ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/333098647

Hyperspectral image classification using Non-negative Tensor Factorization and 3D Convolutional Neural Networks

Article · May 2019

DOI: 10.1016/j.image.2019.05.004

CITATIONS
9

READS
106

3 authors, including:



Sayeh Mirzaie
University of Tehran
17 PUBLICATIONS **89** CITATIONS

SEE PROFILE



Hugo Van hamme
KU Leuven
292 PUBLICATIONS **3,752** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

ALADIN View project

INSPIRE View project

All content following this page was uploaded by Sayeh Mirzaie on 24 June 2019.

The user has requested enhancement of the downloaded file.

Hyperspectral Image Classification Using Non-negative Tensor Factorization and 3D Convolutional Neural Networks

Sayeh Mirzaei,[a] Hugo Van hamme,[b] Shima Khosravani[c]
[a]School of Engineering Science, College of Engineering, University of Tehran, Tehran, IRAN
[b]ESAT, PSI, KULeuven, Belgium
[c]Sensifai Company

Abstract. In this paper, we address the task of hyperspectral image classification using a 3-D Convolutional Neural Network (CNN). Instead of commonly used raw spectral features, discriminative features are obtained by factorizing the hyperspectral data with Non-negative Tensor Factorization (NTF). The factors consist of the endmembers' spectral signature matrix as well as the abundance matrix. The acquired abundance maps are exploited to extract the feature vectors representing the spatio-spectral properties of the image. Unlike Non-negative Matrix Factorization (NMF) where the pixels' spectra are stacked in columns of the data matrix, the spatial information is preserved with NTF. Morphological attribute filters are also applied to the extracted abundance maps and construct the discriminative training features that are fed to a 3D CNN. We develop and optimize a 3D CNN framework for image classification. Using joint spatio-spectral features and parameter sharing, a 3D CNN yields promising classification performance. It is shown in the experiments that the proposed feature sets can lead to better results in terms of classification accuracy compared to both raw spatio-spectral data and to NMF. We also demonstrate the effectiveness of the proposed CNN by comparing the results with a SVM classifier.
Keywords: Hyperspectral image classification, Non-negative Tensor Factorization (NTF), Convolutional Neural Network (CNN).

1 Introduction

Hyperspectral image (HSI) processing has found multiple applications in various fields including medical image analysis [24], minerology [25], agriculture [26] and remote sensing data analysis [1,2]. Hyperspectral data can cover the 300-2500-nm region of the electromagnetic spectrum and provides contiguous spectral bands [1,3]. Hyperspectral imagery provides information about both the spatial and spectral distribution of the constituent materials. The sensors utilized for HSI capturing have low spatial resolution. Hence, the spectrum of each pixel is composed of the spectrum of different materials. The objective of the hyperspectral unmixing (HSU) methods is to decompose the mixed pixels into the spectral signatures of the constituent substances and the corresponding fractional abundances. The linear mixing model (LMM) is commonly adopted in the literature. It supposes that a pixel spectrum is a linear combination of the endmembers' spectra. Since the Hyperspectral data is non-negative, non-negative matrix factorization (NMF) methods

have been widely applied to accomplish the unmixing task [4,5]. The columns of the data matrix correspond to the spectral vectors of the pixels in the image. The factorization matrices denote the endmember matrix containing the spectral signatures of the constituent materials and the abundance matrix which specifies the fraction of the endmembers at each pixel. Different constraints such as sparsity of the abundance matrix or some spatial or spectral conditions can be enforced to the NMF model.

Classification of every pixel of an image is of great importance in remote sensing applications. Various classification methods have been proposed in the literature. In [6], unmixing results have been used to improve the classification performance. In [7], a low-rank group inspired dictionary learning has been used for HSI classification. A residual neural network is proposed in [8] which takes the raw 3-D cubes as the input to the classifier. In [9], the hyperspectral tensor containing the spatio-spectral information of the target pixel and the neighboring pixels is organized as a multi-feature cube used for image classification using a convolutional neural network (CNN). In [10], a deep learning based method called random patches network is proposed which directly regards the random patches taken from the image as the convolutional kernels without any training. A so-called morphological-based feature space discriminant analysis (MBFSDA) method is proposed in [11] for spatio-spectral feature extraction.

In this paper, we focus on the well adopted LMM. First, we propose to use Bayesian NMF (BNMF) for HSI decomposition which assumes that the abundance matrix elements are drawn from a gamma prior distribution and the generative model for the data is a Poisson distribution. It has been shown that imposing the sparsity constraint on the abundance factors through choosing a

sparse Gamma prior is well fitted to various hyperspectral datasets and leads to good unmixing results.

Inspired by the 3D structure of the hyperspectral data, we propose to apply two different non-negative tensor factorization (NTF) models to the data tensor to extract the abundance and spectral signature matrices. This way, the spatial structure is maintained in the 2D abundance maps, while in NMF the spatial information is disregarded by stacking the pixels' spectra in the columns of the data matrix. Morphological attribute filters are applied to the obtained 2D abundance maps to make the training features which are fed to the proposed 3D CNN classifier.

During the past few years, with the proliferation of the computational power and the amount of available training data, CNNs have gained superhuman performance in many machine learning tasks. They have some major advantages over other traditional deep neural networks; a neuron in a convolutional layer is connected to a rectangular-shaped field containing neighboring neurons in the previous layer. Sharing rectangular-shaped receptive patterns among all the neurons in the convolutional layer not only considers local connectivity and correlation among adjacent neurons in the previous layer, but also substantially reduces the number of parameters, which alleviates the problem of overfitting in traditional neural network architectures. Recently, CNNs have gained state of the art results in HSI. A typical CNN including convolution, pooling, and fully connected layers is proposed in [12] and compared with SVM. A self-improving CNN is presented by using the concept of dither which resolves the problems of the curse of dimensionality and lack of available training samples [13]. A combination of a CNN and an optimized extreme learning machine is proposed in [14] as a hyperspectral image classification framework. A novel CNN architecture with a multiple feature learning method is proposed in [27]. After producing various

features from raw data, a concatenation layer outputs joint feature maps to the prediction layers which are finally used for classifying pixels.

In [28], a deformable CNN so called DHCNet is used for feature extraction, In this model, the size and shape of the convolutional sampling locations can be learned during the training procedure according to the complex spatial contexts of the HSIs. Moreover, instead of applying typical pooling layers with fixed size, learnable strided convolution is used for down-sampling according to spatial contexts. Novel squeeze multibias network (SMBN) is proposed in [29]. The multibias module (MBM) adaptively selects the best feature maps as input to the classifier. Using a squeeze convolution module instead of the traditional convolutional layer decreases the number of parameters and reduces training time. The R-3D-CNN model in [30] is designed to address the issue of noisy patches and best exploitation of spectral context.

In the current work, we develop a 3D CNN framework to jointly use spectral and spatial features. Dropout and the rectified linear unit (ReLU) activation function are utilized to overcome overfitting. The major contribution of this paper is that it proposes NTF techniques in combination with morphological attribute filters as effective feature extraction approaches. It is shown experimentally that applying attribute filters to the abundance maps obtained through NTF methods provides quite discriminative features for the CNN. They are more descriptive than NMF features since the spatial information is preserved. This is revealed through evaluating the classification performance for the mentioned features. Feature engineering has been also exploited in other classification works with the aim to improve the performance [31,32]. Additionally, we

designed and optimized a 3D CNN architecture for addressing the classification task which properly uses the extracted features.

The remainder of this paper is organized as follows. Section 2 explains the NMF and NTF approaches used in the experiments. Section 3 is dedicated to the CNN architecture used for classification. The experimental results are reported in section 4. Some conclusive notes are ultimately presented in section 5.

2 Decomposing HSI using NMF or NTF

2-1- Bayesian NMF

Non-negative matrix factorization provides a useful tool for many signal processing applications. The non-negative matrix $\mathbf{V}_{F\times N}$ is approximated as the product of two non-negative matrices $\mathbf{E}$ and $\mathbf{A}$ of size $F\times K$ and $K\times N$ respectively. In the unmixing task, $\mathbf{V}$ indicates the HSI data with F spectral bands and N pixels. $\mathbf{E}$ denotes the endmember matrix containing individual substance spectral signatures in its columns and $\mathbf{A}$ indicates the abundance matrix. K is the number of different classes i.e. constituent materials of the image. Here, we utilize a Bayesian version of NMF. In the Bayesian framework, the elements of the abundance matrix are presumed random variables drawn from a sparse gamma prior distribution: $a_{kn} \sim \text{Gamma}(a_{kn} \mid \alpha_k, \beta_k)$ where

$$\text{Gamma}(x \mid \alpha, \beta) = \left[\beta^{\alpha}\Gamma(\alpha)\right]^{-1} x^{\alpha-1} e^{\frac{-x}{\beta}},\ x \geq 0,\ \alpha > 0,\ \beta > 0$$

where α and β are the shape and scale parameter of the distribution respectively. The shape parameter will be chosen less than 1, such that the probability density is highest for x=0, hence

inducing sparsity in the solution, a common and reasonable assumption in hyperspectral unmixing. The generative model of the data matrix elements v_{fn} is taken as below:

$$v_{fn} \sim \mathrm{P}(v_{fn} \mid \sum_{k} e_{fk} a_{kn}) \qquad (1)$$

where P(x|λ) denotes the Poisson distribution with rate parameter λ defined by:

$$\mathrm{P}(x \mid \lambda) = e^{-\lambda} \frac{\lambda^{x}}{\Gamma(x+1)} \qquad (2)$$

e_{fk} denotes the elements of the endmember matrix and a_{kn} indicates the abundance matrix elements. From this model, we also see that the choice of the scale parameter of the (gamma) distribution of the abundance matrix is not important, since a_{kn} is multiplied with the deterministic parameters e_{fk}, which would scale inversely. The variational Bayesian update relations [15] are used to estimate the end member and abundance matrix factors. The proposed sparse Bayesian NMF (BNMF) model might lead to satisfactory unmixing results. However, as will be shown in the experiments, for the classification task, the abundance maps obtained through NTF result in superior performance compared to BNMF.

2-2- Beta-NTF

Let $\mathbf{V}$ be a third order HSI tensor ($\mathbf{V} \in \mathbb{R}^{F,I,J}$) where $I \times J$ is the number of pixels and F is the number of spectral bands. The beta-NTF model decomposes tensor $\mathbf{V}$ into three matrices $\mathbf{E}_{F\times K}$, $\mathbf{H}_{I\times K}$ and $\mathbf{Q}_{J\times K}$ according to the PARAFAC model:

$$\begin{aligned} &\mathbf{V} \approx \sum_{k=1}^{K} \mathbf{e}_k \otimes \mathbf{h}_k \otimes \mathbf{q}_k \quad \mathbf{E},\mathbf{H},\mathbf{Q} \geq \mathbf{0} \\ &\mathbf{W} = [\mathbf{e}_1 ... \mathbf{e}_K] \\ &\mathbf{H} = [\mathbf{h}_1 ... \mathbf{h}_K] \\ &\mathbf{Q} = [\mathbf{q}_1 ... \mathbf{q}_K] \end{aligned} \qquad (3)$$

where $\otimes$ denotes the outer product operation and the outer product of three vectors $\mathbf{e}_k \otimes \mathbf{h}_k \otimes \mathbf{q}_k$ is called a rank-1 tensor. The smallest *K* for which the equation turns to equality is defined as the rank of the tensor. The approximation in (3) minimizes the beta divergence. Here, we take $\beta = 2$ which makes the Beta divergence equivalent to the Euclidean distance. Multiplicative update rules [16] are used to solve this optimization problem. $\mathbf{E}$ contains the endmember spectral signatures in its columns. The Khatri-Rao product of the matrices $\mathbf{H}$ and $\mathbf{Q}$ constructs the abundance matrix $\mathbf{A}_{IJ\times K}$: $\mathbf{A}_{IJ\times K} = \mathbf{H} \odot \mathbf{Q}$. We obtain *K* abundance maps ($\mathbf{A}_{I\times J\times K}$) by reshaping the abundance matrix. Contrary to the case of NMF, the spatial structure is preserved in the derived 2D abundance maps.

2-3- Bayesian CANDECOMP / PARAFAC (BCP) Decomposition

In the nonnegative CANDECOMP/PARAFAC (NCP) framework [17], the aim is to approximate a tensor $\mathbf{V}_{F\times I\times J}$ with the sum of *K* rank-1 tensors like what is stated in Eq. (3).

In the Bayesian framework, we assume that the tensor data elements are random variables drawn from a Poisson distribution. The alternating Poisson regression method has been developed in

[18] for solving the Poisson CP model. It has been shown that Poisson generative model can be properly fitted to sparse tensor data [18]. The difference with Beta-NTF model lies in the cost function: The Poisson data model is equivalent to a CP problem with KL divergence as discrepancy measure between data (**V**) and model (right hand side) in Eq. (3), Again, the abundance matrix $\mathbf{A}_{IJ\times K}$ is calculated by Khatri-Rao product of the matrices $\mathbf{H}$ and $\mathbf{Q}$.

3 Feature extraction and classification using CNN

The overall architecture is presented in Figure 1. The input HSI data is either not processed (raw), the result of EMAP-processing (explained below) after preprocessing with one of the three decompositions of section 2. Pixels in the resulting 3D images are classified with a CNN (explained below).

Several authors have used morphological attribute profiles (MAP) to extract the spatial information of the image [19-20]. Multiple attribute filters are applied to the image to construct the attribute profile (AP). The filters merge connected components in the image. When APs are acquired for different features and stacked together, the obtained profile is called extended multi-attribute profile (EMAP) [19]. The output of the attribute filter is compared with a threshold for each region. If the output of the attribute filter is smaller than the threshold, the region grayscale values are replaced with the neighboring region with closest value. The operation is called thinning when the region is merged with a lower grayscale value and it is called thickening when it is merged with larger grayscale value. Some useful attributes for HSI analysis include area, volume (sum of the intensities of the pixels belonging to each region), length of the diagonal of the box bounding each region, moment of inertia, shape factor, homogeneity, standard deviation, and

entropy of the grayscale values of the pixels [20]. Here, we apply attribute filters to the obtained abundance maps to construct the training features fed to the 3D CNN classifier. The EMAP vector of a pixel is obtained by repeating the thickening and thinning for T different threshold values to each of the K images, and keeping the original image as well. Hence, the EMAP feature vector has length $(2\times T+1)\times K$

To further clarify the proposed feature extraction framework, a block diagram is presented in figure 1. Since BNMF requires 2D input, the HSI tensor is reshaped and the pixels' spectra are stacked in columns of the data matrix $\mathbf{V}_{F\times IJ}$. The length of the feature vector for each training pixel is $(2\times T+1)\times K$ as described above. For raw data, the original data tensor is used so the feature vector size is F.

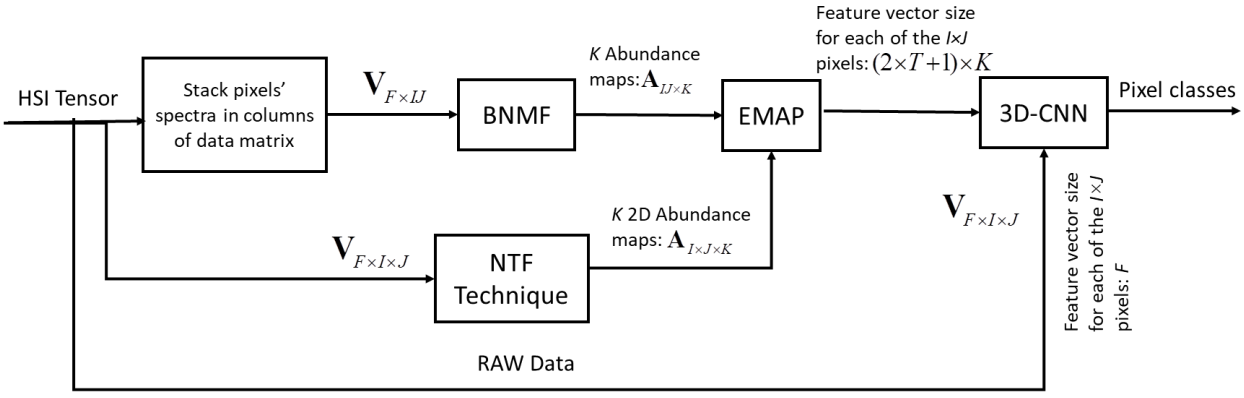


Figure 1- Feature extraction process flowchart

A 3D CNN makes use of both spectral and spatial features. We decompose the network input into 3D patches with dimensions $s\times s\times c$ where c is the same as the spectral dimension of the input. Each of these patches centers at a specific pixel and has a label corresponding to this center pixel. Suppose that the input data of a 3D convolutional layer consists of M cubes, $\mathbf{x}_m$ $(m=1,2,...,M)$ with

size $I \times J \times K$. Then, the output of each kernel with size $A \times B \times D$ would be a 3D tensor $\mathbf{Y}$ which is derived as below:

$$\mathbf{Y}(i, j, k) = \sum_{m=1}^{M} \sum_{a=0}^{A-1} \sum_{b=0}^{B-1} \sum_{d=0}^{D-1} \mathbf{X}_m(i+a, j+b, k+d).Kernel_m(a, b, d)$$
$$i = 1, 2, ..., I - A + 1 \quad j = 1, 2, ..., J - B + 1 \quad k = 1, 2, ..., K - D + 1 \tag{4}$$

Our proposed 3D-CNN architecture consists of two consecutive convolutional layers, a max-pooling layer and then three consecutive fully connected layers. The first and second convolutional layers have n_1 kernels with size $a_1 \times a_1 \times d_1$ and n_2 kernels with size $a_2 \times a_2 \times d_2$ respectively. For both convolutional layers we choose strides of 1×1×1 without padding, so the output of the first convolution layer will be n_1 cubes each with dimensions $(s\text{-}a_1+1) \times (s\text{-}a_1+1) \times (c\text{-}d_1+1)$ and output of the second convolutional layer will be n_2 cubes with dimensions $(s\text{-}a_1\text{-}a_2+2) \times (s\text{-}a_1\text{-}a_2+2) \times (c\text{-}d_1\text{-}d_2+2)$.

To avoid the vanishing gradient problem, we choose ReLU as the activation function for all layers except for the last layer which uses softmax activation. We apply dropout [21] and early stopping [22] as solutions to address overfitting. Gradient descent with momentum is used, a very popular optimization algorithm in deep learning [23]. The parameters of the proposed architecture include height and width of the patches, the number of convolutional layers, the number of kernels in the convolutional layers, the size of the kernels in the convolutional layers, the number of fully connected layers and the number of neurons in the fully connected layers. They are all optimized such that maximum classification accuracy is attained for the validation samples.

4 Experiments

To evaluate the proposed method, experiments have been carried out on the well-known real hyperspectral datasets Indiana Pines, Pavia and Salinas. A description of the datasets is given in the following subsections. The hyperparameters of the gamma distribution are chosen as $\alpha = 0.2$ and $\beta = 1$. We use two morphological attributes: the area of the region and the length of the diagonal of the box bounding the region. The corresponding threshold values are taken as (10,15) and (50,100) respectively. These values are selected empirically. Therefore, the size of the feature vector after applying EMAP equals to $9 \times K$. The termination criterion showing the convergence of NMF and NTF algorithms is defined by comparing the relative optimization cost function gain with a threshold. All of the factorization tasks are repeated 10 times and the results related to the minimum cost function value are withheld.

To find the optimal values for the parameters of the 3D CNN, we randomly selected 10% of the labeled samples of each class for training, 10% for validation and the remaining 80% for testing. The optimum network architecture is illustrated in Table 1. The first 3D convolution layer has 2 kernels with size 3×3×7. The second 3D convolution layer has 4 kernels with dimension 3×3×1. We used dropout with ratio 0.25 after the second convolution layer. Finally, 3 fully connected layers with 64, 42 and *K* neurons are implemented where *K* is equal to the number of classes for each dataset. The size of the batch is set to 20. Cross entropy is used to define the loss function. The average results using tenfold cross validation of the proposed classification algorithm are reported.

Table1- 3D CNN architecture

Layer#	Type	#kernles	kernel-size	Activation
1	conv3d	2	(3, 3, 7)	ReLU
2	conv3d	4	(3, 3, 1)	ReLU
	Dropout(.25)	-	-	
3	Dense(64)	-	-	ReLU
4	Dense(42)	-	-	ReLU
5	Dense(*K*)	-	-	softmax

A) Indiana Pines dataset

This scene was collected by the AVIRIS sensor over the Indian Pines test site in North-Western Indiana. It contains 145×145 pixels and 224 spectral bands in the wavelength range 0.4–2.5 µm. The spatial resolution of this dataset is 20 meters per pixel. The image consists of two-thirds agriculture, and one-third forest or other natural perennial vegetation. The ground truth contains sixteen classes as demonstrated in Figure 2. The number of bands have been reduced to 200 by removing those bands which contain the regions of water absorption: (104-108), (150-163) and 220. Indiana Pines is a challenging dataset because the samples are quite unevenly distributed among 16 classes in such a way that some classes possess very few samples. The resultant classification performance of the proposed framework can be observed in figure 3. The results are depicted for different input features. The input patch size is set to 7×7×*c*. Raw features correspond to the original HSI tensor. Thus, the spectral dimension *c* is equal to *F* for raw data and $(2\times T+1)\times K$ for the BNMF and NTF methods. To analyze the overall classification performance,

average and overall accuracies (AA and OA respectively) are also computed and listed in Table 2. As can be seen, BCP leads to best performance in terms of both average and overall accuracies.

Applying morphological filters can enhance the spatial context extracted from the 2D abundance maps. Furthermore, it is an ideal way to increase the size of the feature vectors from K (number of classes) to $(2\times T+1)\times K$ where T specifies the length of the threshold vector.

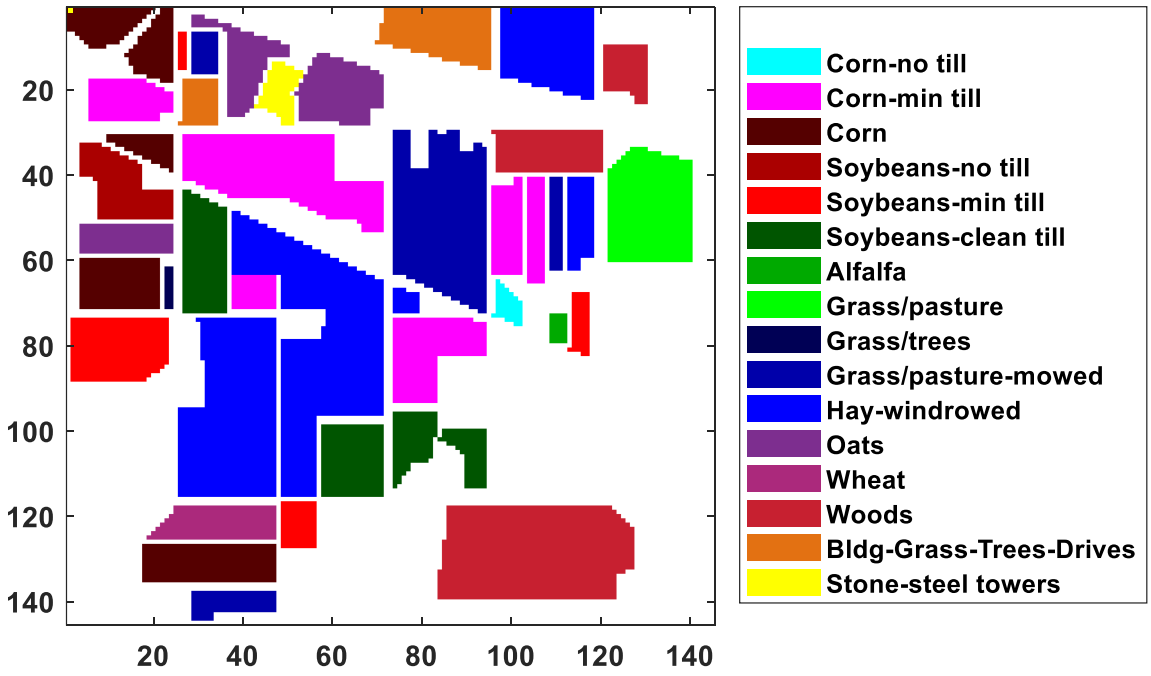


Figure 2. Ground truth map of Indiana Pines scene containing 16 classes

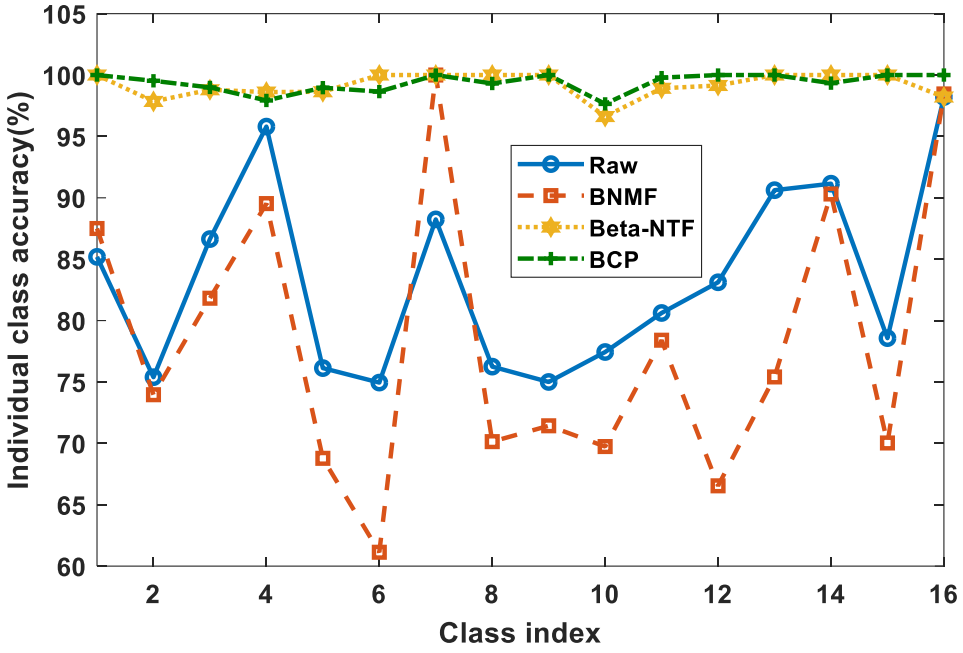


Figure 3- The individual class accuracies (in percent) obtained for Indiana Pines dataset

Table 2- Classification accuracies obtained for Indiana using 3D CNN

` *METHODS*	*RAW*	*NMF*	*BETA-NTF*	*BCP*
AVERAGE ACCURACY(%)	83.29	78.29	99.16	**99.36**
OVERALL ACCURACY(%)	81.4	75.97	98.86	**99.18**

To have a better representation of the achieved accuracy measures, we provide the classification map obtained for the BCP method with 10 percent training. The result is depicted in figure 4 and can be compared with groundtruth given in figure 2. As can be seen, misclassification mostly occurs for small classes.

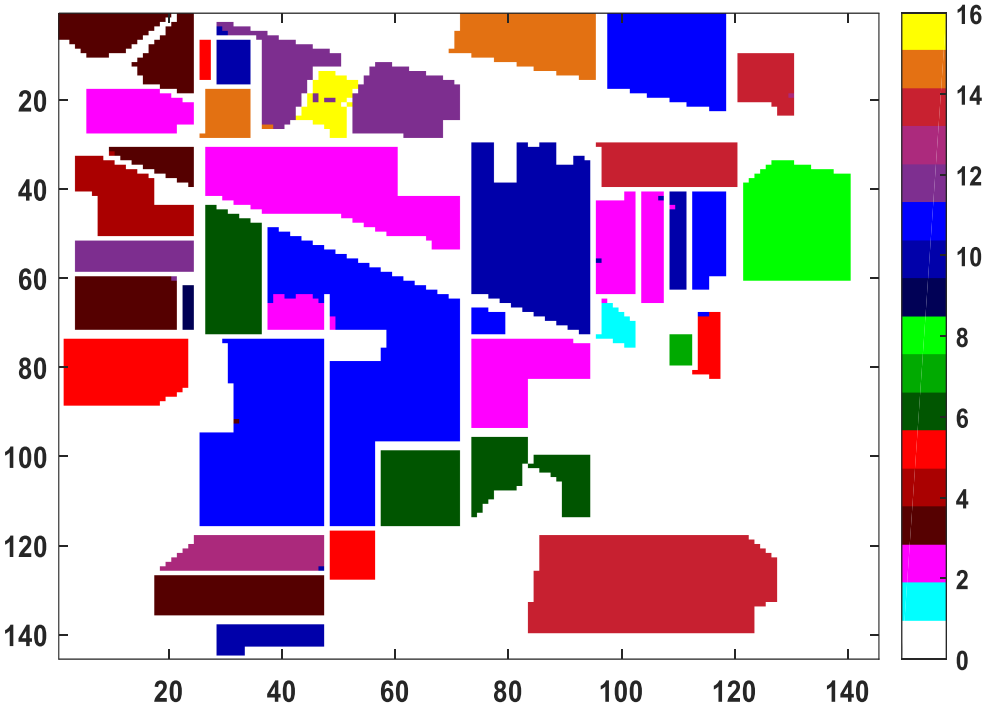


Figure 4. Classification maps of Indiana obtained by applying BCP

B) Pavia University Dataset

This scene was gathered by ROSIS sensor during a flight campaign over Pavia, northern Italy. There are 610×340 pixels in the image and the number of spectral bands is 103. The spatial resolution is 1.3 meters. The ground truth data includes 9 classes as depicted in Figure 5. The resulting individual class accuracies can be seen in Figure 6. The AA and OA values are reported in Table 3. Again, BCP outperforms the other methods.

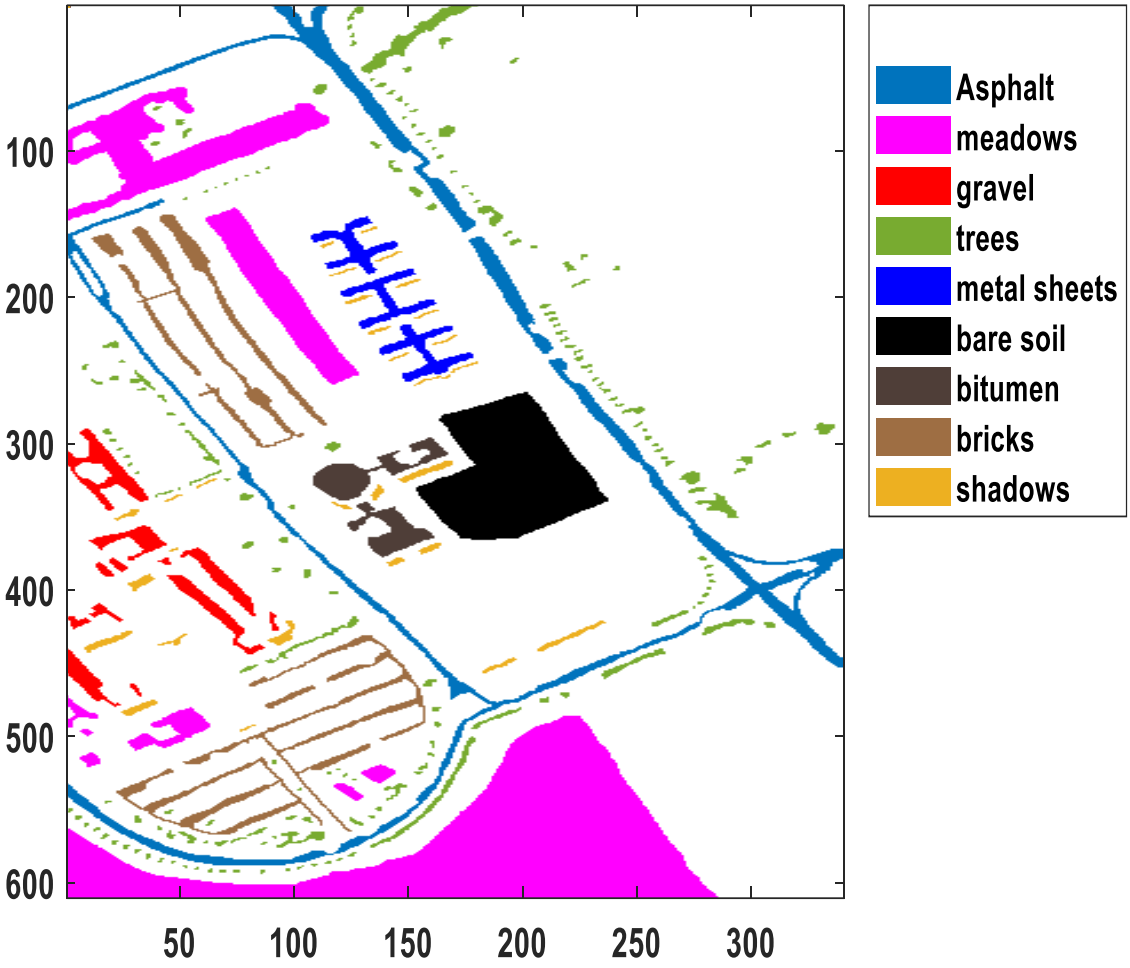


Figure 5. Ground truth map of Pavia University scene containing 9 classes

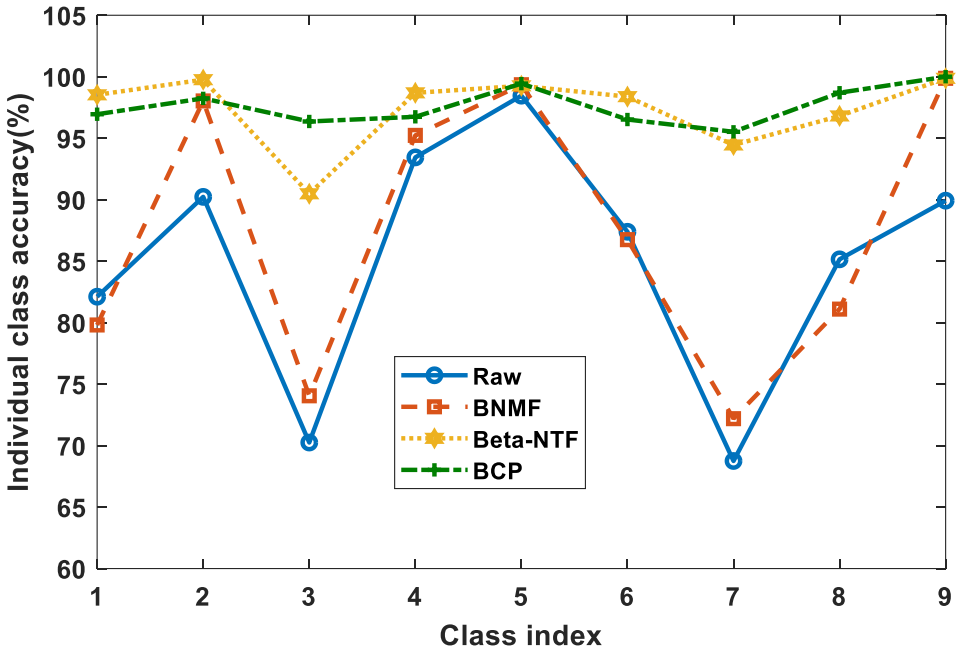


Figure 6- The individual class accuracies (in percent) obtained for Pavia University dataset

Table3- Classification accuracies obtained for Pavia using 3D CNN

` METHODS	RAW	NMF	BETA-NTF	BCP
AVERAGE ACCURACY(%)	75.94	75.97	97.40	**97.65**
OVERALL ACCURACY(%)	75.42	79.87	97.24	**97.53**

Classification map obtained for BCP method with 10 percent training data is represented in figure 7 and can be compared with the ground truth given in figure 5. Pavia contains more combined and narrow classes. However, as can be observed, due to high spatial resolution of this dataset, satisfying performance is obtained for almost all classes.

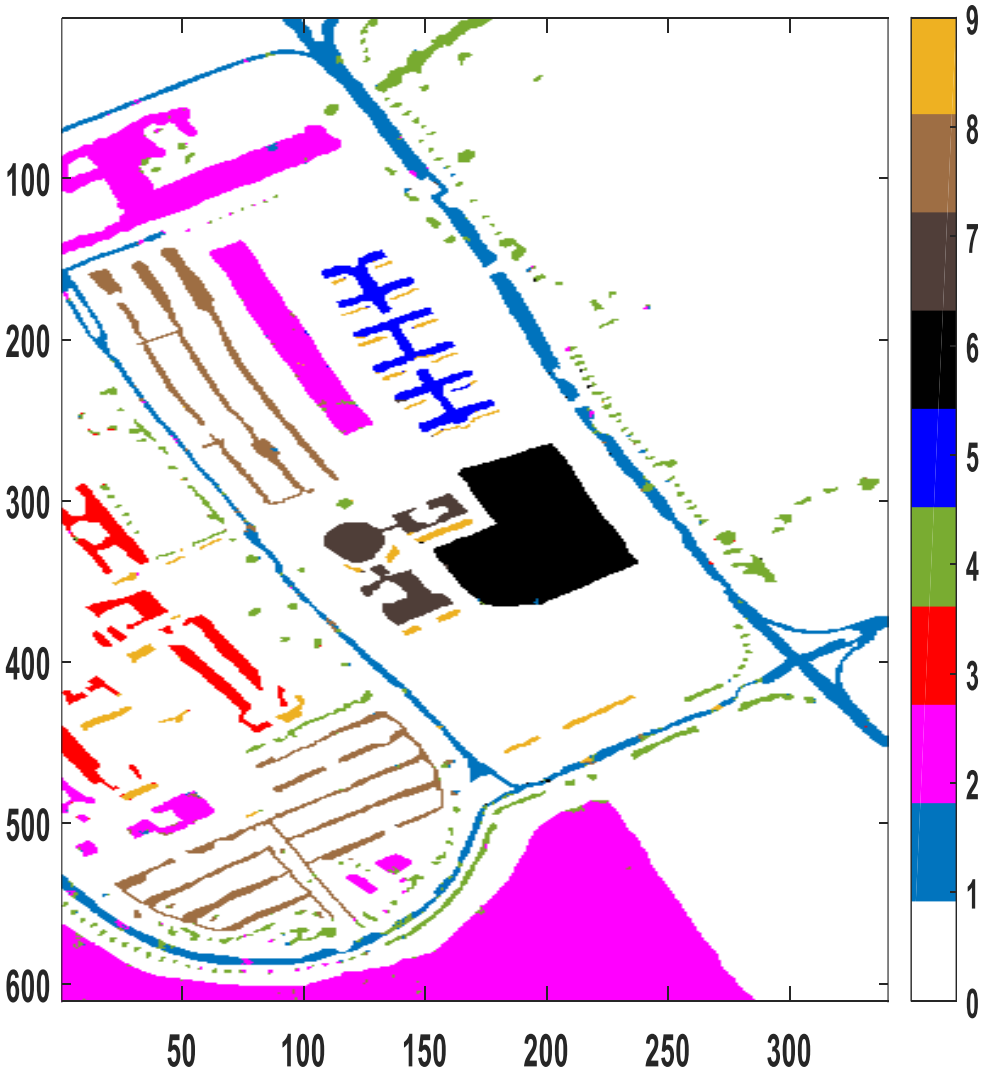


Figure 7. Classification maps of Pavia obtained by applying BCP

C) Salinas Dataset

This scene was collected by the AVIRIS sensor over Salinas Valley, California. The spatial resolution is 3.7 meters. The image consists of 512×217 pixels and 224 spectral bands. As with Indian Pines scene, 20 water absorption bands, (108-112), (154-167), 224 are discarded. Salinas ground truth contains 16 classes including bare soils, vegetables and vineyard fields (see Figure

8). The outcome of the suggested classification methods is represented in Figure 9. OA and AA values are listed in Table 4. BCP provides superior results.

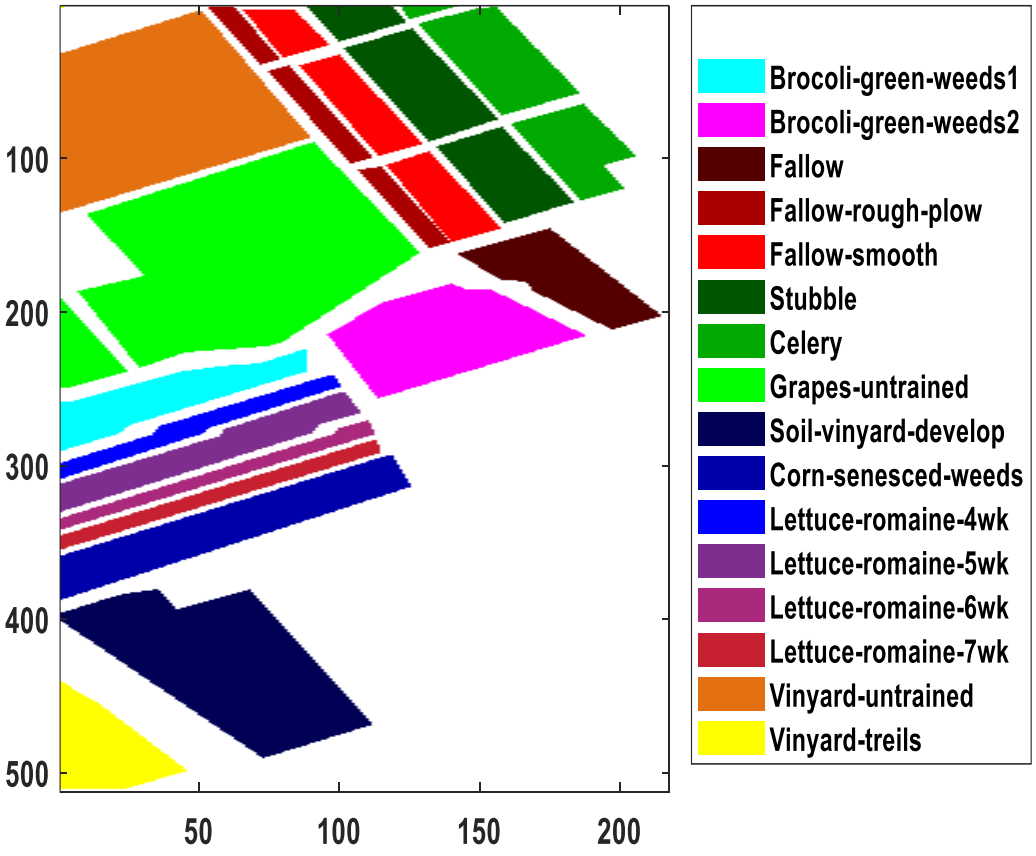


Figure 8. Ground truth map of Salinas scene containing 16 classes

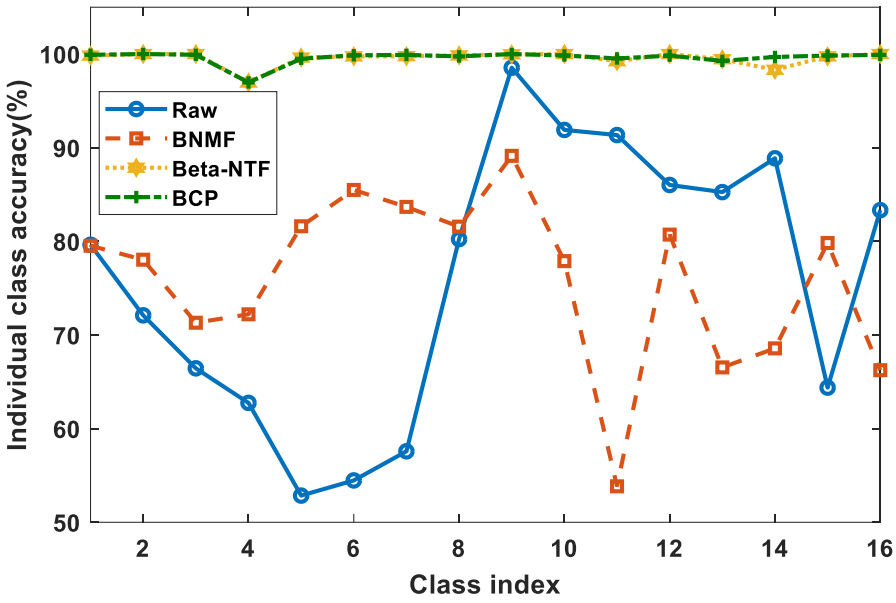


Figure 9- The individual class accuracies (in percent) obtained for Salinas dataset

Table4- Classification accuracies obtained for Salinas using 3D CNN

` METHODS	RAW	NMF	BETA-NTF	BCP
AVERAGE ACCURACY(%)	75.94	75.97	99.53	**99.57**
OVERALL ACCURACY(%)	75.42	79.87	99.74	**99.76**

Classification map obtained for BCP method is represented in figure 10. As can be seen, there are very few misclassified samples mostly located in the boundaries of small classes.

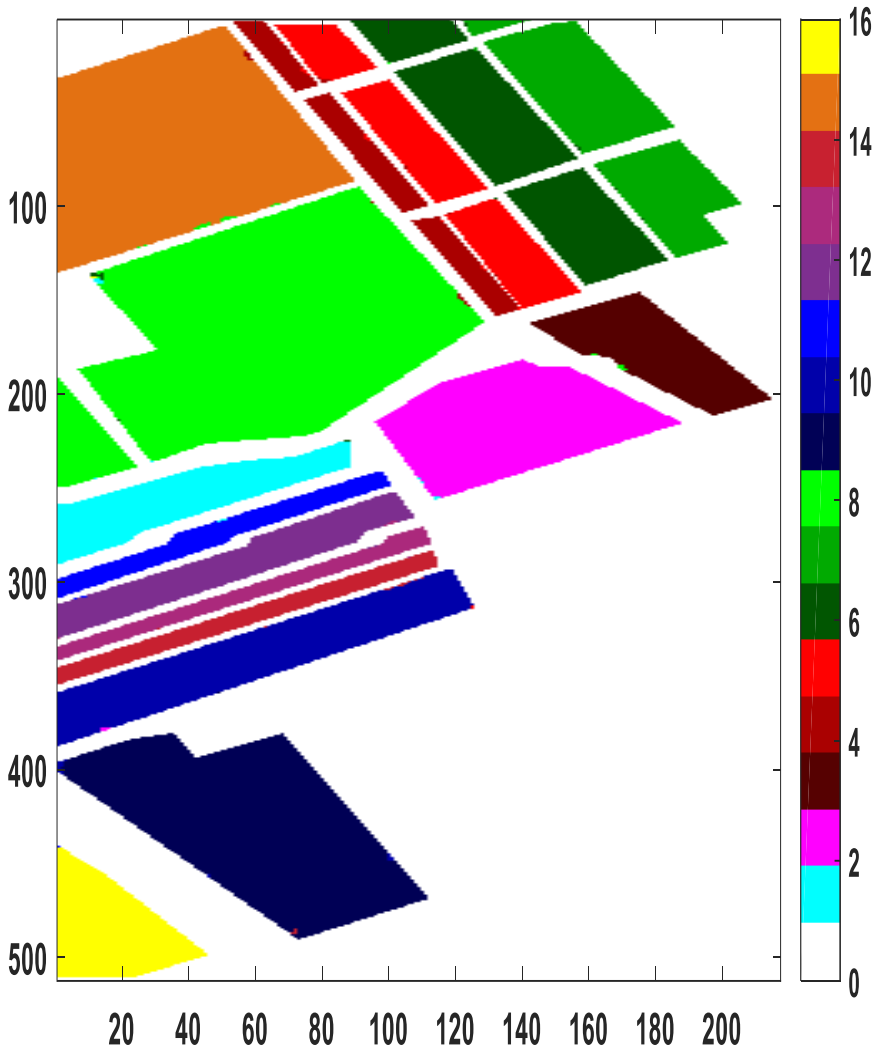


Figure 10. Classification maps of Salinas obtained by applying BCP

5 Discussion

In this section, we analyze the sensitivity of the results with respect to different parameters such as input patch size, kernel size and percentage of training samples. We have also provided comparison with other recently proposed methods.

A) Impact of patch size

We are going to investigate the effect of input patch sizes on classification performance. For this purpose, the average accuracies obtained by BCP are depicted against the patch sizes in figure 11, showing that a patch size of 7×7×*c* leads to satisfying performance for all datasets. Compromising between model complexity and classification performance, we select *s*=7 as the optimum patch size.

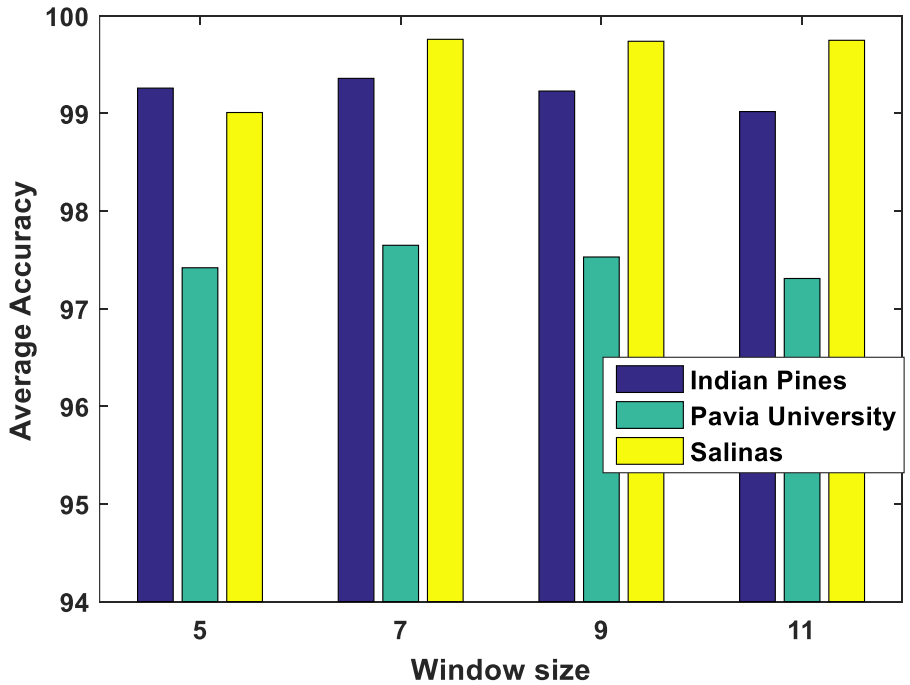


Figure 11- Average Accuracy (%) against the patch size

B) Impact of Kernel size

One of the most effective factors related to convolutional layers is kernel size. Kernel size, specifies the amount of neighbor information taken into account for the current pixel. Here, we explore the results for $k \times k$ kernels with k=3,5,7,9. Averge accuracy is depicted in figure 12 against the kernel size. The patch size is set to s=7 and 20% of the samples of each class are used for training. It can be observed that the best kernel size for all three datasets is 3. This can be assigned to the fact that smaller filters capture finer details of the image leading to better classification performance.

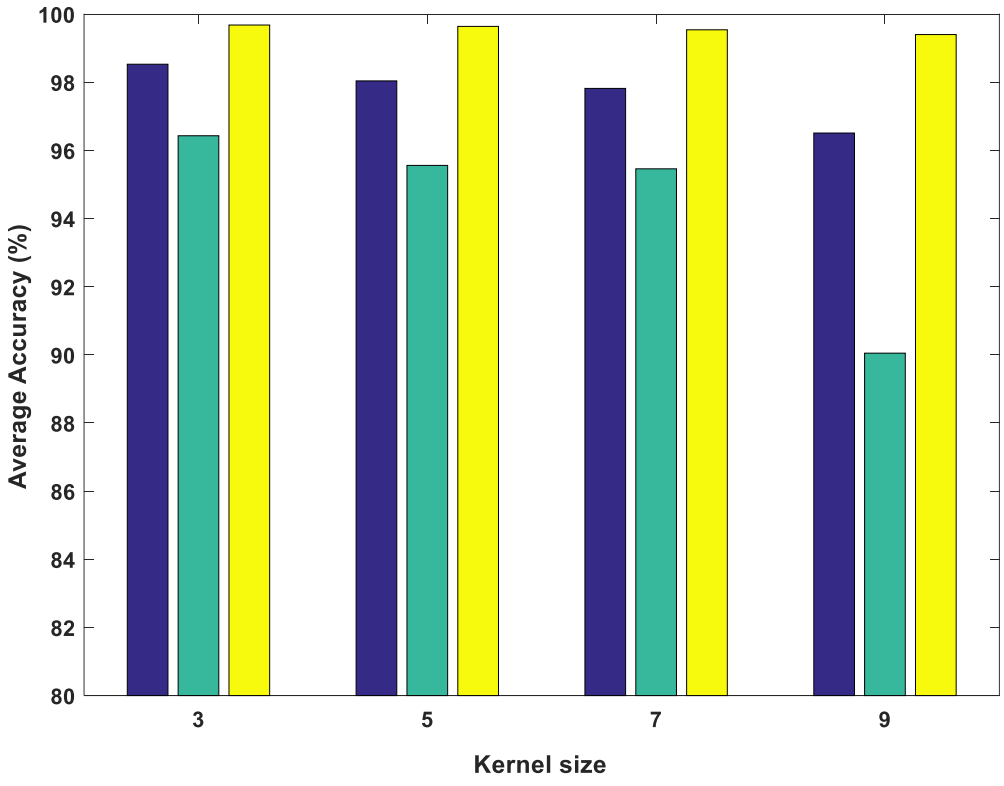


Figure 12- Average Accuracy (%) against the kernel size

C) Impact of percentage of training samples

To assess the effect of increasing the training samples, the average accuracy is evaluated against the percentage of training samples in Figure 13. Again, the results are obtained for BCP and the patch size is set to s=7. It can be observed that with increasing the number of training samples, greater performance improvement occurs for Indiana Pines. This can be associated to the fact that the total number of labeled samples of Indiana Pines dataset are smaller than in the other two datasets. Furthermore, the spatial resolution of Indiana Pines is lower than the others which leads to an increased occurrence risk of mixed pixels. Pavia has the highest spatial resolution among the three datasets which has led to well generalized model. Salinas contains the highest number of elements in the tensor data (512×217×204) which has resulted in the best overall performance.

Indiana Pines is a challenging dataset for performing classification due to its quite non-uniform distribution of the labeled samples among classes leading to insufficient training data for some classes. As can be inferred from the acquired classification accuracies, combination of the proposed feature extraction and the CNN classifier can result in robust classification performance for this dataset.

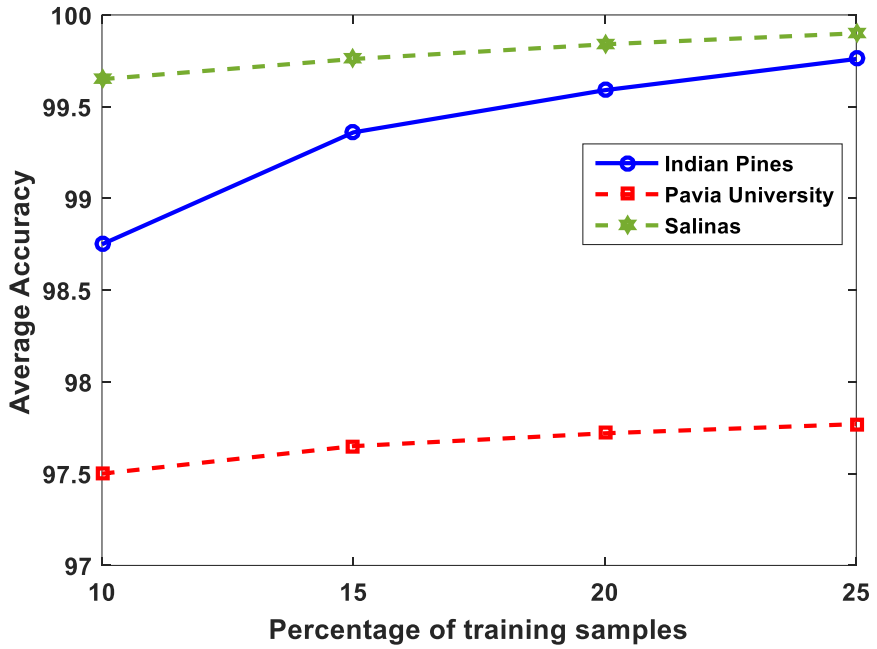


Figure 13- Average Accuracy (%) against the percentage of training samples

D) Comparison with other methods

To provide comparative material for examining the effectiveness of the propose classifier, we have computed the evaluation metrics for different feature extraction methods using a SVM classifier (Table 5). As before, 10% of data is used for training. It is obvious that the proposed classifier outperforms SVM. Again, BCP achieves superior performance, which is attributed to its exploitation of spatial structure in the data

Table 5- Classification accuracies obtained for Indiana using SVM

` *METHODS*	*RAW*	*NMF*	*BETA-NTF*	*BCP*
AVERAGE ACCURACY(%)	64.92	37.46	88.10	**88.34**
OVERALL ACCURACY(%)	70.53	45.33	91.01	**92.22**

We have also provided a comparison with results of [28] and [29]. In [28], the accuracy statistics are reported for Pavia dataset against different number of training samples per class (Tr=45,55 and 65). So we do the same thing applying our proposed method with BCP features. The accuracy statistics listed in Table 6 compare CNN [28], DCNN[28] and DHCNET[28] outcomes with our proposed method.

Table 6. Accuracy statistics obtained for Pavia against different number of Training samples per class

	Tr=45			Tr=55			Tr=65		
Method	OA	AA	Kappa	OA	AA	Kappa	OA	AA	Kappa
CNN [28]	96.53	96.82	95.42	97.09	97.36	96.16	97.43	97.80	96.60
DCNN [28]	96.97	97.30	96.00	97.65	97.83	96.89	98.17	98.29	97.58
DHCNET [28]	97.37	97.70	**96.52**	**97.95**	**98.03**	**97.28**	**98.44**	98.54	**97.93**
BCP-3D CNN (proposed)	**97.47**	**97.78**	**96.65**	97.62	97.97	97.01	98.14	**98.58**	97.42

In Table 7, we have compared the performance of the proposed techniques in [29] with our method for Indiana data with 10 percent training. Class accuracies reported in [29] applying Multi-Bias Network (MBN) and Squeeze Multi-Bias Network (SMBN) are repeated here.

Table 7. Individual class accuracies obtained for Indiana with 10% training

method Class	MBN	SMBN	BCP-3D-CNN(Proposed)
1	92.74	92.68	**100**
2	**99.10**	98.83	96.23
3	99.77	**99.87**	98.52
4	100	100	100
5	97.87	97.47	**99.36**
6	100	**100**	99.58
7	100	100	100
8	100	**100**	99.68
9	**100**	96.23	92.31
10	**99.51**	98.52	98.26
11	99.35	**100**	97.49
12	95.78	**99.36**	98.96
13	**100**	99.58	**100**
14	99.55	**100**	99.15
15	97.72	99.68	**100**
16	98.56	92.31	**100**
OA	**98.87**	98.26	98.33
AA	**98.75**	97.49	98.72
Kappa	**99.09**	98.96	98.10

5 Conclusion

In this study, we propose a classification framework for HSI data. A 3D CNN architecture is developed for this purpose. Effective feature extraction methods based on NTF techniques are suggested. We have shown that exploiting the abundance maps acquired through NTF methods as training features, a 3D CNN can lead to superior classification performance compared to raw or BNMF features. This happens due to the fact that NTF preserves the spatial structure of HSI. We also proposed to apply morphological attribute filters to improve the discriminatory property of

the training features which also effectively increases the size of the feature vectors. The 3D CNN also compares favorably compared to a SVM.

References

[1] J. A. Richards, Remote Sensing Digital Image Analysis: An Introduction, Springer Berlin Heidelberg, NJ, USA, 5th edition, 2013.

[2] Hyperspectral Imaging Remote Sensing: Physics, Sensors, and Algorithms 1st Edition ,Dimitris G. Manolakis, Ronald B. Lockwood, Thomas W. Cooley, Cambridge University Press 2016.

[3] J. Bioucas-Dias et al., "Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches," IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 5, no. 2, pp. 354–379, April 2012.

[4] X. Lu, H. Wu, Y. Yuan, P. Yan, and X. Li, "Manifold regularized sparse nmf for hyperspectral unmixing," IEEE Transactions on Geoscience and Remote Sensing, vol. 51, no. 5, pp. 2815–2826, May 2013.

[5] L. Tong, J. Zhou, X. Li, Y. Qian, and Y. Gao, "Region-based structure preserving nonnegative matrix factorization for hyperspectral unmixing," IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 10, no. 4, pp. 1575–1588, April 2017.

[6] A. Villa, J. Chanussot, J. A. Benediktsson and C. Jutten, "Spectral Unmixing for the Classification of Hyperspectral Images at a Finer Spatial Resolution," in IEEE Journal of Selected Topics in Signal Processing, vol. 5, no. 3, pp. 521-533, June 2011.
doi: 10.1109/JSTSP.2010.2096798

[7] He, Zhi, Lin Liu, Ruru Deng, and Yi Shen. "Low-rank group inspired dictionary learning for hyperspectral image classification." Elsevier Signal Processing 120 (2016): 209-221.

[8] Zhong, Zilong, Jonathan Li, Zhiming Luo, and Michael Chapman. "Spectral–Spatial Residual Network for Hyperspectral Image Classification: A 3-D Deep Learning Framework." IEEE Transactions on Geoscience and Remote Sensing 56, no. 2 (2018): 847-858.

[9] Leng, Jiabing, Tao Li, Gang Bai, Qiankun Dong, and Han Dong. "Cube-CNN-SVM: A Novel Hyperspectral Image Classification Method." In Tools with Artificial Intelligence (ICTAI), 2016 IEEE 28th International Conference on, pp. 1027-1034. IEEE, 2016.

[10] Xu, Yonghao, Bo Du, Fan Zhang, and Liangpei Zhang. "Hyperspectral image classification via a random patches network." ISPRS Journal of Photogrammetry and Remote Sensing (2018).

[11] Imani, Maryam, and Hassan Ghassemian. "Discriminant analysis in morphological feature space for high-dimensional image spatial–spectral classification." *Journal of Applied Remote Sensing* 12, no. 1 (2018): 016024.

[12] W. Hu, Y. Huang, W. Li, F. Zhang, and H. Li, "Deep convolutional neural networks for hyperspectral image classification," J. Sensors, vol. 2015, Jan. 2015, Art. no. 258619.

[13] P. Ghamisi, Y. Chen and X. X. Zhu, "A Self-Improving Convolution Neural Network for the Classification of Hyperspectral Data," IEEE Geoscience and Remote Sensing Letters, vol. 13, no. 10, pp. 1537-1541, Oct. 2016.

[14] J. Li, X. Zhao, Y. Li, Q. Du, B. Xi and J. Hu, "Classification of Hyperspectral Imagery Using a New Fully Convolutional Neural Network," IEEE Geoscience and Remote Sensing Letters, vol. 15, no. 2, pp. 292-296, Feb. 2018.

[15] O. Dikmen, C. Fevotte, “Maximum Marginal Likelihood Estimation for Nonnegative Dictionary Learning in the Gamma-Poisson Model,” IEEE Transactions on Signal Processing, vol.60, no. 10, pp. 5163-5175, Oct. 2012.
[16]Algorithms for non-negative tensor factorization, Markus Flatz, Technical report 2013-05, Salzburg University, Nov. 2013.
[17] Jingu Kim, Yunlong He, and Haesun Park. Algorithms for Nonnegative Matrix and Tensor Factorizations: A Unified View Based on Block Coordinate Descent Framework. Journal of Global Optimization, 58(2), pp. 285-319, 2014.
[18] Chi, Eric C., and Tamara G. Kolda. "On tensors, sparsity, and nonnegative factorizations." SIAM Journal on Matrix Analysis and Applications 33, no. 4 (2012): 1272-1299.
[19] M. D. Mura, J. A. Benediktsson, B. Waske, and L. Bruzzone,“Extended profiles with morphological attribute filters for the analysis of hyperspectral data,” Int. J. Remote Sens., vol. 31, no. 22, pp. 5975–5991, Jul. 2010.
[20] M. D. Mura, J. A. Benediktsson, B. Waske, and L. Bruzzone, “Morphological attribute profiles for the analysis of very high resolution images,” IEEE Trans. Geosci. Remote Sens., vol. 48, no. 10, pp. 3747–3762,
Oct. 2010.
[21] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov;”Dropout: A Simple Way to Prevent Neural Networks from Overfitting
” 15(Jun):1929−1958, 2014.
[22] Prechelt, Lutz. "Early stopping—but when?." Neural networks: tricks of the trade. Springer, Berlin, Heidelberg, 2012. 53-67.
[23] Ruder (2016). An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747
[24] Madooei, Ali, Ramy Mohammed Abdlaty, Lilian Doerwald-Munoz, Joseph Hayward, Mark S. Drew, Qiyin Fang, and Josiane Zerubia. "Hyperspectral image processing for detection and grading of skin erythema." In *Medical Imaging 2017: Image Processing*, vol. 10133, p. 1013322. International Society for Optics and Photonics, 2017.
[25] Boubanga-Tombet, Stephane, Alexandrine Huot, Iwan Vitins, Stefan Heuberger, Christophe Veuve, Andreas Eisele, Rob Hewson, Eric Guyot, Frédérick Marcotte, and Martin Chamberland. "Thermal Infrared Hyperspectral Imaging for Mineralogy Mapping of a Mine Face." *Remote sensing* 10, no. 10 (2018): 1518.
[26] Mahesh, S., D. S. Jayas, J. Paliwal, and N. D. G. White. "Hyperspectral imaging to classify and monitor quality of agricultural materials." *Journal of Stored Products Research* 61 (2015): 17-26.
[27] Gao, Qishuo, Samsung Lim, and Xiuping Jia. "Hyperspectral image classification using convolutional neural networks and multiple feature learning." *Remote Sensing* 10, no. 2 (2018): 299.
[28] Zhu, Jian, Leyuan Fang, and Pedram Ghamisi. "Deformable convolutional neural networks for hyperspectral image classification." *IEEE Geoscience and Remote Sensing Letters* 15, no. 8 (2018): 1254-1258.
[29] Fang, Leyuan, Guangyun Liu, Shutao Li, Pedram Ghamisi, and Jón Atli Benediktsson. "Hyperspectral Image Classification With Squeeze Multibias Network." *IEEE Transactions on Geoscience and Remote Sensing* 99 (2018): 1-11.
[30] Yang, Xiaofei, Yunming Ye, Xutao Li, Raymond YK Lau, Xiaofeng Zhang, and Xiaohui Huang. "Hyperspectral image classification with deep learning models." *IEEE Transactions on Geoscience and Remote Sensing* 56, no. 9 (2018): 5408-5423.

[31] Khoshdeli, Mina, and Bahram Parvin. "Feature-based representation improves color decomposition and nuclear detection using a convolutional neural network." *IEEE Transactions on Biomedical Engineering* 65, no. 3 (2018): 625-634.
[32] Luong, Thanh Xuan, Bo-Kyeong Kim, and Soo-Young Lee. "Color image processing based on nonnegative matrix factorization with convolutional neural network." In *2014 International Joint Conference on Neural Networks (IJCNN)*, pp. 2130-2135. IEEE, 2014.

View publication stats