PROCEEDINGS OF SPIE

SPIEDigitalLibrary.org/conference-proceedings-of-spie

Iterative constrained energy minimization convolutional neural network for hyperspectral image classification

Bai Xue, Xiaodi Shang, Shengwei Zhong, Peter F. Hu, Chein-I Chang

Bai Xue, Xiaodi Shang, Shengwei Zhong, Peter F. Hu, Chein-I Chang, "Iterative constrained energy minimization convolutional neural network for hyperspectral image classification," Proc. SPIE 10986, Algorithms, Technologies, and Applications for Multispectral and Hyperspectral Imagery XXV, 109861L (14 May 2019); doi: 10.1117/12.2519046

Event: SPIE Defense + Commercial Sensing, 2019, Baltimore, Maryland, United States

Iterative Constrained Energy Minimization Convolutional Neural Network for Hyperspectral Image Classification

Bai Xue*[a], Xiaodi Shang[b], Shengwei Zhong[c], Peter F. Hu[d], and Chein-I Chang[a,b]
[a]Department of Computer Science and Electrical Engineering, University of Maryland Baltimore County (UMBC), 1000 Hilltop Circle, Baltimore, MD 21250; [b]Center of Hyperspectral Imaging in Remote Sensing (CHIRS), Information and Technology College, Dalian Maritime University, Dalian 116026, China; [c]Department of Information, Engineering, Harbin Institute of Technology, Harbin 150001, China; [d]Department of Anesthesiology, University of Maryland School of Medicine, Baltimore, MD 21201

ABSTRACT

In hyperspectral image classification, how to jointly take care of spectral and spatial information received considerable interest lately, and many spectral-spatial classification approaches have been proposed. Unlike spectral-spatial classifications which are developed from traditional aspect, iterative constrained energy minimization (ICEM) and iterative target-constrained interference-minimized classifier (ITCIMC) approaches are developed from subpixel detection and mixed pixel classification point of view, and generally performs better than existing spectral-spatial approaches in terms of several measurements, such as accuracy rate and precision rate. Recently, convolutional neural networks (CNNs) have been successfully applied to visual imagery classification and have received great attention in hyperspectral image classification, due to the outstanding ability of CNN to capture spatial information. This paper extends ICEM to iterative constrained energy minimization convolution neural network approach for hyperspectral image classification. In order to capture spatial information, instead of Gaussian filter, CNN is utilized to generate binary pixelwise classification map for constrained energy minimization (CEM) detection results, and CNN classification map is feedbacked into hyperspectral bands, and then CEM detection is reprocessed in an iteration manner. Since CNN can reduce the performance of precision rate, a background recovery procedure is designed, to recover background detection map from CEM detection map and add it into CEM result as a new detection map.

Keywords: Hyperspectral image classification, iterative constrained energy minimization (ICEM), convolutional neural network (CNN), band selection and nonlinear expansion (BSNE).

1. INTRODUCTION

The spectral-spatial hyperspectral image classification has been received considerable interests in recent years, due to the advantage of hyperspectral imaging techniques which enable to capture the spectral signature of material substances together with the spatial image information, and many approaches has been proposed [1]. The general idea of a spectral-spatial classifier is that first perform a pixel-based spectral classification, then it is followed by spatial technique in order to capture spatial information and enhance the classification performance from the spectral classification results. However, such a spectral-spatial classification still has some issues need to be address. First, the commonly used spectral classification techniques are pure pixel-based approaches, and they do not take advantage from mixing properties of hyperspectral image. Second, how much spatial information is added is not determined. Finally, the performance analysis of most of the spectral-spatial classification works do not analysis background issue. In order to address aforementioned issues, several approaches such as iterative constrained energy minimization (ICEM) [2] and iterative target-constrained interference-minimized classifier (ITCMIC) [3] have been proposed, and such approaches utilize spatial filter such as Gaussian filter to capture spatial information from the results of hyperspectral target detection approaches, and feed the results back to original band images. The constrained energy minimization (CEM) [4] and target-constrained interference-minimized classifier (TCIMC) [5] are utilized in ICEM and ITCIMC, and both approaches are designed from subpixel detection point of view. The amont of spatial information which added into classification can be determined by the iteration and its termination rule. And also, instead of performing multi-class classification, both approaches are detecting each class as target from background. As a result, not only pixel can be determined whether it belongs to the class or belongs to the background. However, in order for traditional classification

Algorithms, Technologies, and Applications for Multispectral and Hyperspectral Imagery XXV,
edited by Miguel Velez-Reyes, David W. Messinger, Proc. of SPIE Vol. 10986, 109861L
 · doi: 10.1117/12.2519046

techniques such as support vector machine (SVM) [6] to determine background, the priori knowledge is necessarily required. Due to the background is complex, and lack of labeled background data, the background priori knowledge is hardly to obtain. This paper takes advantages of ICEM and extend it to iterative constrained energy minimization convolutional neural network (ICEM-CNN). Unlike ICEM utilizes Gaussian filter to capture spatial texture information, in this paper the convolutional neural network (CNN) [7] is utilized to capture spatial information for the proposed method by generating binary pixelwise classification map from constrained energy minimization (CEM) detection results, and feed CNN classification maps back into original band images. In addition, since CNN cannot determine whether pixel belongs to the class or background and reduce the performance of precision rate, a background recovery procedure is designed, and it is implemented before CNN to obtain background information from binary CEM classification results.

2. BSNE-ICEM

2.1 Band selection and nonlinear expansion

Band selection and nonlinear expansion (BSNE) is a preprocess stage before ICEM is implementation, which consists of band selection (BS) and then nonlinear bands expansion (NE). The aim of utilize BS is to avoid the redundant bands to be used for classification, due to hyperspectral image is acquired with highly correlated spectral bands information. The uniform band selection (UBS) is chosen for BS, since extensive experiments in the literature has shown that UBS is a reasonably good technique [8].

The NE aims to expand the original set of hyperspectral band image with nonlinear functions, such as autocorrelation, cross correlation, et al. Such NE technique is referred to as correlation band expansion process (CBEP), and the detailed implementation of CBEP is provided as following.

Correlation Band Expansion Process (CBEP)

Step 1. 1st-order band image: $\{\mathbf{B}_l\}_{l=1}^{L}$ = set of original band images

Step 2. 2nd-order correlated band images:

(i) $\{\mathbf{B}_l^2\}_{l=1}^{L}$ = set of auto-correlated band images

(ii) $\{\mathbf{B}_k\mathbf{B}_l\}_{k=1,l=1,k\neq l}^{L,L}$ = set of cross-correlated band images

In case, a re-scaling is needed, auto- or cross-correlated band images can be normalized by the variances of band images such as $\left(\sigma_{\mathbf{B}_l}^2\right)^{-1}\{\mathbf{B}_l^2\}$ and $\left(\sigma_{\mathbf{B}_k}\sigma_{\mathbf{B}_l}\right)^{-1}\{\mathbf{B}_k\mathbf{B}_l\}$.

Step 3. 3rd order correlated band images

(i) $\{\mathbf{B}_l^3\}_{l=1}^{L}$ = set of auto-correlated band images

(ii) $\{\mathbf{B}_k^2\mathbf{B}_l\}_{k=1,l=1,l\neq k}^{L,L}$ = set of two cross-correlated band images

(ii) $\{\mathbf{B}_k\mathbf{B}_l\mathbf{B}_m\}_{k=1,l=1,m=1,k\neq l\neq m}^{L,L,L}$ = set of three cross-correlated band images

Similarly, like step 2, if a re-scaling is needed, auto- or cross-correlated band images can be normalized by the variances of band images such as $\left(\sigma_{\mathbf{B}_l}^3\right)^{-1}\{\mathbf{B}_l^3\}$, $\left(\sigma_{\mathbf{B}_k}^2\sigma_{\mathbf{B}_l}\right)^{-1}\{\mathbf{B}_k^2\mathbf{B}_l\}$ and $\left(\sigma_{\mathbf{B}_k}\sigma_{\mathbf{B}_l}\sigma_{\mathbf{B}_m}\right)^{-1}\{\mathbf{B}_k\mathbf{B}_l\mathbf{B}_m\}$.

Step 4. Nonlinear correlated band images

(i) $\{\sqrt{\mathbf{B}_l}\}_{l=1}^{L}$ = set of band images stretched out by the square-root.

(ii) $\{\log(\mathbf{B}_l)\}_{l=1}^{L}$ = set of band images stretched out by the logarithmic function.

2.2 Iterative CEM

Suppose that a hyperspectral image is represented by a collection of image pixel vectors, denoted by $\{\mathbf{r}_1, \mathbf{r}_2, \cdots, \mathbf{r}_N\}$ where $\mathbf{r}_i = (r_{i1}, r_{i2}, \cdots, r_{iL})^T$ for $1 \le i \le N$ is an L-dimensional pixel vector, N is the total number of pixels in the image and L is the total number of spectral channels. Further assume that $\mathbf{d}$ is specified by a desired signature of interest

to be used for target detection. The output energy of a finite impulse response (FIR) filter denoted by $\mathbf{w} = \left(w_1, w_2, \cdots, w_L\right)^T$ is minimized and constraint $\mathbf{d}^T\mathbf{w} = 1$. This can be expressed as the following.

$$\min_{\mathbf{w}} \left\{\mathbf{w}^T\mathbf{R}\mathbf{w}\right\} \text{ subject to } \mathbf{d}^T\mathbf{w} = \mathbf{w}^T\mathbf{d} = 1. \tag{1}$$

where $\mathbf{R} = \left(1/N\right)\sum_{i=1}^{N}\mathbf{r}_i\mathbf{r}_i^T$ is the auto-correlation sample matrix of the hyperspectral image.

So, the optimal solution to (1) is given by as follows.

$$\mathbf{w}^{\text{CEM}} = \left(\mathbf{d}^T\mathbf{R}^{-1}\mathbf{d}\right)^{-1}\mathbf{R}^{-1}\mathbf{d} \tag{2}$$

With the optimal weight, $\mathbf{w}^{\text{CEM}}$ specified by (2) a filter called CEM, denoted by $\delta^{\text{CEM}}(\mathbf{r})$ was derived [4] specified by

$$\delta^{\text{CEM}}(\mathbf{r}) = \left(\mathbf{w}^{\text{CEM}}\right)^T\mathbf{r} = \left(\mathbf{d}^T\mathbf{R}^{-1}\mathbf{d}\right)^{-1}\left(\mathbf{R}^{-1}\mathbf{d}\right)^T\mathbf{r}. \tag{3}$$

The iterative CEM (ICEM) is implemented in conjunction of BSNE. More specifically, ICEM utilizes the band images result of BSNE as its input. The detailed step-by-step implementation of BSNE-ICEM is provided in the following.

Algorithm 1 BSNE-ICEM

1) *Initial condition*: Let $\left\{\mathbf{B}_l\right\}_{l=1}^{L}$ be the original set of band images (or a band subset generated by BSNE procedure), in which L is number of bands. Let m=1, and let $\mathbf{B}_{\text{CEM}} = \varnothing$ be the final classification fractional results.
2) For each class *m*, where *m*=1,…,*M*, let $\mathbf{\Omega}^{(0)} = \left\{\mathbf{B}_l\right\}_{l=1}^{L}$ be the initial band set, and $\mathbf{d}_m^{(0)} = \left(d_1, \cdots, d_L, d_1^{\text{NB}}, \cdots, d_{n_{\text{NB}}}^{\text{NB}}\right)^T$ be the desired target pixels in $\mathbf{\Omega}^{(0)}$. Let δ_0^{CEM} be CEM using $\mathbf{d}_m^{(0)}$ and $\mathbf{R}^{(0)}$ which are obtained from $\mathbf{\Omega}^{(0)}$. Let $k = 1$.
3) At the kth iteration update $\mathbf{d}_m^{(k)}$ with $\mathbf{d}_m^{(k)} = \begin{bmatrix}\mathbf{d}_m^{(k-1)} \\ 1\end{bmatrix}$, and $\mathbf{R}^{(k)} = \sum_{i=1}^{N}\mathbf{r}_i^{(k)}\left(\mathbf{r}_i^{(k)}\right)^T$ from $\mathbf{\Omega}^{(k)}$.
4) Use new generated $\mathbf{d}_m^{(k)}$ and $\mathbf{R}^{(k)}$ for δ_k^{CEM} to be implemented on $\mathbf{\Omega}^{(k)}$. Let $\mathbf{B}_{\text{CEM}}^{(k,m)}$ be the detection abundance fractional map produced by δ_k^{CEM} for m^{th} class.
5) Use a Gaussian filter to blur $\left|\mathbf{B}\right|_{\text{CEM}}^{(k,m)}$ where $\left|\mathbf{B}\right|_{\text{CEM}}^{(k,m)}$ is the absolute value of $\mathbf{B}_{\text{CEM}}^{(k,m)}$. The resulting image is denoted by Gaussian-filter $|\mathbf{B}|_{\text{GFCEM}}^{(k,m)}$.
6) Check if $|\mathbf{B}|_{\text{GFCEM}}^{(k,m)}$ satisfies a given stopping rule. If no, continue. Otherwise, go to step 8.
7) Form $\mathbf{\Omega}^{(k+1)} = \mathbf{\Omega}^{(k)} \cup \left\{|\mathbf{B}|_{\text{CEM}}^{(k,m)}\right\}$. Let $k \leftarrow k+1$ and go to step 3.
8) If *m*=*M*, continue. Otherwise, let $\mathbf{B}_{\text{CEM}} = \mathbf{B}_{\text{CEM}} \cup \left\{|\mathbf{B}|_{\text{CEM}}^{(k,m)}\right\}$ and $m \leftarrow m+1$, go to step 2.
9) $\mathbf{B}_{\text{CEM}}$ is the desired detection abundance fractional map and ICEM is terminated.

In order to effectively terminate ICEM, Tanimoto index (TI) is utilized [9] by

$$\text{TI}^{(k)} = \frac{|S_k \cap S_{k-1}|}{|S_k \cup S_{k-1}|} \tag{4}$$

TI is used as a stopping criterion where $|S|$ is size of a set S, S_k and S_{k-1} are the kth thresholded binary image of the kth CEM detection abundance fractional map, $\left|\mathbf{B}\right|_{\text{CEM}}^{(k,m)}$ and k-1st thresholded binary image of the k-1st CEM detection abundance fractional map, $\left|\mathbf{B}\right|_{\text{CEM}}^{(k\text{-}1,m)}$.

The detail of stopping rule is, first let $|\mathbf{B}|_{\text{GFCEM}}^{(k,m)}$ be the tested fractional classification results. Then a binary classification results $|\mathbf{B}|_{\text{Binary}}^{(k,m)}$ is obtained by utilize Otsu's method on $|\mathbf{B}|_{\text{GFCEM}}^{(k,m)}$ for class *m*, where m=1,…,*M*. Finally, TI is calculated and tested if TI achieve pre-defined TI thresholding value. If greater, then algorithm is terminated, otherwise, keep iterating.

3. ITERATIVE CEM CNN

This section presents the iterative CEM-CNN approach, to be called I(CEM-CNN), which iterative feed back the CNN processed CEM subpixel classification maps into original band images set as a new expansion set of band images after each iteration. The feedback loop is continued until the stopping rule is satisfied in ICEM.

The aim of utilizing CNN on CEM classification maps, is to capture the spatial information of each class as well as background, and then performing classification from spatial point of view. Since there is no priori spatial information can be provided to CNN for training process, in this case, in each iteration, training samples of CNN is obtained from CEM classification maps with training samples location for background is determined by recovered background map. The reason which background map is necessary to be recovered, is that unlike subpixel target detection-based classification approach, the conventional neural network classification cannot classify background since there is no training samples for it, therefor, background training samples can be randomly selected from recovered background map from binary subpixel class detection maps of CEM.

Figure 2 describes a flowchart of iterative CEM convolutional neural network approach to hyperspectral classification.

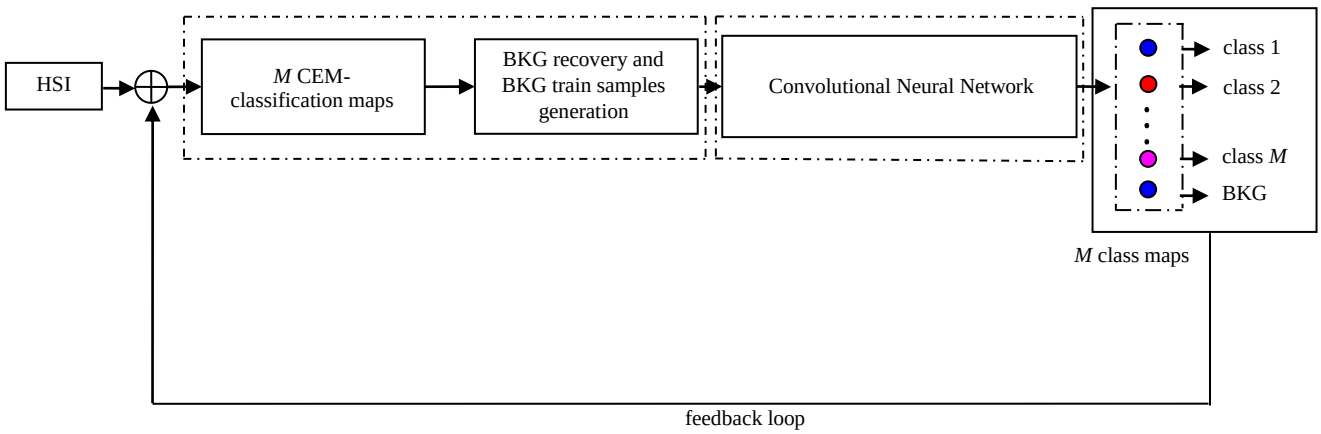


Figure 1. A diagram of iterative CEM convolutional neural network (CEM-CNN).

The detail implementation steps of iterative CEM convolutional neural network I(CEM-CNN) is provided as following.

Algorithm 2 Iterative CEM-CNN

1) *Initial Condition*: Let $\mathbf{\Omega}^{(0)}$ be the original set of band images. Input a train sample location pool, $\{TS_m\}_{m=1}^{M}$ for M classes, in which $\{TS_m\}_{m=1}^{M}$ is a set of train sample location for mth class.

2) Let $\{\mathbf{d}_m^{(0)}\}_{m=1}^{M} = \left\{\left(d_{m,1},\cdots,d_{m,L},d_{m,1}^{NB},,d_{m,n_{NB}}^{NB}\right)\right\}_{m=1}^{M}$ be the desired target pixels of M-class, which is formed by taking sample mean of each class $\{TS_m\}_{m=1}^{M}$ located pixels on $\mathbf{\Omega}^{(k)}$. Let δ_k^{CEM} be CEM subpixel detector on $\mathbf{\Omega}^{(k)}$ using $\{\mathbf{d}_m^{(k)}\}_{m=1}^{M}$ and $\mathbf{R}^{(k)}$ to produce $\{\mathbf{B}_k^{\mathrm{CEM}}\}_{m=1}^{M}$,which is M class detection map.

3) Form $\left\{\left|\mathbf{B}\right|_{\mathrm{BGCEM}}^{(k)}\right\}_{m=1}^{M}$ which is binary value class map of $\left\{\left|\mathbf{B}\right|_{\mathrm{GCEM}}^{(k)}\right\}_{m=1}^{M}$ by using Otsu's method. Find background detection map by

$$\left|\mathbf{B}\right|_{\mathrm{BKG}}^{(k)} = \left[\bigcup_{m=1}^{M}\left(\left\{\left|\mathbf{B}\right|_{\mathrm{BGCEM}}^{(k)}\right\}_m\right)\right]^C \tag{5}$$

Then randomly generate background train samples $TS_{BKG}^{(k)}$ with a specified amount of number, from the location which $\left|\mathbf{B}\right|_{\mathrm{BKG}}^{(k)}$ detected as background.

4) Implement convolutional neural network classification on $\left\{\left|\mathbf{B}\right|_{\mathrm{GCEM}}^{(k)}\right\}_{m=1}^{M}$, with train sample obtained from $\left\{\left|\mathbf{B}\right|_{\mathrm{GCEM}}^{(k)}\right\}_{m=1}^{M}$

by taking each train samples' *h*-by-*h* square surrounding, whose location specified by $\{TS_m\}_{m=1}^{M}$ and $TS_{BKG}^{(k)}$. The resulting classification map is denoted by $\left\{\mathbf{B}_{\mathrm{NN}}^{(k)}\right\}_{m=1}^{M+1}$, in which it contains *M* class classification results and background classification result of neural network.

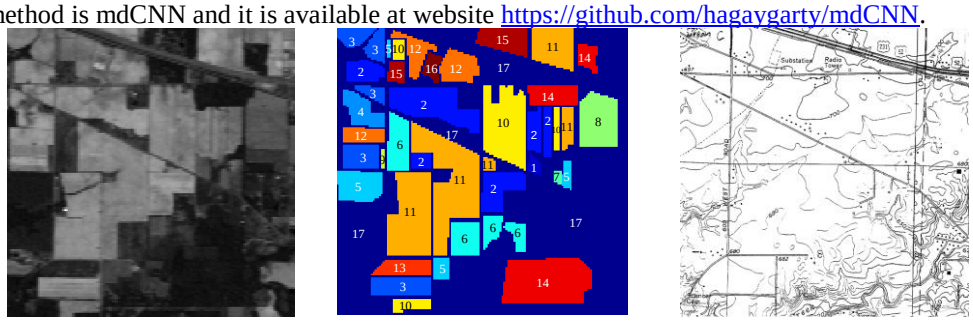
5) Implement maximum likelihood to obtain binary classification results from $\left\{\mathbf{B}_{\mathrm{GP}}^{(k)}\right\}_{m=1}^{M+1}$, and denote the *M* class and background classification resulting image separately as $\left\{\mathbf{B}_{\mathrm{binary}}^{(k)}\right\}_{m=1}^{M}$ and $\mathbf{B}_{\mathrm{BKG,binary}}^{(k)}$.

6) Check if $\left\{\mathbf{B}_{\mathrm{binary}}^{(k)}\right\}_{m=1}^{M}$ satisfies a given stopping rule. If no, continue. Otherwise, algorithm is terminated.

7) Form $\mathbf{\Omega}^{(k+1)} = \mathbf{\Omega}^{(k)} \cup \left\{\mathbf{B}_{\mathrm{GP}}^{(k)}\right\}_{m=1}^{M}$. Let $k \leftarrow k+1$ and go to step 2).

4. REAL HYPERSPECTRAL IMAGE EXPERIMENT

In this section, a popular real hyperspectral image the Indiana Indian Pines hyperspectral image is utilized on proposed method, and the detail description of the dataset can be found in [2]. The Indiana Indian Pines test site shown in Fig. 2(a), its ground truth in Fig. 2(b) and aerial view in Fig. 2(c). The data set is available at website http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes. The CNN utilized in this section for the proposed method is mdCNN and it is available at website https://github.com/hagaygarty/mdCNN.



(a) (b) (c)

Figure 2. Indiana Indian Pines scene: (a) Band 186 (2162.56nm). (b) Ground truth. (c) USGS quadrangle map of the test site.

A comprehensive comparative analysis was conducted among ICEM and I(CEM-CNN) using 5%, 10% and 15% of labeled samples as CNN training samples. In order to evaluate classification performance of the approaches in this experiment, the following experiments performance are evaluated with several parameters. More specifically, we can define variety factors

C_i = the *i*th class

n_{ij} = the number of samples in the *j*th class to be classified into *i*th class

n_{jj} = the number of samples in the *j*th class correctly classified into the *j*th class

n_j = the number of samples in the *j*th class, i.e., $n_j = \sum_{j=1}^{M} n_{ij}$

$\hat{n}_{ij}$ = the number of data samples classified in the *i*th class, which are supposed to in the *j*th class C_j

$\hat{n}_i$ = the number of data samples classified in the *i*th class, i.e., $\hat{n}_i = \sum_{j=1}^{M} \hat{n}_{ij}$

$\hat{n}_{\mathrm{BKG}}$ = the number of data samples classified as background

N = total number of data samples, i.e., $N = \sum_{j=1}^{M} n_j$

$$p(C_j) = \frac{n_j}{N}$$

Further, according to the definition above, various quantitative measures can be defined as follows.

$$\mathrm{P_A}(C_j) = \mathrm{P}(C_{jj} \mid C_j) = \frac{n_{jj}}{n_j} \tag{6}$$

$$\mathrm{P_{PR}}(\hat{C}_i) = \frac{\hat{n}_{ii}}{\sum_{j=1}^{M} \hat{n}_{ij}} \tag{7}$$

$$\mathrm{P_A} = \frac{\hat{n}_{\mathrm{BKG}}}{N} + \sum_{i=1}^{M} \frac{n_{ii}}{N} \tag{8}$$

$$\mathrm{P_{AA}} = \frac{1}{M} \sum_{j=1}^{M} \mathrm{P}_A(C_j) = \frac{1}{M} \sum_{i=1}^{M} \frac{n_{ii}}{n_{ii}} \tag{9}$$

$$\mathrm{P_{OA}} = \frac{1}{N} \sum_{j=1}^{M} n_j \left(\frac{n_{jj}}{\sum_{i=1}^{p} n_{ij}} \right) \tag{10}$$

$$\mathrm{P_{PR}} = \frac{1}{N} \sum_{i=1}^{M} \hat{n}_i \, p_{PR}(\hat{C}_i) \tag{11}$$

Table 1 tabulates the parameters used by both I(CEM-CNN) and BSNE-ICEM for this experiment, the number of samples used for background recovery procedure, and also the structure of CNN used in I(CEM-CNN) approach for the Indiana Indian Pines hyperspectral image. Table 2 tabulates the performance measure results produced by BSNE-ICEM and I(CEM-CNN) with different percentage of labeled samples as training samples for the Indiana Indian Pines hyperspectral image. Figure 2 shows the fractional classification results from I(CEM-CNN) with 15% of groundtruth labeled samples as training samples, and Figure 3 shows the classification map from Figure 2, in which each pixel is labeled as the class number or background, and class number is determined by the corresponding class with the maximum value through all fractional classification results, and background is determined using background recovery result.

Table 1. Specifications of parameters used by iterative CEM-CNN for the Indiana Indian Pines hyperspectral image scene.

I(CEM-CNN)																
Class # (C_i)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
train samples # (5%)	23	39	35	33	35	37	14	36	10	37	39	37	33	39	32	33
train samples # (10%)	23	92	80	67	69	73	15	71	10	79	104	72	67	85	69	49
train samples # (15%)	23	128	114	103	115	113	15	107	11	133	182	112	98	138	103	49
d	Class train samples mean															
Iteration # (5%)	14															
Iteration # (10%)	14															
Iteration # (15%)	13															
BKG train samples #	66(5%), 131(10%), 197(15%)															
stopping threshold	0.99															
CNN structure	Input(11×11,16)->2D conv.(5×5)-> 2D conv.(3×3)->Fully conn.(128, Relu activation)->Output(17)															
CNN train circle	4															
BSNE-ICEM																
Band selection method	UBS (29 bands)															
Nonlinear expanded bands	CBEP (812 bands)															
d	Class sample mean															
Class # (C_i)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Iteration #	3	3	3	3	3	3	6	3	6	3	4	3	6	3	3	10
Gaussian window size	11×11															
σ used in Gaussian filter	0.5 with window size 11×11															
Thresholding method	Otsu's method															
stopping threshold (TI)	0.85															

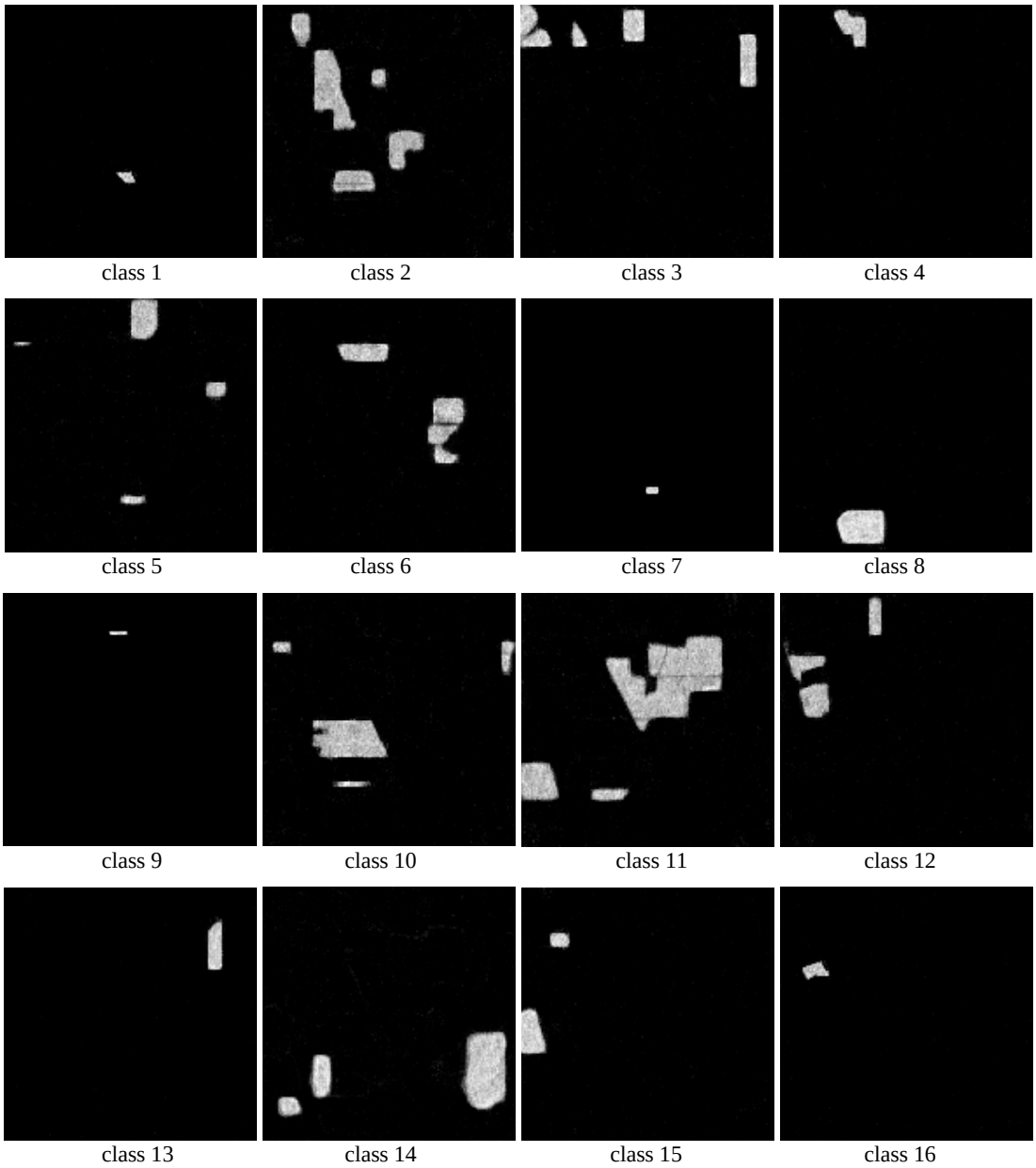


Figure 3. 16 iterative CEM-CNN classification abundance fraction class maps with 15% of labeled samples as train samples of Indiana Indian Pines scene.

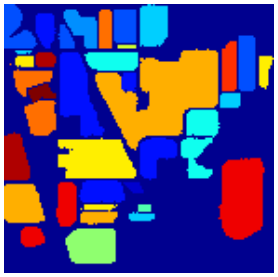


Figure 4. 16 CEM-CNN classification abundance fraction class maps in Fig. 3 binarization by maximum likelihood.

Table 2. Performance evaluation comparison among iterative CEM-CNN, and ICEM for the Indiana Indian Pines hyperspectral image scene.

	BSNE-ICEM		I(CEM-CNN) 5%		I(CEM-CNN) 10%		I(CEM-CNN) 15%	
Class(C_i)	$P_A(C_i)$	$P_{PR}(C_i)$	$P_A(C_i)$	$P_{PR}(C_i)$	$P_A(C_i)$	$P_{PR}(C_i)$	$P_A(C_i)$	$P_{PR}(C_i)$
1	0.7455	1.0000	**1.0000**	**1.0000**	0.9771	0.9771	0.9783	**1.0000**
2	0.7971	0.9287	0.9650	0.9603	**0.9789**	**1.0000**	0.9671	0.9637
3	0.6453	**0.9984**	0.9771	0.9724	0.9358	0.9891	**0.9807**	0.9795
4	0.9580	0.8201	**0.9916**	**1.0000**	0.9685	0.9792	0.9705	**1.0000**
5	0.8675	**1.0000**	0.9565	0.9726	**1.0000**	**1.0000**	0.9710	0.9895
6	0.9041	**1.0000**	0.9699	0.9593	**0.9874**	**1.0000**	0.9630	0.9604
7	0.9655	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
8	**0.9937**	0.9794	0.9854	**1.0000**	0.9681	0.9721	0.9874	**1.0000**
9	0.9000	**1.0000**	**1.0000**	**1.0000**	0.9800	0.9597	**1.0000**	**1.0000**
10	0.7500	0.9801	**0.9650**	0.9660	0.9629	**0.9828**	0.9640	0.9730
11	0.7589	0.9142	0.9825	0.9606	**0.9951**	**1.0000**	0.9817	0.9545
12	0.9224	**1.0000**	0.9578	0.9878	0.9589	0.9735	**0.9629**	0.9678
13	0.9902	**1.0000**	**1.0000**	**1.0000**	0.9741	0.9895	0.9805	0.9853
14	0.8324	0.9972	0.9652	0.9831	**0.9892**	**1.0000**	0.9684	0.9691
15	0.7720	**1.0000**	0.9663	0.9920	**0.9771**	0.9771	0.9663	0.9842
16	0.9892	**1.0000**	**1.0000**	**1.0000**	0.9789	**1.0000**	0.9892	**1.0000**
P_A	0.8754		**0.9732**		0.9729		0.9720	
P_{AA}	0.8620		**0.9801**		0.9761		0.9769	
P_{OA}	0.8089		0.9727		0.9703		**0.9729**	
P_{PR}	0.9612		0.9723		**0.9738**		0.9699	

According to Table 2, the best P_A and P_{AA} was produced by I(CEM-CNN) with 5% of labeled samples as training samples, the best P_{OA} was produced by I(CEM-CNN) with 15% of labeled samples as training samples, and the best P_{PR} was provided by I(CEM-CNN) with 10% of labeled samples as training samples. The most interesting finding is I(CEM-CNN) generated a better performance than BSNE-ICEM in terms P_A, P_{AA}, P_{OA} and P_{PR} with different percentage of labeled samples as training samples, and also, I(CEM-CNN) generally produced better $P_A(C_i)$ and $P_{PR}(C_i)$ than BSNE-ICEM for almost 16 classes.

5. CONCLUSION

This paper takes advantages of hyperspectral subpixel detection approach constrained energy minimization to address hyperspectral classification issue. CNN is utilized to generate binary pixelwise classification map from constrained energy minimization (CEM) detection results, which takes care of spectral information, and also the recovered background map. To capture spatial information, CNN classification map is feedback to hyperspectral bands to reprocess CEM in an iteration manner.

REFERENCES

[1] Kang, X., Li, S., and Benediktsson, J. A, "Spectral–spatial hyperspectral image classification with edge-preserving filtering.", IEEE Transactions on Geoscience and Remote Sensing, 52(5), 2666-2677 (2014).

[2] Xue, B., Yu, C., Wang, Y., Song, M., Li, S., Wang, L., Chen, H.-M., and Chang, C.-I, "A subpixel target detection approach to hyperspectral image classification.", IEEE Transactions on Geoscience and Remote Sensing, 55(9), 5093-5114 (2017).

[3] Yu, C., Xue, B., Song, M., Wang, Y., Li, S., and Chang, C.-I, "Iterative target-constrained interference-minimized classifier for hyperspectral classification.", IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 11(4), 1095-1117 (2018).

[4] Chang, C.-I, [Hyperspectral imaging: techniques for spectral detection and classification], Springer, 54-55 (2003).

[5] Chang, C.-I, "Target signature-constrained mixed pixel classification for hyperspectral imagery.", IEEE Transactions on Geoscience and Remote Sensing, 40(5), 1065-1081 (2002).

[6] Bishop, C. M., [Pattern recognition and machine learning] Springer, 291-319 (2006).

[7] Krizhevsky, A., Sutskever, I., and Hinton, G. E., "Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems.", 1097-1105 (2012).

[8] Chang, C.-I, [Hyperspectral data processing: algorithm design and analysis], John Wiley & Sons (2013).

[9] Theodoridis, S., & Koutroumbas, K., [Pattern Recognition], Academic Press., New York, 366 (1999).