VIRTUAL TRAINING SAMPLE GENERATION BY GENERATIVE ADVERSARIAL NETWORKS FOR HYPERSPECTRAL IMAGES CLASSIFICATION

T. Alipourfard[1], H. Arefi [1, *]

[1] School of Surveying and Geospatial Engineering, University of Tehran, Tehran, Iran - (tayebalipour, hossein.arefi)@ut.ac.ir

KEY WORDS: Deep Learning, Hyperspectral Image, Convolutional Neural Network, Generative Adversarial Networks

ABSTRACT:

Convolutional Neural Networks (CNNs) as a well-known deep learning technique has shown a remarkable performance in visual recognition applications. However, using such networks in the area of hyperspectral image classification is a challenging and time-consuming process due to the high dimensionality and the insufficient training samples. In addition, Generative Adversarial Networks (GANs) has attracted a lot of attentions in order to generate virtual training samples. In this paper, we present a new classification framework based on integration of multi-channel CNNs and new architecture for generator and discriminator of GANs to overcome Small Sample Size (SSS) problem in hyperspectral image classification. Further, in order to reduce the computational cost, the methods related to the reduction of subspace dimension were proposed to obtain the dominant feature around the training sample to generate meaningful training samples from the original one. The proposed framework overcomes SSS and overfitting problem in classifying hyperspectral images. Based on the experimental results on real and well-known hyperspectral benchmark images, our proposed strategy improves the performance compared to standard CNNs and conventional data augmentation strategy. The overall classification accuracy in Pavia University and Indian Pines datasets was 99.8% and 94.9%, respectively.

1. INTRODUCTION

Hyperspectral sensors provide valuable information from the surface of the earth including more than hundreds of image bands in visible and infrared regions of electromagnetic spectrum at a certain spatial resolution. This rich cube of data creates an opportunity to detect and recognize different objects on hyperspectral images. Image classification is one the major processes which is often applied to the hyperspectral images for information extraction purposes (Plaza et al., 2009). Some general challenges are still available in the area of hyperspectral image classification with high consideration, such as high dimensionality of the data, the problem related Small Sample Size (SSS), correlation of spectral signature among different objects in the desired scene, uncertainties and consideration of spatial information during the classification process (Plaza et al., 2009, Li et al., 2012, Melgani et al., 2004). Deep learning (DL), as a novel machine learning method, has achieved state-of-the-art performance in many applications such as object recognition (LeCun et al., 2015), handwritten digit recognition (LeCun et al., 1989), natural language processing (Bordes et al., 2012), image classification and segmentation (Krizhevsky et al., 2012, Szegedy et al., 2013).

Among different neural network architectures, Convolutional Neural Networks (CNNs) have shown remarkable performance in processing images, due to their powerful learning strategy. In hyperspectral image processing, the CNNs are able to extract meaningful features automatically by training data and performing an end-to-end classification procedure. Thus, users are not required to manually select the relevant features (Ghamisi et al., 2016, Leng et al., 2016). Despite many advantages of the CNNs, there are critical problems with respect to the learning parameters in the CNN architecture. A large number of training data are needed to achieve robust classification results due to a huge number of training parameters in models such as GoogleNet and VGGNet (Canziani et al., 2017)

To overcome SSS problem, Generative Adversarial Networks (GANs) provide a unique way to train deep learning algorithms to create training data from the existing training samples. First introduced in 2014 by Goodfellow, GANs consist of two competing models called “generator” and “discriminator”. Genrator takes noise as the input and generates samples. “Discriminator” receives the samples from both generator and training data, and distinguishes between the two sources. These two networks play a continuous game, where the generator is learning to produce more and more realistic samples, and the discriminator is learning to get better and better at distinguishing the generated data from real data (Goodfellow et al., 2014). Zhu et al. pioneered to use GAN for hyperspectral data processing by proposing spectral and spectral-spatial GAN. However, they used large amounts of training data (e.g. about 50% of the whole image in Indian Pines dataset as a training sample). In addition, the proposed architecture for discriminator and generator is designed in such a way (small number of parameters) that it can be consistent with the limitation of training samples (Zhu et al., 2018). On the other hand, dimension reduction methods are one of the most effective tools for overcoming time-consuming and SSS problems which makes hyperspectral image classification tractable. During the recent years, a large body of research were conducted to perform the dimension reduction of hyperspectral images in order to overcome SSS problems (Plaza et al., 2009, Li et al., 2012, Alipour et al., 2014, 2018). Most of these studies used Principle Component Analysis (PCA) method to reduce the number of hyperspectral bands (Ghamisi et al., 2016, Chen et al., 2016, Zhao et al., 2016, Romero et al., 2016). The previous studies which applied CNNs on hyperspectral images could only perform three channels such as three components of PCA, resulting in reducing the separation of classes in hyperspectral images (Ghamisi et al., 2016, Chen et al., 2016, Zhao et al., Li et al., 2017, Canziani et al., 2017). The present study was conducted by considering the advantages of GAN and subspace-based reduction method and proposed a framework called “Generative Adversarial Convolutional Neural Network (GACNN)” based on CNNs. The proposed framework generate virtual training samples for hyperspectral image classification. In order to perform GACNN on real training sample, we have designed generator and discriminator designed based on CNNs to overcome SSS problem and high dimensionality of the

[*] Corresponding author

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLII-4-W18-63-2019 | © Authors 2019. CC BY 4.0 License.

hyperspectral images. To this aim, subspace feature extraction of hyperspectral image was implemented to get dominant feature in input patch for each class and accordingly produced by GACNN virtual training samples for each class. The main contributions of this paper are as follows:

- Generating virtual training samples by deep convolutional generative adversarial networks to overcome the lack of training data in hyperspectral images
- Implementing subspace dimension reduction methods to reduce the dimensionality of hyperspectral images and generating high quality training sample for GANs, which more compatible by the nature of class based CNNs and the logistic regression as a classifier in the last layer
- Designing fixed architecture of multichannel convolutional neural network for classifying hyperspectral images.

The results of study can be considered as an improvement regarding the accuracy of hyperspectral image classification. In addition, the results can confirm an improvement in the quality of hyperspectral classification map.

2. METHODOLOGY

As shown in figure 1, the general flow of the proposed framework consists of three major steps including dimension reduction, generating virtual training sample from real training data by adversarial network technics, and hyperspectral image classification of convolutional neural network. The details of the proposed method are given in the following sections.

2.1 Dimension Reduction

Subspace based on reduction methods project a high dimensional data (L) to a lower dimension (l) by calculating orthonormal basis for each class. The most common way to define a dimensional subspace L is to use a set of linearly independent basis vectors $\{u_1, \ldots, u_L\}$, which can be combined into a d×L matrix U which has rank L. Suppose $U^{(c)}$ is a set of $r^{(c)}$-dimensional orthonormal basis vectors for the subspace associated with classes c = 1, 2,. . . k. Following (Li et al., 2012), $U^{(c)}$ is computed as $U^{(c)} = \{e_1^{(c)}, \ldots, e_{r^{(c)}}^{(c)}\}$, while $E^{(c)} = \{e_1^{(c)}, \ldots, e_d^{(c)}\}$ is the eigenvector matrix computed from correlation matrix $R^{(c)} = E^{(c)} \Lambda E^{(c)'}$. Here, Λ is the eigenvalue matrix with a decreasing magnitude. We use a subspace projection accounting for 99.9% of the original spectral information in order to determine the size of $U^{(c)}$. Feature space defined by the nonlinear functions while X is a 2D matrix of hyperspectral image $h(x_i) = \left[\|x_i\|^2, \|x_i^T U^1\|^2, \ldots, \|x_i^T U^k\|^2\right]^T$ while X_i is a 2D matrix of hyperspectral image (Li et al., 2012).

2.2 Generative Adversarial Training Sample

The idea of GAN comes from the game between two players in which one player is generator (G), is responsible for producing training sample and another one is discriminator (D), which is used to determine whether they are real or fake (Radford et al., 2016). The GAN framework should define two components including discriminator, generator and architecture, as well as training process. The definition of architecture is regarded as the first and important step for design a GAN framework. The discriminator should be trained on real training sample of each class and be locked in advance. Then generator, generates fake image from random variable z and this virtual image (train sample) feeds into discriminator to predict them as fake correctly. Following (Goodfellow et al., 2014) and (Radford et al., 2016), a CNN framework was designed for both generator and discriminator in the present study. Training of these networks occur simultaneously and can be described as a minimax game. The Discriminator is trying to maximize its own performance of distinguishing between real and generated samples, while the Generator G is maximizing its ability to generate samples that manage to fool the Discriminator. A more formal definition of GAN's *min-max* game can be described with the following function V:

$$minmax\, V(D, G) = E_{x\sim pdata\,(x)}\,[\log D\,(x)] + E_{z\sim p\,noise\,(x)}\,[\log\,(1 - D\,(G\,(z)))] \quad (1)$$

x is a sample from the training dataset which has the probability distribution p_{data}. z is a randomly generated noise sample from the distribution p_{noise}. This is often a list of random numbers in range -1 to 1. E_x and E_z indicates which data distribution the sample belongs to. The Discriminator must classify that the sample belong to the expected distribution. In other words, E_x denotes the sample is a real image and E_z denotes that the sample is generated. *G (z)* represents a sample generated by the Generator model *G* with noise *z* as input.

2.3 Designing Convolutional Neural Network

CNNs are popular due to the generating ability of their automatic meaningful feature, by considering spatial information and share weighting (LeCun et al., 2015). A CNN is a sequence of layers, where each different layer of the CNNs plays a different role by imposing a different function (Krizhevsky et al, 2012). In the present study, main types of layers were employed to build these CNNs architectures as Convolutional Layer, ReLU Layer, Pooling Layer, Dropout Layer, and Fully-Connected Layer. The convolutional layer aims to learn feature representations of the inputs. Convolution layer is composed of several convolution kernels (receptive fields), which are used to compute different feature maps. Specifically, each neuron of a feature map is connected to a region of neighboring neurons in the previous layer. The feature map is computed by equation (2).

$$M_{i,j}^{l+1} = w^{l+1} * x_{i,j}^{l} = \sum_a \sum_b \left(w_{i,j}^{l+1} . x_{i-a,j-b}^{l}\right) + bias_{i,j}^{l+1} \qquad (2)$$

Where M indicates a complex feature map at certain location (i,j) in layer l+1, w represents a kernel and a, b are regarded the size of kernel. ReLU is computed after the convolution and accordingly a nonlinear activation functions like TanH or sigmoid. ReLU computes the absolute value for each component of the feature maps by the equation (3).

$$A_{ij}^{l+1} = \max(0, M_{i,j}^{l+1}) \qquad (3)$$

In Eq. (4), A indicates a feature map after applying ReLU operator. Robustness to noise and distortions are regarded as the main motivation of pooling the feature maps obtained by previous layers. Max pooling is used to get faster convergence during training with respect to other strategy like average pooling. Dropout is a technique for addressing overfitting problem in CNNs. The concept of dropout is very simple but very effective to prevent the network from overfitting by removing half of the weight parameters. The output of dropout layers is computed by the equation (5) as follows:

$$y = r.\,(W^T x) \;\rightarrow\; r_i \sim Bernoulli\ Distribution\,(p) \quad (4)$$

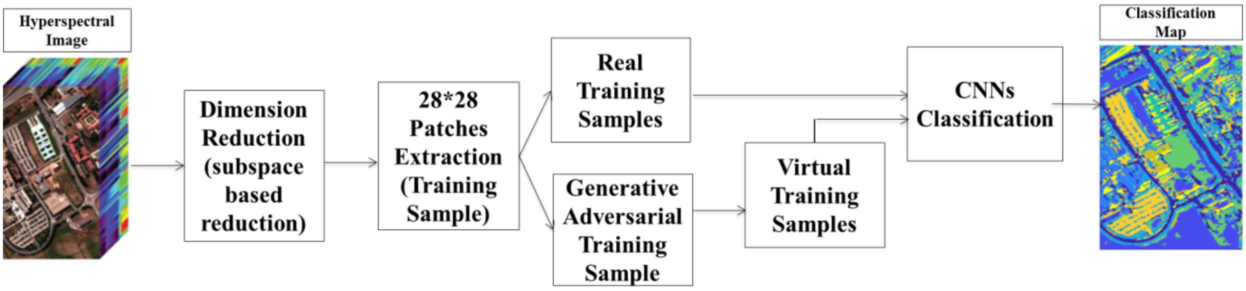

Hyperspectral Image
Dimension Reduction (subspace based reduction)
28*28 Patches Extraction (Training Sample)
Real Training Samples
Generative Adversarial Training Sample
Virtual Training Samples
CNNs Classification
Classification Map

Figure 1. General flow of the proposed framework for hyperspectral image classification

Where x indicates unfold data $x = \{x_1, x_2, \ldots, x_n\}$ which input to fully connected layer. Following dropout layer, the traditional neural network layer is embedded as a fully-connected layer to perform mapping feature domain to class domain. After max pooling (also dropout), the feature map should be unfolded and fully connected to each class. In hyperspectral image classification, an increase in the number of fully-connected layer leads to a significant increase in the number of parameters. Regarding a classification problem with more than two classes, the output unit activation function is the softmax function as follows:

$$p(c_r|x,\theta) = \frac{p(x,\theta|c_r)p(c_r)}{\sum_{j=1}^{k} p(x,\theta|c_j)p(c_j)} = \frac{\exp(y_r)}{\sum_{j=1}^{k} y_j} \quad (5)$$

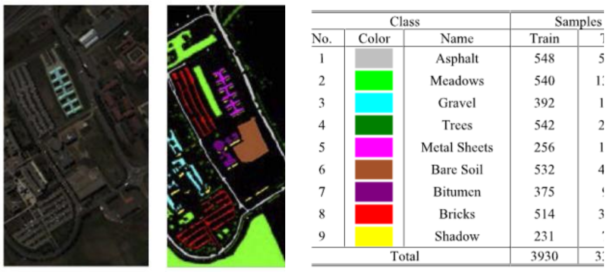
In addition, the softmax output layer provides the assigning probabilities to each class and then a metric is necessary to measure the similarity between output of network (highest probability in previous layer) and label of each pixel (one-hot vector) to perform a backpropagations procedure. The loss (error) function is the cross entropy function for 1-of-k classes.
Classification layer is the last layer in CNNs and Stochastic Gradient Descent (SGD) algorithm updates the parameters θ of objective function (θ) in an iterative manner. Thus, these layers were piled to form a full CNNs architecture in the present study. Additionally, the transition between the introduced layers should have been manipulated by Rectified Linear Unit (ReLU), as an activation and dropout function to reduce the effect of gradient vanishing and overfitting problems, respectively.

3. EXPERIMENTAL RESULTS

In this study, the efficiency and sensitivity of the proposed framework were evaluated based on the virtual training sample generation and dimension reduction method. Since we mainly focused on generating virtual sample and dimension reduction problems, the designed experiment was compared by the state of the art CNNs, which focused on two mentioned problems and successfully applied to hyperspectral image classification during recent years. Three papers were selected from literature to examine the performance of proposed method. The first method was introduced in (Chen et al, 2016) with changing radiation-based virtual samples (Chen's Method), while the second method was proposed in (Ghamisi et al., 2016) with metaheuristic dimension reduction method (Ghamisi's Method). Further, the third method that introduced in (Paoletti et al. 2017) was a 3D-CNN framework to hyperspectral image classification (Paoletti's Method). The performance of each classification procedure was measured by Overall Accuracy (OA), Average Accuracy (AA), as well as Kappa coefficient (K).

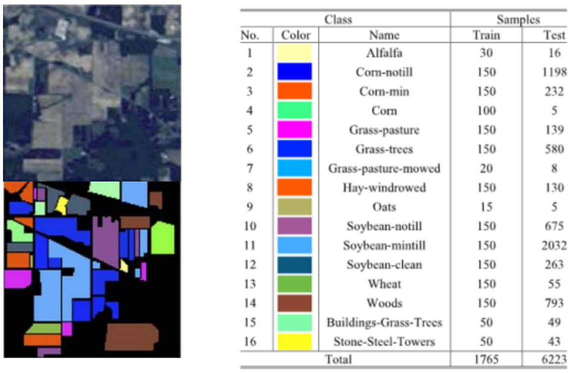
3.1 Data Description

The proposed method was implemented on two well-known hyperspectral images from Pavia University and Indian Pines areas. The Pavia University is related to the Engineering School at the University of Pavia captured by ROSIS-03. In the experiments, 12 noisy data channels were eliminated, and 103 data channels were used for processing. The Indian Pines was captured by AVIRIS over a rural area in NW Indiana. For this data set, 200 data channels were used after removing the spectral bands affected by atmospheric absorption and noise. The total numbers of training and test samples for Pavia University were 3930 and 33940, respectively, while they were 1765 and 6223, respectively, for Indian Pines (Figure 2, 3).



Class			Samples	
No.	Color	Name	Train	T
1	⬜	Asphalt	548	5
2	🟩	Meadows	540	1:
3	🟦	Gravel	392	1
4	🟩	Trees	542	2
5	🟪	Metal Sheets	256	1
6	🟫	Bare Soil	532	4
7	🟪	Bitumen	375	
8	🟥	Bricks	514	3
9	🟨	Shadow	231	
Total			3930	3:

Figure 2: Pavia University Data Set and number of train and test samples



Class			Samples	
No.	Color	Name	Train	Test
1	🟨	Alfalfa	30	16
2	🟦	Corn-notill	150	1198
3	🟧	Corn-min	150	232
4	🟩	Corn	100	5
5	🟪	Grass-pasture	150	139
6	🟦	Grass-trees	150	580
7	🟦	Grass-pasture-mowed	20	8
8	🟧	Hay-windrowed	150	130
9	🟫	Oats	15	5
10	🟪	Soybean-notill	150	675
11	🟦	Soybean-mintill	150	2032
12	🟦	Soybean-clean	150	263
13	🟩	Wheat	150	55
14	🟫	Woods	150	793
15	🟩	Buildings-Grass-Trees	50	49
16	🟨	Stone-Steel-Towers	50	43
Total			1765	6223

Figure 3: Indian Pine Data Set number of train and test samples

3.2 Investigation of Generative Adversarial Convolutional Neural Network (GACNN)

The Discriminator *D*'s task is to rate images on how natural they appear and output a scalar value. Ideally, an image from the training set, a real image, should get a good rating while an image that is generated from *G* should get a poor rating. During training, the Discriminator network is shown different images from both the training set and samples from *G* and taught how it should rate them. This is then used to adjust the weights and improve the Discriminator by slowly modifying what features are used to spot fake images. These adjustments are supplied back to the Generator and in turn used to further improve image generation. The Generator *G* generates its samples given an input *Z*, which is generally an image consisting of randomly generated noise. The network will try to recognize patterns in the noise and apply elements of previously seen images, resulting in an image that should look natural (Fig.4).The maximum probability output from the discriminator is 0.5 (Goodfellow et al., 2014), those virtual samples with a probability of more than 0.45 were selected during iterations (0.45<probability<0.5). Table I indicates the number of passable epochs, as well as virtual training samples generated by GAN process for each dataset. The number of virtual training samples for each class is different and related to the number of real (original) training samples and the number of epochs with high probabilities. Fig.4 illustrates a set of 64 virtual training samples generated by GAN in class tree in Pavia University dataset.

Data set	# Real training samples	# Total epochs	# Epochs (0.45<probability<0.5)	# Samples generated by GAN
PU	3930	50	38	21888
IP	1765	50	20	20480

Table 1. Number of virtual training samples generation by GAN (PU: Pavia University, IP: Indian Pine)

During the training phase, virtual and real training samples were integrated and then the optimization (learning parameters) of the CNN was conducted by the Adam optimization strategy in order to determine the parameters of weights and bias in the hidden layers, learning rate, kernel size, and the number of the convolution layers. In the present experiments, Table II presents the architecture for CNN, discriminator and generator determined by sensitivity analysis and the best architecture in term of OA. The size of input patches for CNN and GAN are 28×28 and normalized into [-1 1]. The learning rate was set to 0.005, and the number of epochs was set to 500 for proposed CNN in the dataset from Indian Pines. Regarding the University of Pavia dataset, the learning rate was sent to 0.01, and the number of epochs reached to 400. Based on the results, ReLU could play a critical role in achieving good robustness. Unlike sigmoid and TanH, ReLU failed to map the convolution output between 0, 1 (-1, 1) and palliate gradient vanishing in primary layers. During the present subspace based on reduction method, the number of features were equal to the number of classes in each scene. To this aim, 9 and 16 features were employed from the dataset for the University of Pavia and the Indian Pines, respectively. Each feature represents the specific class, which is a suitable choice for CNN learning procedure, because the training samples of each class include high quality. During the testing phase, test pixels were classified with the parameters obtained during the training phase. Regarding each test pixel in the Pavia University and Indian Pines dataset, 9 and 16 labels were respectively determined and then winning class was determined by a majority of voting strategies.

Tables 3 and 4 indicate the results of implementing the proposed GACNN methods, which could provide the best performances of OA, AA, and Kappa for the two datasets.

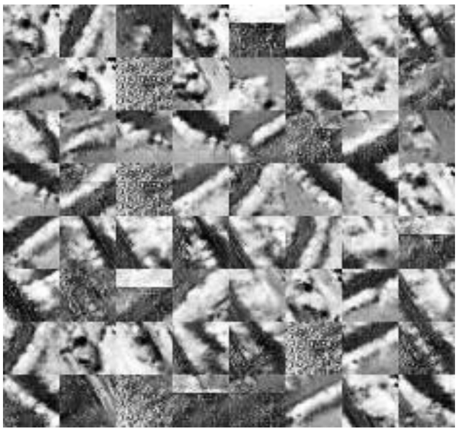


Figure 4.Example of generating 64 virtual training samples by GAN in class #4 in Pavia University Dataset (Tree)

Name	Convolution Layer	ReLU/ Leaky ReLu	Pooling/ Deconvolution	Dropout
CNN	(L1) 5×5×32	Yes	Pooling 2×2	No
	(L2) 3×3×64	Yes	Pooling 2×2	50%
Discriminator	(L1) 7×7×64	No	Pooling 2×2	No
	(L2) 4×4×128	No	Pooling 2×2	No
Generator	(L1) 4×4×128	No	Deconvolution	No
	(L2) 7×7×64	No	Deconvolution	No

Table 2.Proposed architecture for CNN, discriminator, and generator

Table 3 demonstrates the experimental results for the University of Pavia dataset. It is evident that the proposed method could provide better results again, which outperformed Chen’s method by 2.63%, 2.85%, and 0.027 in terms of OA, AA, and K, respectively. However, it is worth noting that the obtained variance is very small. The proposed architecture shows the best classification results in the Indian Pines dataset.

In terms of visual analysis, the quality of classification is highly remarkable, because of generating virtual sample by GAN and subspace base reduction in a same framework. (Fig. 5, 6).

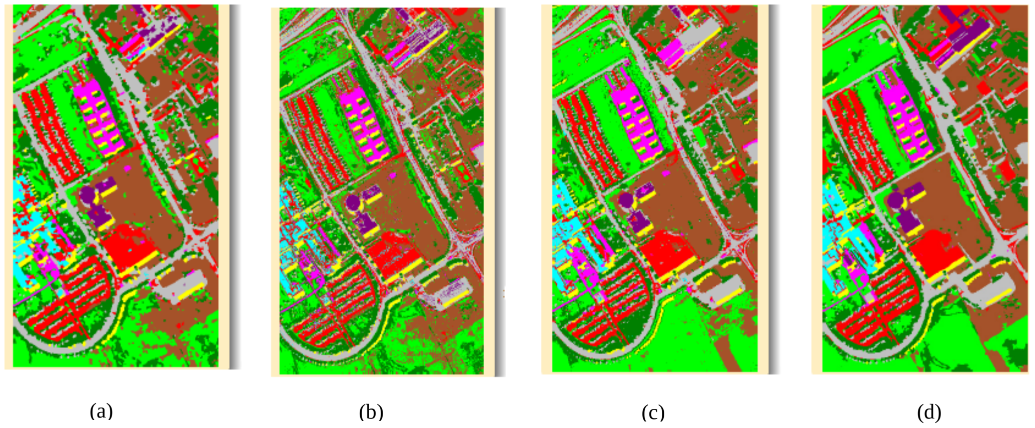
In order to evaluate the performance of the present CNN architecture, a hardware was used including a 6th Generation Intel(R) Core(TM) i7-6800 K processor with 6MB of Cache, 32 GB of DDR4 RAM, a GPU NVIDIA GeForce GTX 1080 Ti with 8 GB RAM and an ASUS motherboard. Tables 5 and 6 indicated the processing time for training and overall accuracy under different number of bands (reduced dimension). In the proposed methods, the processing time reduced by 20% and 30% for Indian Pines and Pavia University, respectively, due to light architecture and appropriate selection of dimension reduction method, compared with considering original bands to classification.

Classification Method	Chen's Method	Ghamisi's Method	Paoletti's Method	GACNN (Proposed Method)		
DR Method	PCA	PSO	Original (Raw)	Original (Raw)	Subspace (without virtual training data)	Subspace (with virtual training data)
# Input Features	3	10	103	103	9	
# Train Sample	3930	3930	1800	3930	3930	3930
# Test Sample	33940	33940	36070	33940	33940	33940
OA (%)	97.21	90.23	93.18	90.02	94.14	**99.84**
AA (%)	96.95	89.91	93.12	90.16	93.94	**99.80**
K×100	96.91	87.41	93.01	88.00	92.81	**99.65**
Training Phase Run Time (min.)	48.2	50.3	128.3	106.3	30.9	34.2
Test Phase Run Time (min.)	2.5	2.9	10.1	9.2	2.0	2.3

Table 3.Classification results on the University of Pavia dataset

Classification Method	Chen's Method	Ghamisi's Method	Paoletti's Method	GACNN (Proposed Method)		
DR Method	PCA	PSO	Original (Raw)	Original (Raw)	Subspace (without virtual training data)	Subspace (with virtual training data)
# Input Features	3	10	200	200	16	16
# Train Sample	1765	1765	2466	1765	1765	1765
# Test Sample	6223	6223	5522	6223	6223	6223
OA (%)	92.42	86.62	94.93	92.19	94.23	**97.91**
AA (%)	92.14	85.77	93.00	92.81	94.75	**99.27**
K×100	91.87	85.27	92.97	91.19	93.23	**97.91**
Training Phase Run Time (min.)	29.1	32.4	265.3	98.2	20.5	23.5
Test Phase Run Time (min.)	2.1	1.8	25.2	6.3	6.3	1.2

Table 4.Classification results on the Indian Pine dataset



(a) (b) (c) (d)

Figure 5. University of Pavia. (a)– (d) Classification maps for different classifiers: (a) Chen's Method with OA=97.21%, (b) Ghamisi's Method with OA=90.23%, (c) Paoletti's Method with OA=93.18% (d) GACNN (proposed) with OA=99.84%

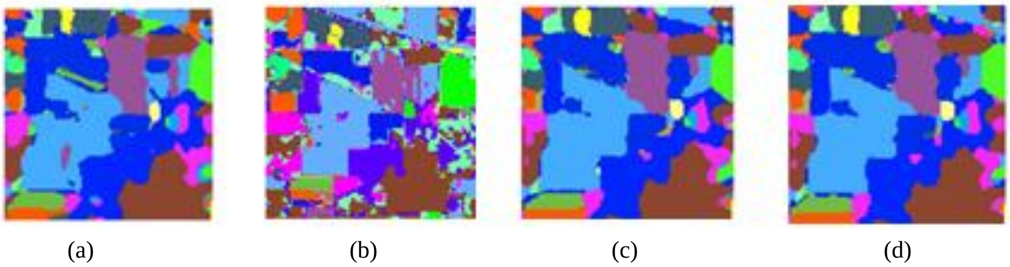


Figure 6. Indian Pine. (a)– (d) Classification maps for different classifiers: (a) Chen's Method with OA=92.42%, (b) Ghamisi's Method with OA=86.62%, (c) Paoletti's Method with OA=94.93% (d) GACNN (proposed) with OA=97.91%

Number of bands	1	2	3	4	5	6	7	8	9	103(Original bands)
Overall accuracy (%)	58.3	65.2	75.0	80.2	86.5	92.0	97.2	98.0	99.8	90.0
Train time (min)	22.4	23.8	25.2	26.6	28.0	29.4	30.8	32.2	34.2	106.3

Table 5. Overall accuracy and training time of the University of Pavia dataset under different number of bands (proposed method)

Number of bands	2	4	6	8	10	12	14	16	200(Original bands)
Overall accuracy (%)	48.5	62.3	80.2	89.2	91.2	92.8	93.2	94.91	85.1
Train time (min)	8.1	10.2	12.2	14.3	16.3	18.3	20.6	23.5	98.2

Table 6. Overall accuracy and training time of the Indian Pines dataset under different number of bands (proposed method)

4. CONCLUSIONS

In the present work, GACNN architectures were proposed by subspace dimension reduction methods for classifying hyperspectral images based on the features with high quality. The advantage of generating virtual samples by GAN and proposed architecture for CNN is equal for the two datasets and the conventional number of training pixels (about 2% of whole image) is appropriate for learning parameters. By fixed architecture for the CNN, generator, and discriminator, the proposed method could provide a remarkable classification performance under the condition of limited training samples.
Future studies can be conducted for more improvement of generalizing the CNN-based methods. In this regard, a boosting method is required since the CNN is considered as a weak learner. Boosting refers to a family of the algorithms, which converts weak learner to the stronger one. Therefore, the generalization of CNN-based classification can be improved based on this kind of motivation and GAN idea.

REFERENCES

Alipour Fard, T. and Arefi, H., 2014, Dimensionality reduction of hyperspectral images by combination of non-parametric weighted feature extraction (NWFE) and modified neighborhood preserving embedding (NPE), *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, XL-2/W3, 31-34.

Alipour Fard, T., Arefi, H. and Mahmoudi, S., 2018, A novel deep learning framework by combination of subspace-based feature extraction and CNN for hyperspectral image classification, *IEEE International Geoscience and Remote Sensing Symposium*, Valencia, Spain.

Bordes, A., Glorot, X., Bengio, Y., Weston, J., 2012, Joint learning of words and meaning representations for open-text semantic parsing, *in Proc. AISTATS*,La Palma, Island ,22.

Canziani, A., Culurciello, E., Paszke, A., 2017, An analysis of deep neural network models for practical applications, *arXiv:1605.07678v4.*

Chen, Y., Jiang, H., Li, C., Jia, X., Ghamisi, P., 2016, Deep feature extraction and classification of hyperspectral images based on convolutional neural networks, *IEEE Transactions on Geoscience and Remote Sensing*, 54(10), 6232-6251.

Ghamisi, P., Chen, Y., Zhu, X.X., 2016, A Self-Improving convolution neural network for the Classification of hyperspectral data, *IEEE Geoscience and Remote Sensing Letters*, 13(10), 1537-1541.
Goodfellow,I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014, A Generative Adversarial Nets, *arXiv: 1406.2661v1.*

Krizhevsky, A., Hinton, G., 2012, ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems 25 (NIPS 2012),* 1–9.

LeCun, Y., Bengio, Y., Hinton, G., 2015, Deep learning, *Nature,* 521(7553), 436–444.

LeCun, Y., Boser, B., Denker, J.S., Henderson, D., 1989, Back propagation applied to handwritten zip code recognition". *Neural Computation,* vol. 1, pp. 541-551, 1989.

https://doi.org/10.5194/isprs-archives-XLII-4-W18-63-2019 | © Authors 2019. CC BY 4.0 License.

Leng, J., Li, T., Bai, G., Dong, Q., Dong, H.,2016, Cube-CNN-SVM: A novel hyperspectral image classification method, *IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI)*, San Jose, CA, pp. 1027-1034.

Li, J., Bioucas-Dias, J.M., Plaza, A., 2012, Spectral–spatial hyperspectral image segmentation using subspace multinomial logistic regression and markov random fields, *IEEE Transactions on Geoscience and Remote Sensing*, 50(3), 809-823.

Li, W., Wu, G., Zhang, F., Du, Q., 2017, Hyperspectral image classification using deep pixel-pair features, *IEEE Transactions on Geoscience and Remote Sensing*, 55(2), 844-853.

Melgani, F., Bruzzone, L., 2004, Classification of hyperspectral remote sensing images with support vector machines, *IEEE Transactions on Geoscience and Remote Sensing*, 42(8), 1778-1790.

Paoletti, M., Haut, J.M., Plaza, J., Plaza, A., 2017, A new deep convolutional neural network for fast hyperspectral image classification, *ISPRS Journal of Photogrammetry and Remote Sensing, https://doi.org/10.1016/j.isprsjprs.2017.11.021.*

Plaza, A., Benediktsson, J., Boardman, J, Brazile, J., Bruzzone, L., Camps-Valls, G., Chanussot, J., Fauvel, M., Gamba, P., Gualtieri, A., Marconcini, M., Tilton, J., Trianni, G., 2009. Recent advances in techniques for hyperspectral image processing. *Remote Sensing of Environment*, 113, 110-122.

Radford, A., Metz, L., Chintala, S., 2016, Unsupervised representation learning with deep convolutional generative adversarial networks, *arXiv: 1511.06434v2.*

Romero, A., Gatta, C., Camps-Valls, G., 2016, Unsupervised deep feature extraction for remote sensing image classification, *IEEE Transactions on Geoscience and Remote Sensing*, 54(3), 1349-1362.

Szegedy, C.,Toshev, A., Erhan, D., 2013, Deep neural networks for object detection, *Advances in Neural Information Processing Systems 26 (NIPS 2013)* 1–9.

Zhao, W., Du, S., 2016, Spectral–spatial feature extraction for hyperspectral image classification: a dimension reduction and deep learning approach, *IEEE Transactions on Geoscience and Remote Sensing*, 54(8), 4544-4554.

Zhu, L., Chen, Y., Ghamisi, P., Benediktsson, J., 2018, Generative adversarial networks for hyperspectral image classification, *IEEE Transactions on Geoscience and Remote Sensing, DOI: 10.1109/TGRS.2018.2805286.*

Xiaorui Ma, Hongyu Wang, Jie Wang, 2016, Semisupervised classification for hyperspectral image based on multi-decision labeling and deep feature learning, *ISPRS Journal of Photogrammetry and Remote Sensing, http://dx.doi.org/10.1016/j.isprsjprs.2016.09.001*

Hao Wu, Saurabh Prasad, 2018, Semi-Supervised Deep Learning Using Pseudo Labels for Hyperspectral Image Classification, *IEEE Transactions on Geoscience and Remote Sensing*, 27(3), 1259-1270

Yan Ju, Lingling Li, Licheng Jiao, Zhongle Ren, Biao Hou, Shuyuan Yang, 2018, Modified Diversity of Class Probability Estimation Co-training for Hyperspectral Image Classification, *arXiv:1809.01436v1*

Shiqi Yu , Sen Jia , Chunyan Xu,2017, Convolutional neural networks for hyperspectral image classification, Neurocomputing, 219, 88–98

Lloyd Windrim , Arman Melkumyan, Richard J. Murphy, Anna Chlingaryan, Rishi Ramakrishnan, 2017, Pretraining for Hyperspectral Convolutional Neural Network Classification, *IEEE Transactions on Geoscience and Remote Sensing, Digital Object Identifier 10.1109/TGRS.2017.2783886*

Essa Basaeed , Harish Bhaskar, Mohammed Al-Mualla, 2016, Supervised remote sensing image segmentation using boosted convolutional neural networks, Knowle dge-Base d Systems, http://dx.doi.org/10.1016/j.knosys.2016.01.028

Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, Mu Li, 2018, Bag of Tricks for Image Classification with Convolutional Neural Networks, *arXiv:1812.01187v2*

Xueyuan Dai , Xiaofeng Wu ,Bin Wang,Liming Zhang, 2019, emisupervised Scene Classification for Remote Sensing Images: A Method Based on Convolutional Neural Networks and Ensemble Learning, *IEEE Transactions on Geoscience and Remote Sensing Letters,* doi: 10.1109/LGRS.2018.2886534

X. Zhang, Q. Song, R. Liu, W. Wang and L. Jiao, "Modified Co-Training With Spectral and Spatial Views for Semisupervised Hyperspectral Image Classification," in *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 6, pp. 2044-2055, June 2014.