS data and the original data mixed with the hashing semantic feature map. The methods of CNNS, CNNSD, and CNNG with hash extraction step are listed as CNNSH, CNNDH, and CNNGH. Fig. 8 shows the specific flowchart of every compared CNN framework separately. All the CNN networks mentioned for comparison have the same parameters as our proposed CNN architecture. Each execution of all the CNN networks have been repeated 5 times and the classification accuracy used in our experiment is averaged by the results. We use overall accuracy (OA) as evaluation criteria to evaluate the CNN method, which is the ratio between hyperspectral pixels that are classified correctly and the number of all the test samples.

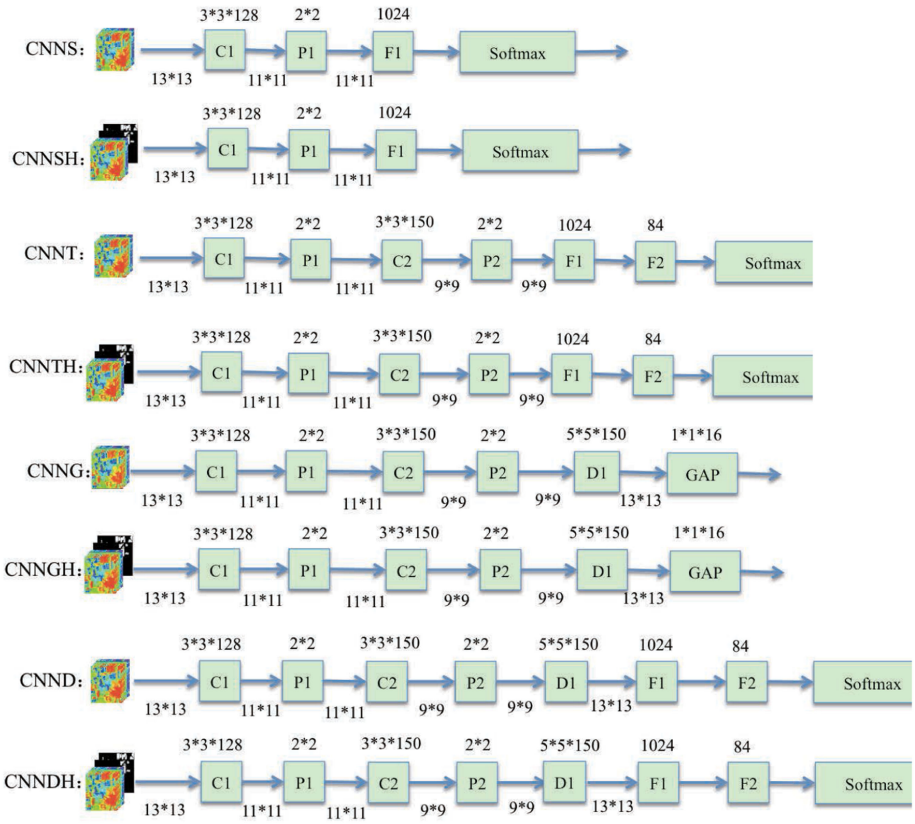


Fig. 8. Flowchart of all the proposed CNN frameworks for comparison.

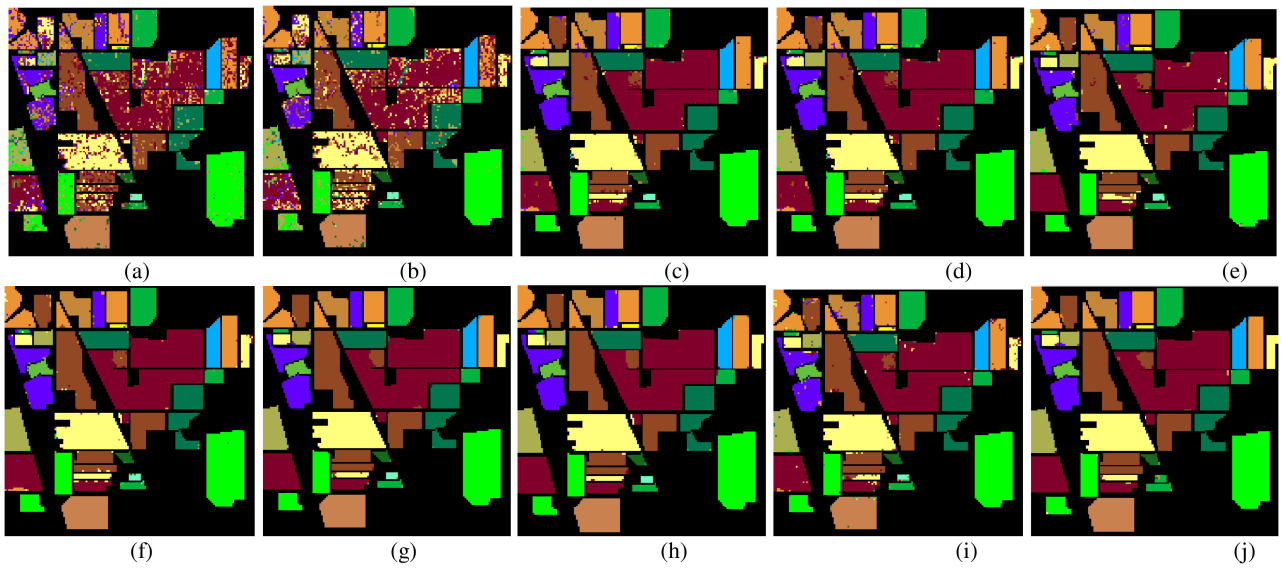


Fig. 9. Classification results of Purdue Indiana Indian Pines Scene with compared methods. (a) SVM. (b) SVMH. (c) CNNS. (d) CNNSH. (e) CNNT. (f) CNNTH. (g) CNNG. (h) CNNGH. (i) CNND. (j) CNNDH.

D. Results and Analysis

For the Purdue Indiana Indian Pines Scene, the training sample percentage is 10%, the drop out rate is set to 0.5, the sample size of the dataset is fixed as 13 × 13. The classification results of the Indian Pines data with the above CNN methods are shown in Fig. 9, Table V shows the accuracy of each class and OA of Indian Pines data objectively. The results show that CNNSH, CNNTH, CNNGH, and CNNDH classification network have better performance than the CNN methods without

TABLE V
OA CALCULATED FROM THE CLASSIFICATION RESULTS OF PURDUE INDIANA INDIAN PINES SCENE WITH ALL THE COMPARED METHODS (10%)

Class P_{OA}%	SVM	SVMH	CNNS	CNNSH	CNNT	CNNTH	CNNG	CNNGH	CNND	CNNDH
1	90.87±4.96	94.35±2.48	72.61±9.67	93.91±2.83	88.7±5.19	96.09±3.57	78.7±41.57	**98.26±1.82**	91.74±7.9	95.65±3.07
2	66.16±4.34	72.24±2.66	88.66±1.99	93.98±2.19	97.06±1.69	97.2±1.6	**98.49±0.55**	98.1±1.91	96.57±1.3	97.77±0.7
3	64.12±3.52	69.42±2.25	92.55±3.6	96.75±1.03	97.04±1.74	97.08±0.81	99.59±0.25	99.28±0.15	96.39±1.19	**99.73±0.1**
4	88.19±3.07	84.73±4.1	84.14±7.03	97.38±0.91	97.97±0.75	98.99±1.14	98.06±1.35	**99.75±0.38**	94.18±1.82	98.57±0.97
5	88.28±2.81	93.25±1.39	92.67±2.54	97.52±1.37	97.52±1.22	98.47±0.73	99.13±0.71	98.3±0.61	97.23±2.02	**99.67±0.11**
6	93.15±1.73	93.67±1.33	97.51±1.55	99.34±0.43	99.26±0.51	99.48±0.11	99.51±0.25	**99.7±0.11**	98.03±0.39	99.23±0.16
7	90.71±4.07	85±6.39	65±8.89	90.71±3.19	86.43±7.74	91.43±3.19	76.43±42.9	**100±0**	80.71±7.82	59.29±54.14
8	97.15±1.32	97.82±1.59	98.7±1.01	99.92±0.11	99.71±0.24	99.75±0.18	99.96±0.09	99.96±0.09	99.58±0.71	**100±0**
9	81±5.48	82±10.37	64±23.02	87±9.75	94±6.52	91±4.18	97±2.74	**100±0**	85±5	98±2.74
10	72.76±4.04	74.09±4.48	89.38±6	95.97±1.35	96.69±1.16	98.09±0.33	98.23±1.6	98.42±1.34	96.01±1.13	**98.85±0.29**
11	64.92±2.74	74.8±3.95	94.37±2.45	97.81±0.46	98.18±0.61	97.67±0.62	99.3±0.12	**99.4±0.21**	98.31±0.75	99.37±0.23
12	67.86±5.02	81.28±3.3	89.31±3.81	96.32±1.17	96.32±1.85	98.15±0.8	**99.93±0.09**	98.65±0.6	94.94±1.17	99.56±0.44
13	98.93±0.87	98.73±0.74	97.17±1.81	99.9±0.22	99.8±0.27	99.32±1.01	**100±0**	99.61±0.41	99.32±0.56	**100±0**
14	88.09±1.98	92.19±2.11	98.47±0.43	99.68±0.06	98.85±1.11	99.73±0.35	99.84±0.19	99.84±0.06	99.4±0.53	**99.91±0.07**
15	68.91±2.45	77.46±3.52	86.89±1.43	99.22±0.48	96.11±2.4	99.43±0.5	99.02±0.34	**99.38±0.35**	95.28±3.1	98.55±0.3
16	98.71±0.9	98.06±1.18	92.04±6.02	98.28±1.44	97.85±2.01	98.49±1.8	**98.71±0.9**	79.14±44.25	92.9±4.34	97.2±0.96
POA	75.31±1.06	80.68±1.4	92.79±1.13	97.39±0.44	97.74±0.58	98.24±0.18	**99.06±0.38**	98.95±0.55	97.28±0.51	99.05±0.2

TABLE VI
OA CALCULATED FROM THE CLASSIFICATION RESULTS OF SALINAS VALLEY WITH ALL THE COMPARED METHODS (4.5%)

Class P_{OA}%	SVM	SVMH	CNNS	CNNSH	CNNT	CNNTH	CNNG	CNNGH	CNND	CNNDH
1	99.21±0.16	99.09±0.36	98.62±2.22	99.81±0.19	94.74±9.97	99.76±0.28	**99.98±0.04**	99.92±0.11	79.98±44.68	79.96±17.87
2	99.65±0.23	99.58±0.28	98.3±1.71	99.84±0.21	99.35±0.75	92.71±16.02	99.95±0.07	**99.99±0.01**	99.92±0.08	99.92±0.13
3	98.94±0.55	99.31±0.24	89.8±6.74	96.76±2.35	94.92±9.57	98.71±2.3	99.29±1.39	95.83±9.24	94.28±6.42	**99.35±0.46**
4	99.45±0.11	99.47±0.13	93.63±1.95	**99±0.9**	98.69±1.25	99.81±0.13	99.67±0.21	99.64±0.29	99.53±0.32	99.28±0.77
5	98.18±0.35	98.1±0.54	94.52±2.88	97.63±2.02	97.99±3.79	98.81±0.9	**99.95±0.08**	99.42±0.46	99.44±0.45	99.01±1.92
6	99.77±0.14	99.77±0.1	99.61±0.41	99.98±0.01	99.85±0.31	99.96±0.07	**100±0**	80±17.89	100±0	99.97±0.05
7	99.65±0.18	99.48±0.15	98.5±1.1	99.96±0.04	99.78±0.2	99.28±0.83	**99.98±0.03**	98.44±3.5	99.66±0.45	99.89±0.1
8	74.74±4.06	74.21±5.64	87.12±4.26	95.74±1.95	95.68±1.98	94.88±2.13	95.79±3.02	**96.99±2.67**	96.51±1.72	96.12±1.56
9	99.3±0.27	98.95±0.4	99.51±0.23	99.94±0.07	99.76±0.09	99.69±0.61	99.97±0.02	**100±0**	99.83±0.12	99.95±0.04
10	93.04±0.73	93.98±1.09	94.81±2.7	99.11±0.81	99.05±0.64	99.45±0.19	59.88±54.67	**99.66±0.17**	98.83±0.77	99.61±0.37
11	97.47±1.13	98.09±0.94	86.57±11.83	98.05±0.6	95.17±8.51	98.65±0.99	99.61±0.72	99.27±0.71	96.52±6.45	**99.78±0.11**
12	99.39±0.27	99.47±0.19	96.26±2.9	98.18±3.89	99.61±0.57	99.96±0.07	79.99±44.72	80±22.36	99.16±1.26	**99.8±0.3**
13	98.62±0.32	98.52±0.34	95.72±2.58	99.76±0.18	99.91±0.09	99.87±0.24	**99.96±0.1**	**99.96±0.06**	99.5±0.53	99.91±0.14
14	97.25±0.95	95.93±1.83	96.97±2.82	99.18±0.57	99.1±0.59	99.44±0.35	59.53±54.35	**99.61±0.34**	99.08±0.85	99.44±0.39
15	70.03±3.59	75.34±3.75	91.86±2.64	94.27±5.74	67.56±34.9	**96.37±2.06**	96.08±2.99	90.53±6.88	92.13±5.86	95.9±3.37
16	98.75±0.47	98.65±0.21	95.04±3.65	99.14±0.63	95.19±8.12	99.35±0.18	**99.77±0.09**	99.27±0.95	98.03±2.08	99.71±0.19
POA	89.79±0.49	90.38±0.98	94.05±1.49	97.85±0.65	93.79±4.27	97.64±1.45	94.59±4.44	95.56±3	96.92±40.98	**97.73±1.52**

hashing extraction. It also can be seen that the SVMH generates better performance than the original SVM method, after the embedding semantic feature, the SVMH significantly improves the OA from 76.91% to 79.41%. It also can be observed that the proposed CNNDH network has the best classification result amongst the other CNN frameworks. The CNNG architecture has the highest OA of 99.06%, and the OA of CNNDH architecture is 99.05%, which is nearly the highest accuracy, and we think the reason is that the accuracy of Oats class is low which is caused by the random sampling. Also, the proposed method generates the best accuracy of class 3, class 5, class 8, class 10, class 13, and class 14, especially the accuracy is 100% of class 8 and class 13.

For the Salinas Valley data the training sample percentage is 4.5%, the drop out rate is set to 0.5, and the sample size of the dataset is fixed as 13 × 13. The classification results with different methods are shown in Fig. 10, Table V shows the accuracy of each class and OA of Salinas Valley. To see the classification results more clearly, we show the zoomed region result of each of the CNN methods by selecting a partiuclar region of interest from Salinas Valley scene. The classification results of class 5 (painted metal sheets) and class 3 (gravel) colored by pink and light blue in the extracted zoomed-in areas in Fig. 6 showed that the CNNDH framework generates the best performance. Objectively, we can see that our CNNDH architecture has the highest OA of 99.73%, and the proposed method generates the best accuracy of class 3, class 11, and class 13 especially. It can be seen that the CNN models with the hashing semantic feature have better performance than the CNN method without embedding the hash feature, which is listed in the 4–11 column of Table VI, whereby, CNNSH improves OA from 94.05% to 97.85%, OA from 93.79% to 97.64% for CNNTH, OA from

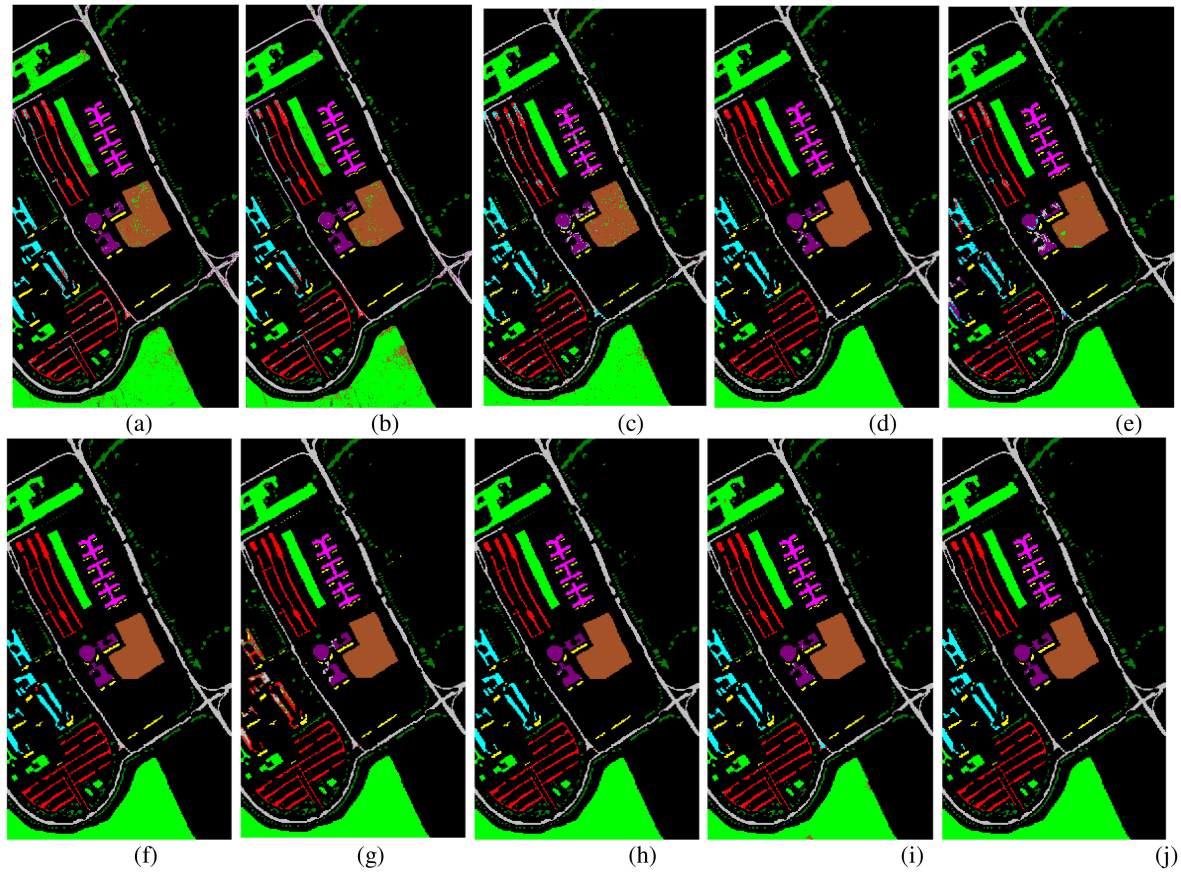


Fig. 10. Classification results of University of Pavia with compared methods. (a) SVM. (b) SVMH. (c) CNNS. (d) CNNSH. (e) CNNT. (f) CNNTH. (g) CNNG. (h) CNNGH. (i) CNND. (j) CNNDH.

TABLE VII
OA CALCULATED FROM THE CLASSIFICATION RESULTS OF UNIVERSITY OF PAVIA WITH ALL THE COMPARED METHODS (5%)

Class P_{OA}%	SVM	SVMH	CNNS	CNNSH	CNNT	CNNTH	CNNG	CNNGH	CNND	CNNDH
1	86.56±1.64	88.38±0.69	91.49±3.18	98.27±1.46	97.62±0.44	99.45±0.36	90.53±17.62	**99.82±0.33**	98.82±0.38	99.79±0.2
2	92.23±1.32	92.09±0.96	97.55±0.81	99.88±0.04	99.44±0.91	99.81±0.28	**99.84±0.21**	99.92±0.09	99.4±0.61	99.96±0.05
3	86.21±0.44	85.37±1.06	74.74±10.62	96.53±1.66	86.84±11.53	98.35±0.56	53.21±49.09	89.27±22.45	93.86±3.97	**99.04±0.59**
4	96.99±0.61	97.74±0.45	89.92±7.7	99.14±0.78	98.17±0.55	**99.7±0.23**	95.1±6.16	98.09±2.93	97.69±1.98	99.62±0.36
5	99.72±0.08	99.9±0.07	90.99±5.46	99.9±0.04	99.75±0.26	**99.99±0.03**	99.02±1.83	99.9±0.1	99.84±0.25	99.91±0.2
6	93.82±0.73	95.35±0.28	88.67±7.17	99.83±0.02	97.7±4.08	99.96±0.04	**99.9±0.1**	99.34±1.15	99.46±0.35	99.71±0.62
7	95.11±0.83	95.05±0.42	71.4±17.1	94.29±1.91	88.59±10.75	**98.38±1.77**	95.61±4.31	97.04±5.76	90.29±6.67	97.4±3.86
8	87.65±1.12	88.31±1.23	89.48±7.63	97.97±1.48	93.94±6.93	99.16±1.17	99.38±0.57	99.53±0.15	98.51±0.8	**99.34±0.49**
9	99.98±0.05	100±0	81.82±11.24	99.66±0.46	97.47±1.8	**99.87±0.14**	72.99±43.39	98.27±1	99.56±0.12	99.66±0.29
OA	91.69±0.72	92.16±0.39	91.84±3.52	99.07±0.26	97.4±1.49	99.6±0.34	94.99±2.9	99.02±1.18	98.58±0.38	**99.69±0.23**

93.79% to 97.64% for CNNGH, and OA from 96.92% to 97.73% for CNNDH.

Fig. 11 illustrates the classification results of the University of Pavia data with the CNN methods. In this experiment, we fix the training sample percentage at 5%, the data size is 13 × 13, and the drop out rate is 0.5. It can be seen that the classification performance of SVMH is better than SVM, CNNSH is better than CNNS, CNNTH is better than CNNT, CNNGH is better than CNNG, CNNDH is better than CNND.

Table VII shows the accuracy of each class and OA of Pavia dataset objectively, from the table, it is observed that our CNNDH architecture has the best OA of 99.69%, and the proposed method shows the best OA of class 2, 3, and 9 especially, which are 99.96%, 99.04%, and 99.69%.

Fig. 12 illustrates the classification results of the KSC data with the CNN methods. In this experiment, we fix the training sample percentage at 10%, the data size is 13 × 13, and the drop out rate is 0.5. It can be observed that the classification performance of CNNGH is better than CNNG and CNNDH is better than CNND. The accuracy of each class and OA of KSC dataset are shown in Table VIII objectively, it can be seen from the table that our CNNDH architecture has the best OA of 95.01%,

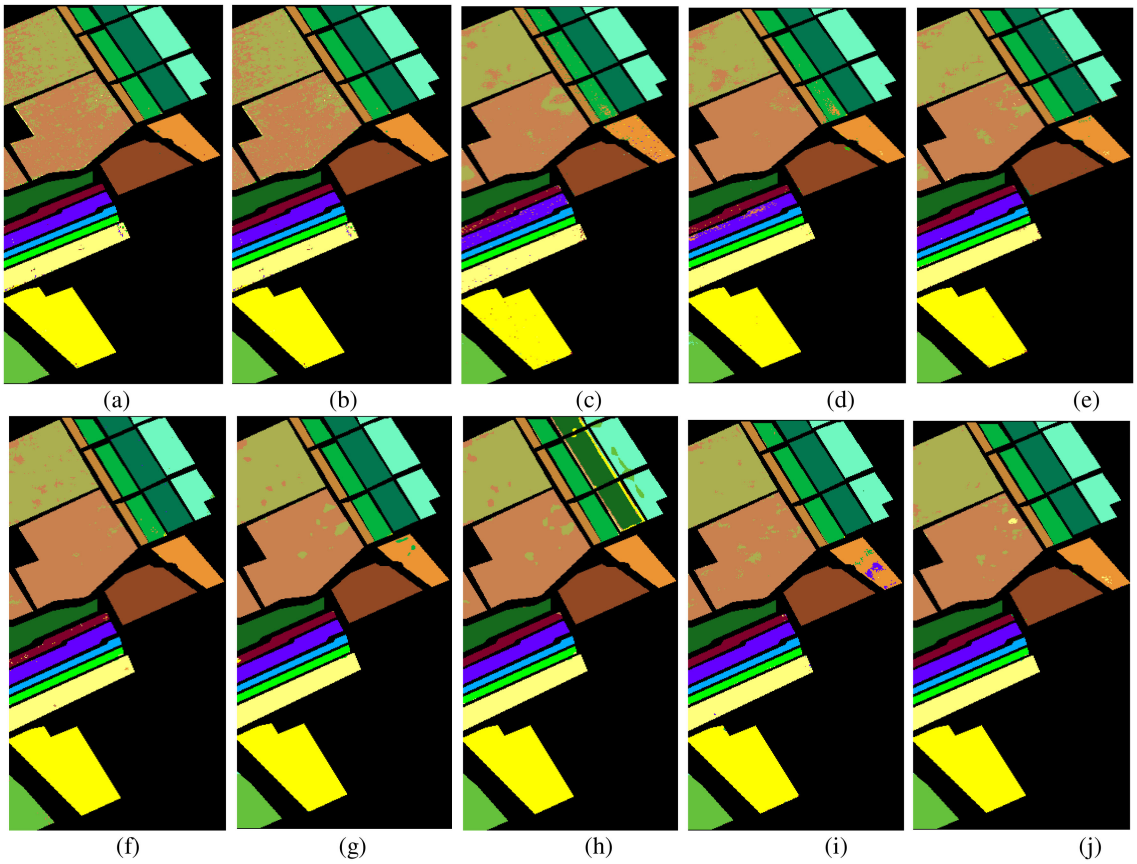


Fig. 11. Classification results of Salinas Valley with compared methods. (a) SVM. (b) SVMH. (c) CNNS. (d) CNNSH. (e) CNNT. (f) CNNTH. (g) CNNG. (h) CNNGH. (i) CNND. (j) CNNDH.

TABLE VIII
OA CALCULATED FROM THE CLASSIFICATION RESULTS OF KSC WITH ALL THE COMPARED METHODS (10%)

Class P_{OA}%	SVM	SVMH	CNNS	CNNSH	CNNT	CNNTH	CNNG	CNNGH	CNND	CNNDH
1	89.41±1.95	88.04±1.95	96.16±2.47	98.53±1.12	98.13±1.6	99.11±0.59	96.56±4.67	99.34±1.25	93.43±4.85	**99.4±0.52**
2	88.48±4.32	84.03±3.98	71.44±4.82	74.65±2.51	83.62±4.98	82.14±9.54	60.99±34.97	85.27±4.82	87.49±3.52	**89.14±4.03**
3	92.97±2.23	89.45±2.96	86.17±2.57	86.56±5.16	91.02±2.52	91.02±2.39	63.2±40.63	**94.61±4.8**	89.3±0.59	93.52±2.11
4	76.59±4.17	74.21±4.95	79.05±2.91	77.86±3.52	85.71±1.09	76.9±8.04	75.87±9.87	72.86±6.55	**86.83±4.81**	83.97±4.55
5	80.37±6.4	69.94±6.39	79.63±2.26	76.77±3.87	84.35±2.03	77.52±4.84	**86.83±1.88**	84.47±5.12	84.35±3.96	82.11±1.19
6	80.61±1.47	66.38±4.83	81.66±2.49	68.21±7.81	86.11±7.92	79.21±9.38	64.54±37.43	76.42±7.06	**84.98±4.12**	84.89±3.5
7	96.38±2.06	95.24±2.94	88.00±3.96	84.76±6.25	99.62±0.85	92.76±5.58	98.86±1.24	87.24±7.21	**99.43±0.85**	93.33±4.9
8	92.9±1.64	88.49±3.37	84.22±4.4	87.19±3.27	94.48±2.99	84.32±3.54	53.23±49.03	91.6±5.34	**95.27±1.01**	88.63±2.18
9	94.58±1.38	95.35±1.62	94.42±1.95	90.85±2.95	94.31±0.86	94.46±1.43	95.58±5.73	92.46±2.29	96.00±1.84	**97.73±0.6**
10	94.16±1.69	89.65±3.03	90.59±4.27	89.16±2.92	93.91±2.09	96.19±1.26	95.84±2.87	93.66±1.57	**96.83±1.47**	96.19±2.27
11	97.76±0.99	98.04±0.52	99.62±0.4	99.38±0.27	99.28±0.29	98.9±0.78	99.24±0.39	**99.81±0.2**	**99.81±0.2**	99.33±0.57
12	93.48±2.29	92.21±3.81	94.35±1.93	95.63±2.92	97.38±2.2	**98.49±0.92**	97.22±2.57	91.81±3.3	97.1±3.91	97.3±1.79
13	99.78±0.11	99.83±0.12	99.94±0.06	99.74±0.33	99.98±0.05	99.96±0.06	**99.98±0.05**	99.85±0.18	99.98±0.05	99.01±2.22
POA	92.48±0.48	90.08±0.83	91.56±0.45	91.12±0.83	94.92±0.59	93.34±1.16	87.75±5.86	93.1±0.79	94.8±1.61	**95.01±0.82**

and the proposed method shows the best OA of class 1, 2, 6, and 9 especially, which are 99.40%, 89.14%, 84.89%, and 97.73%.

The training time of all the versions of CNNs proposed to compare the performance in Section II is shown in Fig. 13. It is observed that the training time is related to the size of the image and the total number of the samples, in particular CNNDH has the longest training time for the Salinas data, while Purdue data has the smallest training time for CNNS model. And it can be observed that the models with hashing codes CNNDH has almost the same time as the ones without semantic hash features of CNND, which means we can have better performance with no more time cost of CNNDH.

From the above experiments, it can be clearly concluded that the classification accuracy has better performance of HSIC with

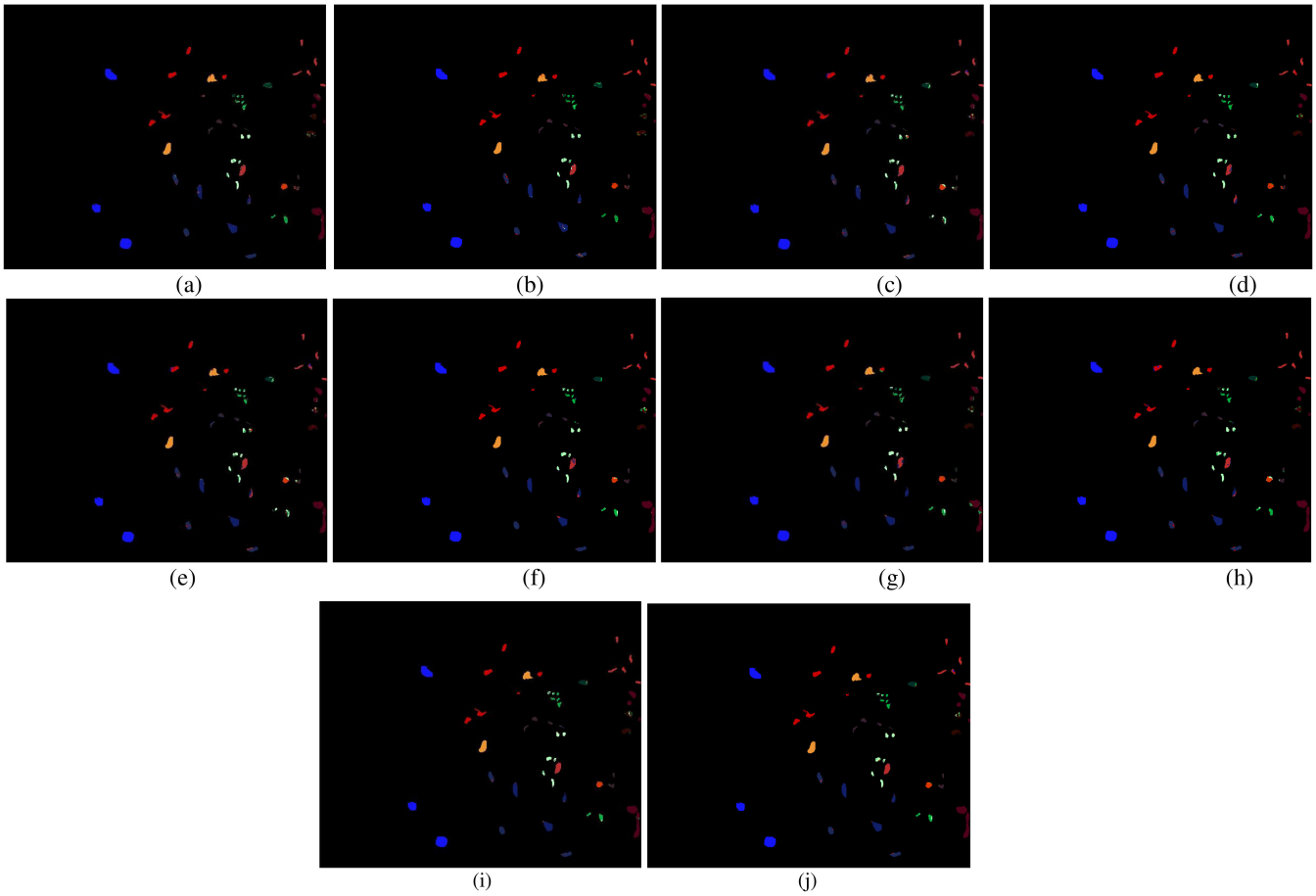


Fig. 12. Classification results of KSC with compared methods. (a) SVM. (b) SVMH. (c) CNNS. (d) CNNSH. (e) CNNT. (f) CNNTH. (g) CNNG. (h) CNNGH. (i) CNND. (j) CNNDH.

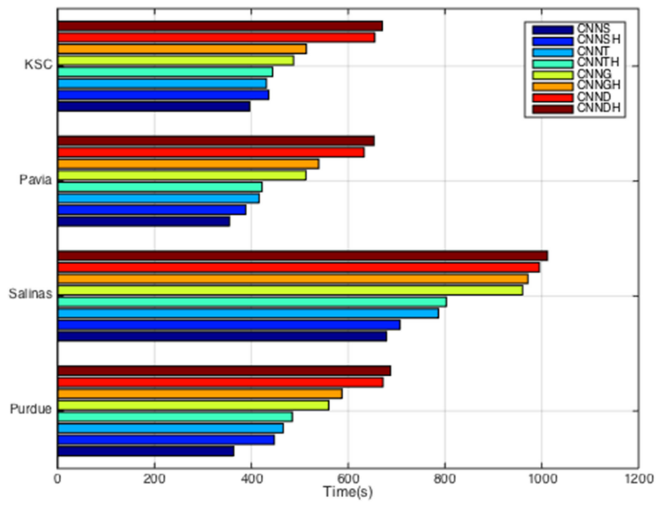


Fig. 13. Training time of the above CNN frameworks.

TABLE IX
OA CALCULATED FROM THE CLASSIFICATION RESULTS OF PURDUE WITH THE COMPARED METHODS

Class P_{OA}%	H^2F[1]	MugNet [2]	CNN[3]	CNNS H	CNNT H	CNNG H	CNND H
1	100.0	100.0	99.56	77.39	87.39	92.17	80.00
2	81.88	78.34	82.57	84.82	91.78	95.10	99.01
3	87.00	89.88	47.18	88.99	90.67	96.94	97.86
4	99.21	99.09	71.43	85.99	90.63	95.27	96.03
5	90.53	93.38	96.28	91.39	94.99	94.33	95.65
6	97.21	99.16	47.36	97.15	98.52	98.41	98.66
7	100.0	99.85	72.85	72.86	73.57	99.29	79.29
8	99.98	99.94	75.45	99.37	99.96	100.00	100.00
9	100.0	100.0	75.35	77.00	83.00	97.00	85.00
10	88.97	90.20	100.00	92.67	95.23	96.67	98.02
11	83.97	85.17	80.47	92.29	95.85	94.61	97.84
12	87.61	95.01	100.00	85.40	91.33	93.76	97.37
13	99.84	99.89	48.53	99.90	99.61	99.90	99.41
14	96.70	98.78	53.90	99.53	99.37	99.60	99.79
15	98.46	99.47	86.55	94.92	96.58	96.42	92.33
16	99.75	99.57	99.06	96.77	98.49	98.71	99.35
POA	89.55	90.65	64.19	92.14	95.20	96.37	**97.93**
AA	94.44	95.48	77.28	89.78	92.54	**96.76**	94.73

all the CNNs framework embedded hash semantic features than the original CNNs, and it can be observed that the CNNDH classification framework has better performance than other CNN methods.

E. Comparison With the State-of-the-Art CNN Architectures

In this part, we compare the proposed model with the other three state-of-the-art CNN models [35]–[37] on the Indiana Purdue dataset. In [35] and [37], the number of the training sample of each class is 20, in [36], only 3 samples per class is chosen

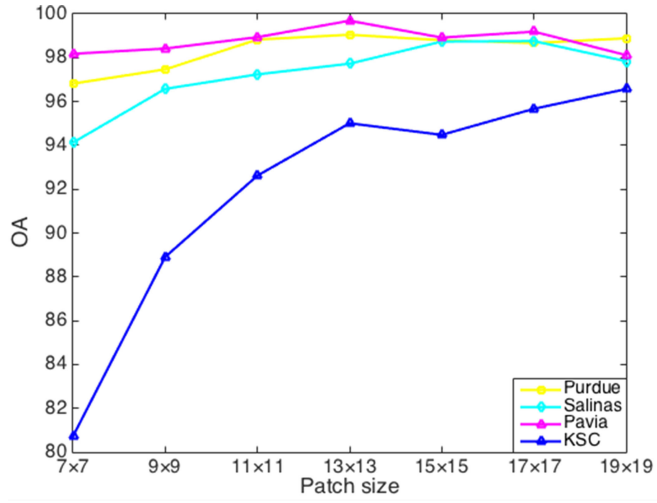


Fig. 14. OA (%) obtained by CNNDH network with the different patch size of training samples on three datasets.

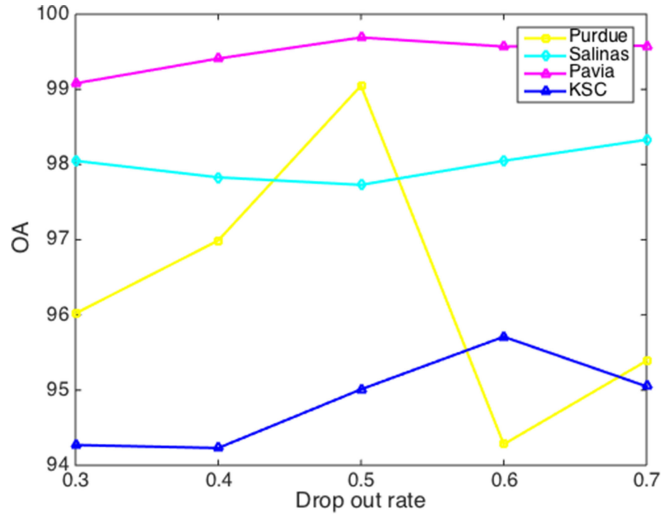


Fig. 15. OA (%) obtained by CNNDH network with different drop out rate of training samples on three datasets.

in the most extreme case, other numbers of sample per class are different from 4 to 15. The sample data size of H2F is 1×1 in [35], Pan *et al*. processed the spectral and spatial separately in Mugnet model [37], the sample size for spectral information is 1×1, and the size for spatial training is 3×3, the sample data size in [36] is 5×5 and the size of convolution kernels is 1×1 in the convolution layer. The sample percentage of this experiment of our CNN networks is 1%, the size of training sample is 13×13 with spatial and spectral information processed simultaneously. The OA results are shown in Table IX, it can be observed than the proposed model has the best performance than the compared CNN networks.

F. Comparison of the Different Size of the Training Sample

Next, we evaluate all the eight CNN networks with different size of sample data. The patch sizes of input data of four data sets are set to 7×7 to 19×19. The percentages of the training set of Purdue Indiana Indian Pines Scene, Salinas Valley, Pavia dataset, and KSC are set to 10%, 4.5%, 5%, and 10%, respectively, the drop out rate is fixed to 0.5 for the three datasets. Fig. 10 shows the OA obtained by our proposed network with the different size of training samples on three datasets, it can be observed that OA varies as the drop out rate changes. Fig. 14 illustrates how the sizes of the training sample influence the effectiveness of the overall performance of relevant image datasets. It can be observed that the size of 13×13 produces the best classification results for the Purdue and the Pavia datasets, the specific OA is 99.05%, and 99.6%. For Salinas, the best OA is 98.76% with the size 17×17, and for KSC, the best OA gets 96.57% when the size of training sample is 19×19. For the four datasets, the OA is 96.82%, 94.13%, 98.18%, 80.73%, when the size is decreased to the patch size of 7×7.

G. Comparison of the Drop Out Rate

By randomly deleting some neurons of the fully connected layer, drop out policy can relieve the over-fitting problem, therefore the effectiveness of the drop out rate is evaluated in this section. The proposed CNNDH networks are compared with different drop out percentage. The size of the training set of Purdue Indiana Indian Pines Scene, Salinas Valley, University of Pavia, and KSC dataset is set to 10%, 4.5%, 5%, and 10%, respectively, the sample size of the three datasets are fixed as 13×13. Fig. 15 shows OA obtained by the CNNDH network with different drop out rates of training samples on three datasets, it can be seen that OA varies as the drop out rate changes. Figs. 10–12 illustrate the comparison of different drop out rate of the three datasets. It can be observed that the drop out rate on 0.5 generates the best result, it leads the best OA of all the three datasets, specific, the OA for the Purdue data is 99.05%, the OA of Salinas data is 98.65%, and for the Pavia data, the OA is 99.69%. When the drop out rate is 0.3, it generates the lowest value, the OA is worst of Purdue data and Pavia data (96.02% and 99.08%, respectively), while when the drop rate is 0.4, it brings the worst OA (97.83%) of the Salinas data, the best OA of KSC data is 95.71% when the drop out rate is 0.6, and the drop out rate of 0.4 brings the worst OA (94.23%).

H. Effect of Different Numbers of Training Samples

In this section, we analyze the evolution of the impact of the different portion of the training sample of the three datasets in the proposed CNN. The labeled pixels for each class are randomly selected as training samples. For the Indian Pines and KSC, different percentages of the training sample are changing from 1% to 10%, and for Salinas image, the change is from 1% to 5%, and for the University of Pavia image, the portion changes from 1% to 5%. The drop out rate of Purdue, Salinas, Pavia, and KSC are all set to 0.5, the sample size of the four datasets are fixed as 13×13. The OA of the different portion of training sample with eight CNN networks of the four datasets is shown in Figs. 16–19, it can be seen that the performances of all methods generally improve as the numbers of training samples increase. Especially, for Purdue Indiana Indian Pines Scene, the best OA is 99.05% when the percentage is 10%, for Salinas Valley, the best OA is 98.65% when the percentage is 4.5%, for Pavia dataset, the OA can reach 99.73% when the percentage is 1%, for KSC data,

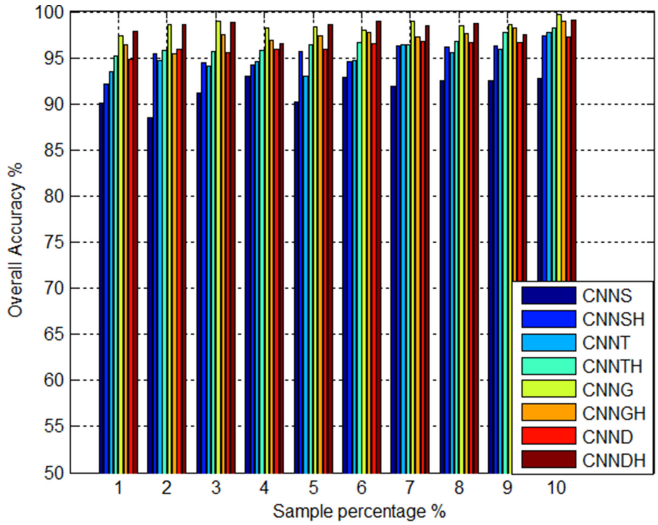


Fig. 16. OA (%) obtained by all CNN networks with different proportions of training samples on Purdue Indiana Indian Pines Scene.

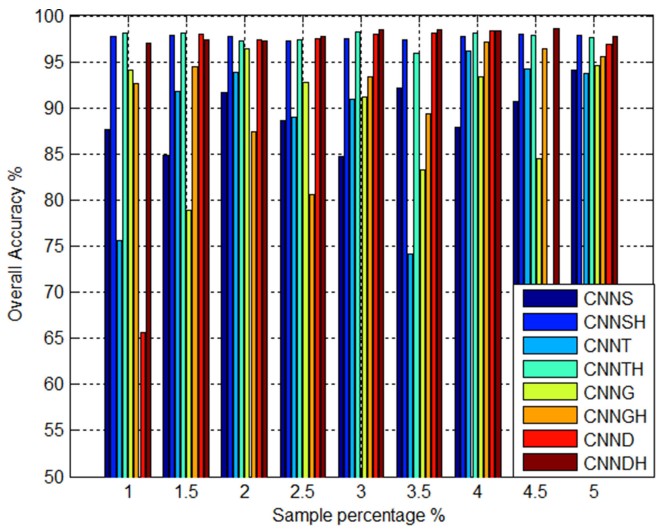


Fig. 17. OA (%) obtained by all CNN networks with different proportions of training samples on Salinas Valley.

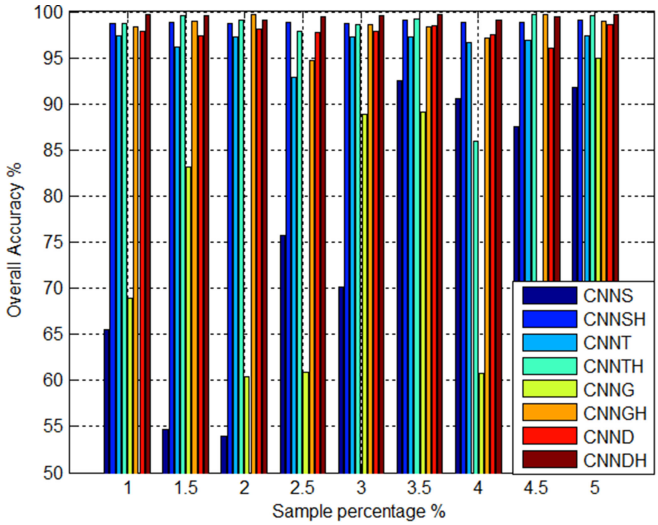


Fig. 18. OA (%) obtained by all CNN networks with different proportions of training samples on University of Pavia.

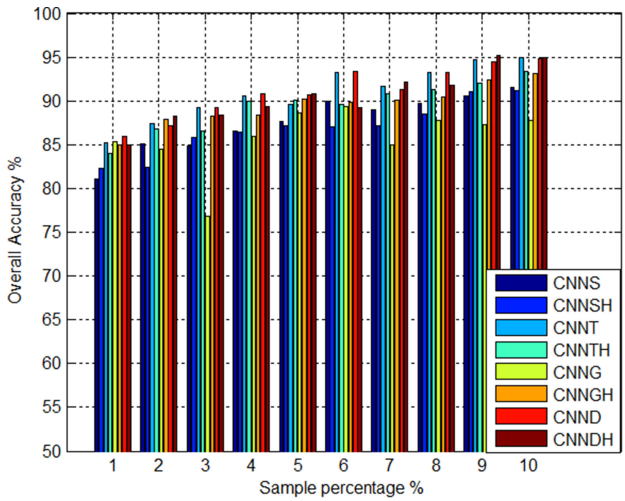


Fig. 19. OA (%) obtained by all CNN networks with different proportions of training samples on KSC.

the OA of 95.24% is much better than others when using 9% labeled samples as training set. The CNNDH networks generate the robust performance even when the training set is very small for the datasets. In addition, we can observe that the proposed CNNDH method show robust improvement over the other CNN architectures with the same number of training samples.

Three differences between the proposed classification framework and the other CNN literature are worth mentioning.

1) In addition to the spectral–spatial information, extracted semantic features cooperated jointly can significantly improve the classification accuracy in the proposed CNN framework.
2) Semantic feature extraction for hyperspectral images classification focused on segmentation method and transformation analysis. In stead of region-based or superpixel segmentation profile or PCA profile, in this paper, we extracted hashing code as salient significance by hashing learning constrained by Fisher-like regularization term.
3) GAP layer is adopted for regularization of the entire network structure to prevent the overfitting problem, however, in this paper, due to the fact that hashing code provide semantic information, the CNN network with the fully connected layers generates more stable convolutional feature including semantic information has better performance compared to the GAP layer included framework.

V. Conclusion

In this paper, a CNN hyperspectral image classification method with hash semantic feature extraction method was proposed, which can encode the hyperspectral data to obtain the semantic feature by hashing learning with the defined discriminative similarity constraints. Meanwhile, we designed a simpler

CNN to classify the HSI embedded with semantic feature map, in which a deconvolution layer is contained to enhance the description of the deep feature map. The real hypers image experimental results illustrate that the proposed CNN method achieves the best image classification performance of the four popular testing datasets. Also, the CNN frameworks embedding of the extracted hash features show higher performance of classification than the CNNs without hashing features. In the future work, we plan to perform band selection to choose the salient bands before fusion with the extracted hashing codes to save storage and decrease training time. Moreover, taking advantage of hierarchical semantic feature to improve better performance for HSI classification remains a problem to be investigated, therefore, our further work will focus on extracting contextual features of hashing semantic feature hierarchically.

REFERENCES

[1] A. Brown *et al.*, "Hydrothermal formation of clay-carbonate alteration assemblages in the nili fossae region of mars," *Earth Planet. Sci. Lett.*, vol. 297, pp. 174–182, 2010.

[2] A J. Brown, B. Sutter, and S. Dunagan, "The MARTE imaging spectrometer experiment: Design and analysis," *Astrobiology*, vol. 8, no. 5, pp. 1001–1011, 2008.

[3] W. Deng, J. Xu, and H. Zhao, "An improved ant colony optimization algorithm based on hybrid strategies for scheduling problem," *IEEE Access*, vol. 7, pp. 20281–20292, 2019.

[4] W. Deng, R. Yao, H. Zhao, X. Yang, and G. Li, "A novel intelligent diagnosis method using optimal LS-SVM with improved PSO algorithm," *Soft Comput.*, vol. 23, pp. 2445–2462, 2017.

[5] 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.

[6] X. Lu, X. Zheng, and Y. Yuan, "Remote sensing scene classification by unsupervised representation learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 9, pp. 5148–5157, Sep. 2017.

[7] S. Patra, P. Modi, and L. Bruzzone, "Hyperspectral band selection based on rough set," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 10, pp. 5495–5503, Oct. 2015.

[8] W. Zhao and S. Du, "Spectral–spatial feature extraction for hyperspectral image classification: A dimension reduction and deep learning approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 8, pp. 4544–4554, Aug. 2016.

[9] Z. Ye, L. Tan, and L. Bai, "Hyperspectral image classification based on spectral-spatial feature extraction," in *Proc. Int. Workshop Remote Sens. Intell. Process*., 2017, pp. 1–4.

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

[11] 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

[12] C. Yu *et al.*, "Multi-class constrained background suppression approach to hyperspectral image classification," in *Proc. IEEE/GRSS Int. Geosci. Remote Sens. Symp.*, Fort Worth, TX, USA, Jul. 2017, pp. 3357–3360.

[13] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[14] X. Ma, H. Wang, and J. Geng, "Spectral–spatial classification of hyperspectral image based on deep auto-encoder," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 9, pp. 4073–4085, Sep. 2016.

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

[16] W. Song *et al.*, "Hyperspectral image classification with deep feature fusion network," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 6, pp. 3173–3184, Jun. 2018.

[17] Y. Li, W. Xie, and H. Li, "Hyperspectral image reconstruction by deep convolutional neural network for classification," *Pattern Recognit.*, vol. 63, pp. 371–383, 2017.

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

[19] C. Li *et al.*, "Hyperspectral image classification with robust sparse representation," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 5, pp. 641–645, May 2016.

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

[21] X. Huang and L. Zhang, "An SVM ensemble approach combining spectral, structural, and semantic features for the classification of high-resolution remotely sensed imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 1, pp. 257–272, Jan. 2013.

[22] L. Zhang *et al.*, "Tensor discriminative locality alignment for hyperspectral image spectral–spatial feature extraction," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 1, 242–256, Jan. 2013.

[23] Z. Ye, L. Tan, and L. Bai, "Hyperspectral image classification based on spectral-spatial feature extraction," in *Proc. Int. Workshop Remote Sens. Intell. Process.*, 2017, pp. 1–4.

[24] R. K. Thika and S. N. S. Lakshmi, "Object detection and semantic segmentation using neural networks," *Int. J. Emerg. Trends Technol. Comput. Sci.*, vol. 47, no. 2, pp. 95–100, 2017.

[25] X. Lu, Y. Chen, and X. Li, "Hierarchical recurrent neural hashing for image retrieval with hierarchical convolutional features," *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 106–120, Jan. 2018.

[26] J. Wang, H. T. Shen, J. Song, and J. Ji, "Hashing for similarity search: A survey," 2014, arXiv preprint arXiv:1408.2927.

[27] M. Datar *et al.*, "Locality-sensitive hashing scheme based on p-stable distributions," in *Proc. 20th Symp. Comput. Geometry*, 2004, pp. 253–262.

[28] J. Wang, S. Kumar, and S. F. Chang, "Sequential projection learning for hashing with compact codes," in *Proc. Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 1127–1134.

[29] K. Zhang *et al.*, "Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.

[30] K. Zhang, W. Zuo, and L. Zhang, "FFDNet: Toward a fast and flexible solution for CNN based image denoising," *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4608–4622, Sep. 2018.

[31] C. Cruz *et al.*, "Nonlocality-reinforced convolutional neural networks for image denoising," *IEEE Signal Process. Lett.*, vol. 25, no. 8, pp. 1216–1220, Aug. 2018.

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

[33] Y. Xu *et al.*, "Spectral-spatial unified networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 10, pp. 5893–5909, Oct. 2018.

[34] L. Mou, P. Ghamisi, and X. Zhu, "Deep recurrent neural networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3639–3655, Jul. 2017.

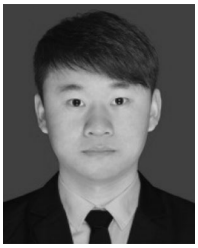
[35] B. Pan *et al.*, "Hashing based hierarchical feature representation for hyperspectral imagery classification," *Remote Sens.*, vol. 9, no. 11, 2017, Art. no. 1094.

[36] S. Yu, S. Jia, and C. Xu, "Convolutional neural networks for hyperspectral image classification," *Neurocomputing*, vol. 219, pp. 88–98, 2017.

[37] B. Pan, Z. Shi, and X. Xu, "MugNet: Deep learning for hyperspectral image classification using limited samples," *ISPRS J. Photogramm. Remote Sens.*, vol. 145, pp. 108–119, 2018.

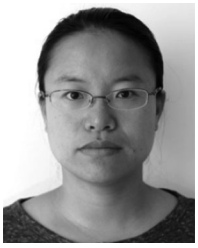
Chunyan Yu (M'17) received the B.Sc. and Ph.D. degrees in environment engineering from Dalian Maritime University, Dalian, China, in 2004 and 2012, respectively.

In 2004, she joined the College of Computer Science and Technology, Dalian Maritime University. From June 2013 to June 2016, she was a Postdoctoral Fellow with the Information Science and Technology College, Dalian Maritime University. From September 2014 to September 2015, she was a visiting scholar with the College of Physicians and Surgeons, Columbia University, New York, NY, USA. She is currently an Associate Professor with the Information Science and Technology College, Dalian Maritime University. Her research interests include image segmentation, hyperspectral image classification, and pattern recognition.

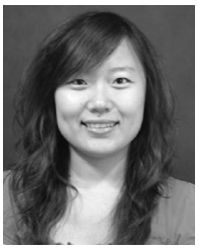


Meng Zhao was born in Shandong, China, in 1993. He is currently working toward the master's degree at the Information Science and Technology College, Dalian Maritime University, Dalian, China.

His research focuses on hyperspectral image classification.



Meiping Song received the Ph.D. degree from the College of Computer Science and Technology, Harbin Engineering University, Harbin, China, in 2006. From 2013 to 2014, she was a Visiting Associate Research Scholar with the University of Maryland, Baltimore County. She is currently an Associate Professor with the College of Information Science and Technology, Dalian Maritime University, Dalian, China. Her research interests include remote sensing and hyperspectral image processing.



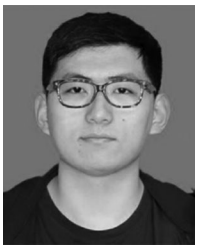
Yulei Wang received the B.S. and Ph.D. degrees in signal and information processing from Harbin Engineering University, Harbin, China, in 2009 and 2015, respectively.

She was a joint Ph.D. student with Remote Sensing Signal and Image Processing Laboratory, University of Maryland, Baltimore County. She is currently a Lecture with the College of Information Science and Technology, Dalian Maritime University, Dalian, China.

Her current research interests include hyperspectral image processing and vital signs signal processing.

Fang Li received the B.E. degree from the College of Information Science and Technology, Qufu Normal University, Rizhao, China, in 2017. She is currently working toward the M.A. degree in computer science and technology at the Dalian Maritime University, Dalian, China.

Her research interests include remote sensing and hyperspectral image processing.



Rui Han received the B.E. degree in software engineering from Ludong University, Yantai, China, in 2018. He is currently working toward the M.A. degree in software engineering at the Dalian Maritime University, Dalian, China.

His research interests include hyperspectral image processing and deep learning.

Chein-I Chang (S'81–M'87–SM'92–F'10–LF'17) received the B.S. degree in mathematics from Soochow University, Taipei, Taiwan, the M.S. degree in mathematics from the Institute of Mathematics, National Tsing Hua University, Hsinchu, Taiwan, the M.A. degree in mathematics from the State University of New York at Stony Brook, Stony Brook, NY, USA, the M.S. and M.S.E.E. degrees from the University of Illinois at Urbana–Champaign, Urbana, IL, USA, and the Ph.D. degree in electrical engineering from the University of Maryland, College Park, MD, USA.

Since 1987, he has been with the University of Maryland, Baltimore County, Baltimore, MD, USA, where he is currently a Professor with the Department of Computer Science and Electrical Engineering. He has been holding the Chang Jiang Scholar Chair Professorship and has been the Director of Center for Hyperspectral Imaging in Remote Sensing, Dalian Maritime University, Dalian, China, since 2016. He has been holding the Hua Shan Scholar Chair Professorship with Xidian University, Xian, China, since 2016, and has been the Feng-Tai Chair Professor with the National Yunlin University of Science and Technology, Douliu, Taiwan, since 2017. In addition, he has also been the Chair Professor of Providence University, Taichung, Taiwan, since 2012. He was a Visiting Research Specialist with the Institute of Information Engineering, National Cheng Kung University, Tainan, Taiwan, from 1994 to 1995, and also a Distinguished Visiting Fellow/Fellow Professor, both of which were sponsored by National Science Council, Taiwan, R.O.C., from 2009 to 2010. Also, he was a Distinguished Lecturer Chair with the National Chung Hsing University, Taichung, Taiwan, sponsored by the Ministry of Education in Taiwan, R.O.C., from 2005 to 2006, and a Distinguished Chair Professor with National Chung Hsing University from 2014 to 2017. He has seven patents on hyperspectral image processing. He has authored four books *Hyperspectral Imaging: Techniques for Spectral Detection and Classification* (Kluwer, 2003) and *Hyperspectral Data Processing: Algorithm Design and Analysis* (Wiley, 2013), *Real Time Progressive Hyperspectral Image Processing: Endmember Finding and Anomaly Detection* (Springer, 2016), and *Recursive Hyperspectral Sample and Band Processing: Algorithm Architecture and Implementation* (Springer, 2017). In addition, he has also edited two books *Recent Advances in Hyperspectral Signal and Image Processing* (Research Signpost, 2006) and *Hyperspectral Data Exploitation: Theory and Applications* (Wiley, 2007), and co-edited a book *High Performance Computing in Remote Sensing* (CRC Press, 2007) with A. Plaza. His research interests include multispectral/hyperspectral image processing, automatic target recognition, and medical imaging.

Dr. Chang was a plenary speaker for the Society for Photo-optical Instrumentation Engineers (SPIE) Optics+Applications, Remote Sensing Symposium in 2009. He was also a keynote speaker at the User Conference of Hyperspectral Imaging in December 30, 2010, Industrial Technology Research Institute, Hsinchu; the 2009 Annual Meeting of the Radiological Society of the Republic of China, Taichung; the 2008 International Symposium on Spectral Sensing Research; and the Conference on Computer Vision, Graphics, and Image Processing in 2003, Kimen and in 2013, Nan-Tou, Taiwan. He was the Guest Editor of a special issue of the *Journal of High Speed Networks on Telemedicine and Applications* in April 2000 and the Co-Guest Editor of another special issue of the same journal on Broadband Multimedia Sensor Networks in Healthcare Applications in April 2007. He is also the Co-Guest Editor of special issues on High Performance Computing of Hyperspectral Imaging for the *International Journal of High Performance Computing Applications* in December 2007, Signal Processing and System Design in Health Care Applications for the EURASIP *Journal on Advances in Signal Processing* in 2009, Multipsectral, Hyperspectral, and Polarimetric Imaging Technology for the *Journal of Sensors*, and Hyperspectral Imaging and Applications for *Remote Sensing* in 2017. He is a Fellow of SPIE. He was an Associate Editor in the area of hyperspectral signal processing for the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING from 2001 to 2007. He is currently an Associate Editor for *Artificial Intelligence Research* and is also on the Editorial Boards of the *Journal of High Speed Networks, Recent Patents on Mechanical Engineering, International Journal of Computational Sciences and Engineering, Journal of Robotics and Artificial Intelligence*, and *Open Remote Sensing Journal*. He was the recipient of the National Research Council Senior Research Associateship Award from 2002 to 2003 sponsored by the U.S. Army Soldier and Biological Chemical Command, Edgewood Chemical and Biological Center, Aberdeen Proving Ground, Maryland.