CNN HYPERSPECTRAL IMAGE CLASSIFICATION USING TRAINING SAMPLE AUGMENTATION WITH GENERATIVE ADVERSARIAL NETWORKS



Victor-Emil Neagoe[1], Paul Diaconescu[1]

[1]Department of Applied Electronics and Information Engineering,

Polytechnic University of Bucharest, Bucharest, Romania

Abstract

A big challenge for hyperspectral image recognition is to perform pixel classification when not enough hyperspectral labeled training pixels are available.

In this research we have built Generative Adversarial Networks (GANs) that generate additional virtual training hyperspectral pixels based on features extracted from the originally labeled pixels belonging to training dataset.

Abstract

The experiments show a better performance of pixel classification for a classifier based on Deep Convolutional Neural Networks (DCNNs) using GANs for training sample augmentation versus the performance of the DCNN classifier without GAN augmentation.

The score of 95.32% correct classification using DCNN classifier with GANs versus the score of 92.94% of DCNN classifier without GANs, both achieved on Pavia University dataset, proves the obvious advantage of the presented approach.

Overview

I. Introduction
II. Proposed Method
III. Experimental Results
IV. Conclusions

I. Introduction

Introduction - Hyperspectral Image Classification

Hyperspectral imagining is a component of image computing field that includes the collection and processing of images across electromagnetic spectrum. The objective of hyperspectral image recognition is the object detection through classification of the pixels.

The information content of hyperspectral images with hundreds of channels allows us to remotely identify ground materials, based on their spectral signature.

Each hyperspectral image pixel is a high dimensional vector corresponding to the sampling of the incoming radiance at this pixel.

Introduction - Hyperspectral Image Classification

Classification of hyperspectral images is a very challenging task due to the generally unfavorable ratio between the large number of spectral bands and the limited number of training samples available a priori, which results in the Hughes phenomenon.

The sample size is often limited and insufficient for most classifiers to give satisfying performance.

Introduction – Deep Convolutional Neural Networks

Deep Convolutional Neural Networks (DCNNs) are typical feedforward neural networks, which are applied back propagation algorithms to adjust the parameters (weights and biases) of the network in order to reduce the value of the training loss.

DCNNs are optimized for solving computer vision problems like classification and object detection. Their most important layers are the convolutional layers which serve as features extractors, and thus they learn the feature representations of their input images.

Introduction – Deep Convolutional Neural Networks

DCNNs are identifying the most significant features and produce a hierarchy of visual representations used to recognize input patterns associated with each class. Still, the use of a limited number of elements for DCNN training has been corelated with overfitting.

Different techniques are used against overfitting: dropout and batch normalization being the most used, a third often used method involves the generation of new training elements through various mathematical transformations applied to the original training elements.

Introduction - Generative Adversarial Networks

Generative Adversarial Networks belong to the set of generative models. It means that they are able to produce new content. Outside of producing new content, GANs have been also combined with other neural networks to improve their performances.

In our research we have used GANs to generate additional training elements necessary to increase the performance of a DCNN classification.

Introduction - Generative Adversarial Networks

Generative Adversarial Networks are used to automatically generate new elements with the same class characteristics (as opposed to generating new elements with the same element characteristics) as the original training elements.

Interestingly, the class characteristics are built by the GAN by constructing a general knowledge while receiving information from each element.

Introduction - Generative Adversarial Networks

For the hyperspectral pixel classifier, we use GAN based on spatial-contextual information, taking into consideration both the interband and intraband pixel correlation.

The experimental results of the proposed model are given for a hyperspectral image dataset: Pavia University.

II. Proposed Model

Proposed Method - Generative Adversarial Networks

The element generation is done by an ensemble of two networks, one of them called "*the Generator*" and the other called "*the Discriminator*". These networks are unsupervised networks and the accuracy of their work is evaluated by the number of errors they make.

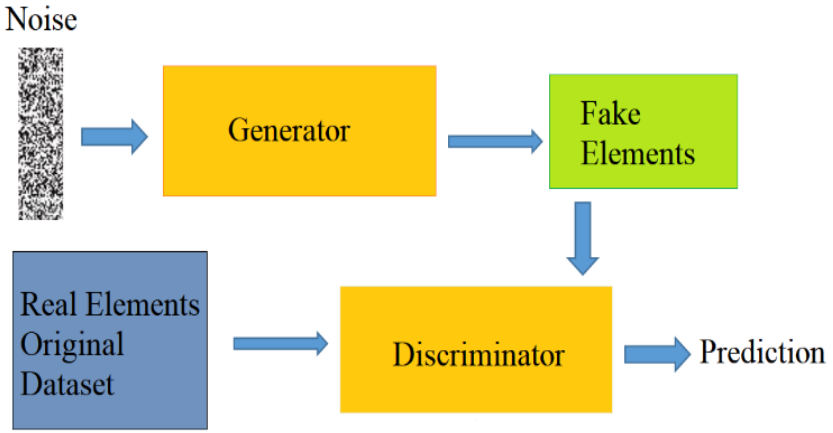


Figure 1. The architecture of a Generative Adversarial Network

Proposed Method - Generative Adversarial Networks

The *Generator* has the role to create new elements with the same characteristics as existing class of elements.

The *Discriminator* tries to evaluate if the newly created elements are genuine or false.

If a correct prediction is made by *Discriminator* (prediction that the element is false), the *Generator* will update its gradients to do a better job. If the prediction made by *Discriminator* is incorrect, the *Discriminator* will have to update its gradients in order to do a better job in future.

Proposed Method - GAN Architecture

The relation between Generator and Discriminator can be summarized as:

- The Generator transforms random numbers in a image.
- This generated image is sent into the discriminator alongside a set of images taken from the original dataset
- The Discriminator takes in both real and fake images and returns probabilities, a number between 0 and 1, with 1 representing a prediction of authenticity and 0 representing fake.

Proposed Method - Generative Adversarial Networks

A calibration between the *Generator* and *Discriminator* includes a few iterations of predictions made by each of them.

GAN training is considered complete when the *Generator* and *Discriminator* start to receive a similar number of corrections from each other.

Proposed Method - GAN Architecture

The *Generator* includes five transposed convolutional layers, the first four of them (in beige and orange colors) being each followed by a batch normalization layer and a ReLU layer, and the last convolutional layers being followed only by a Tanh layer.

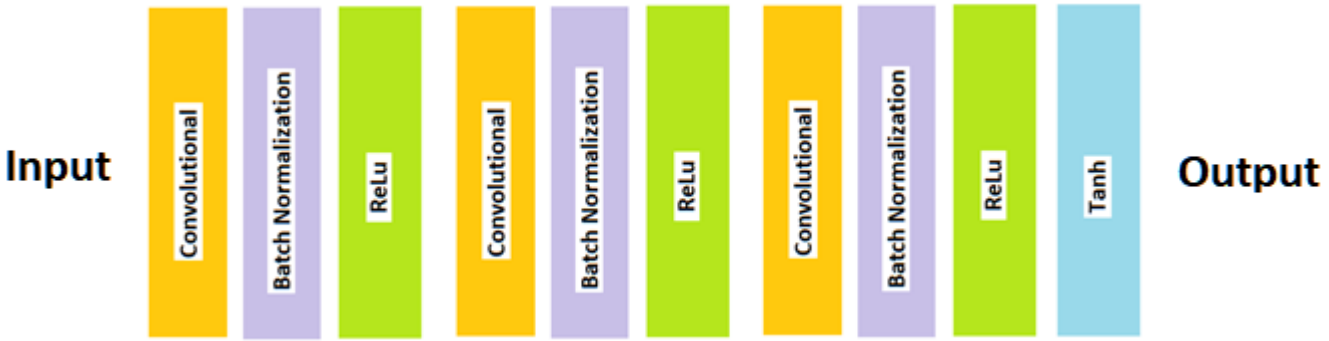


Figure 2. The architecture of the Generator

Proposed Method - GAN Architecture

The *Discriminator* contains five classical convolutional layers, the first four of them (in beige and orange colors) being each followed by a batch normalization layer and a LeakyRelu layer. The last convolutional layer is followed by a Sigmoid layer.

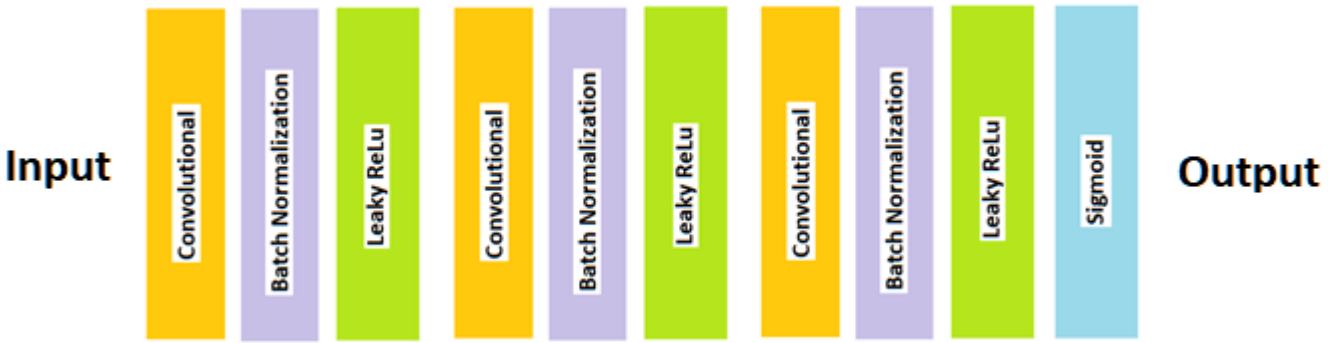


Figure 3. The architecture of the Discriminator

Proposed Method - GAN Training

During the GAN training, a typical GAN training process alternates between training of the *Discriminator* and training of the *Generator*. As batched of real and fake elements are sent to *Discriminator*, one of the *Discriminator* outputs is used to show if the element is closer to real (value 1) or closer to a fake (value 0). ADAM optimization has been used for both the *Discriminator* and *Generator*.

The *Discriminator* uses Binary Cross Entropy loss calculated as the sum of losses for the all real and all fake batches of pixels. The *Generator* also uses Binary Cross Entropy loss.

Proposed Method - DCNN Architecture

The DCNN architecture includes parallel flows: different types of convolutional layers and Gated Recurrent Units (GRU) layers. The top five flows are using convolutional layers while the bottom five flows are using GRU layers.

- Beige and Orange: convolutional layer, dropout layer, LeakyRelu activation layer, normalization layer
- Green and Cold Green: transpose convolutional layer, dropout layer, LeakyRelu activation layer, normalization layer
- Blue: linear layer, dropout layer, LeakyRelu activation layer, linear layer, dropout layer and the last LogSoftmax layer

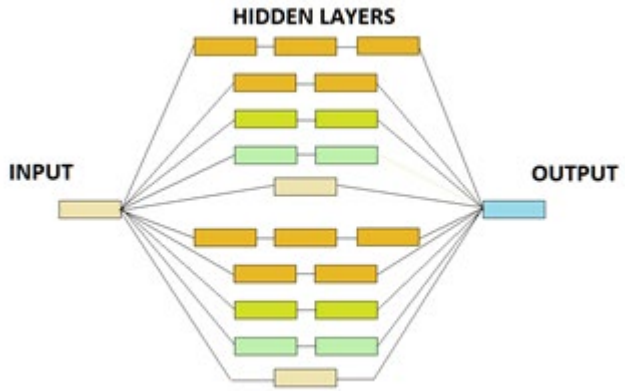


Figure 4. The architecture of the DCNN used for classification

Proposed Method - DCNN Architecture

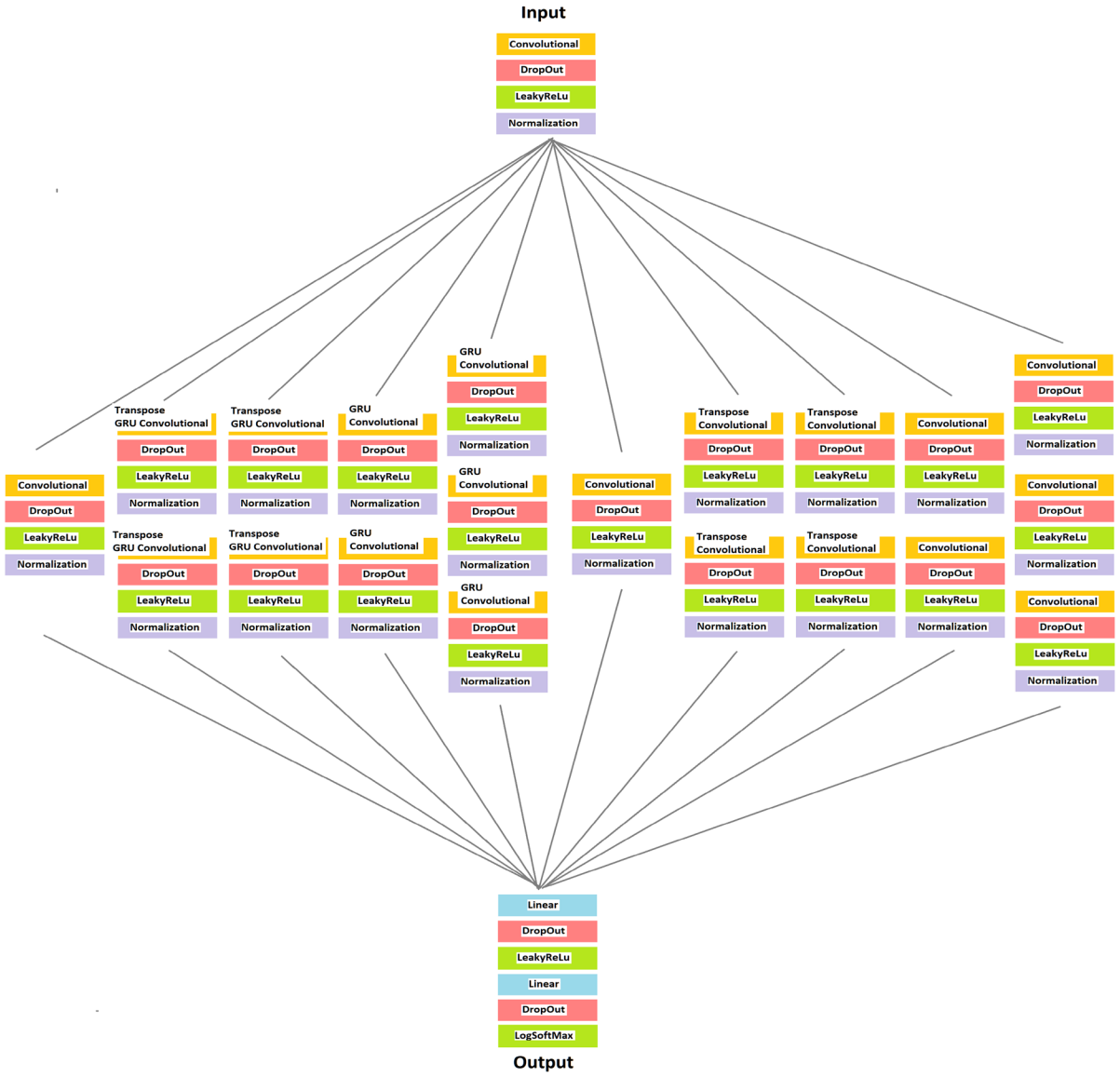


Figure 5. The detailed architecture of the DCNN used for classification. The DCNN includes parallel flows: different types of convolutional layers and Gated Recurrent Units (GRU) layers.

Proposed Method - DCNN Architecture

The presented architecture is the result of numerous tries and includes a novel combination of parallel flows.

The mix of convolutional layers, transpose convolutional layers and GRU layers have the scope of creating different internal models of data for more feature extracting opportunities.

Classical convolution support operations like activations and normalizations are used in an optimum order for increasing the training speed.

III. Experimental Results

Experimental Results – The Dataset

For the image classification described in this paper, we have used Pavia University dataset, containing an image of 610 x 340 pixels.

The dataset has 103 bands and 9 classes with the pixels distribution presented in the next slide.

Experimental Results – The Dataset

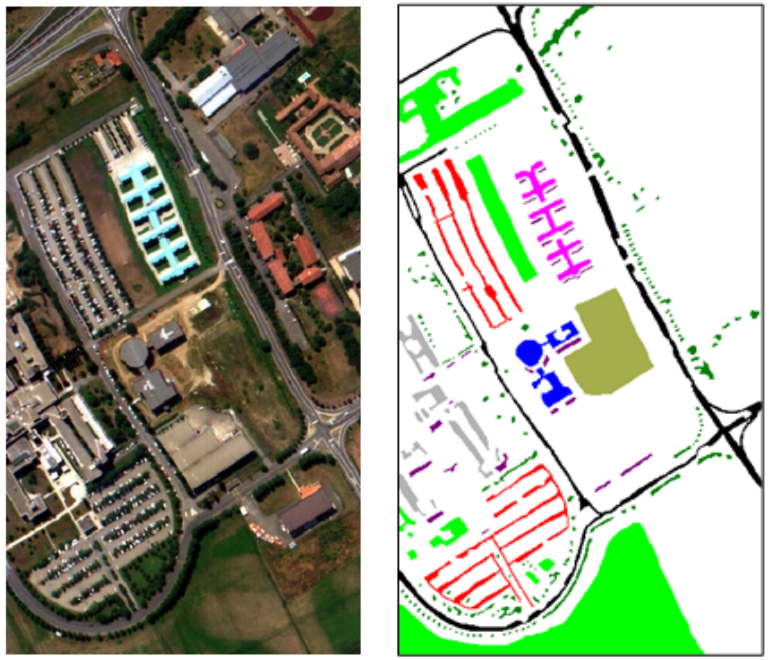


Figure 6. Pavia University map and hyperspectral ground truth image.

TABLE 1. NUMBER OF PIXELS ASSIGNED TO EACH CLASS

#	Class	Samples
1	Asphalt	6,631
2	Meadows	18,649
3	Gravel	2,099
4	Trees	3,064
5	Painted metal sheets	1,345
6	Bare Soil	5,029
7	Bitumen	1,330
8	Self-Blocking Bricks	3,682
9	Shadows	947

Experimental Results - Classifications

We have used the interband and the intraband correlation by representing the K-band pixel by the corresponding hyper rectangle centered in the current hyperspectral pixel.

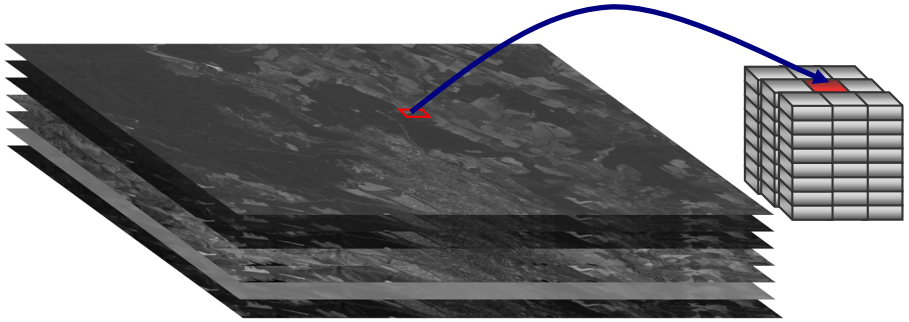


Figure 7.1 Spatial-contextual representation of pixels

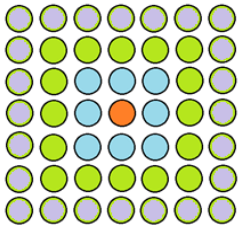


Figure 7.2 Example of pixel neighborhood (3 pixels in each direction)

Experimental Results – The Dataset

The following operations have been done with the original dataset file:

- Two lines of pixels, with all values being zero, have been added all around the image (to allow existence of neighbors to the pixels from the image margins)
- Each pixel has been collected taking into account its 24 neighbors
- The pixels have been split in training pixels, training validation pixels and test pixels.
- The new virtual pixels generated by the GAN system have been added to the training set of pixels.

Experimental Results - DCNN Hyperparameters

TABLE 2. DCNN HYPERPARAMETERS

Hyperparameter	Range
Optimization Algorithm	AdamW
Training Loss Type	CrossEntropyLoss
Learning Rate	cyclical learning rate, between 0.000001 and 0.0001
Number of convolutional filters	21
DropOut	0
Training Mini Batch Size	256
Number of epochs	99

Experimental Results - Cyclical Learning Rate

Regarding selection of the optimal learning rate for DCNN classifier, we had used a cyclical learning rate that went up and down inside an interval (1e-6 and 1e-3). When the intention has been to increase the learning rate, it was multiplied with 1.1 after each epoch and when the intention has been to decrease the learning rate, it was multiplied with 0.91. Initial value has been randomly chosen within the specified interval so the space of the learning rate tried values can be considered very large.

Experimental Results - Validation Loss / Early Stop

When the lowest value of training validation has been reached, the network has been saved and a counter has been started for several epochs previously defined. If during those epochs, no smaller training validation value has been obtained, the last saved network should be loaded and used for test.

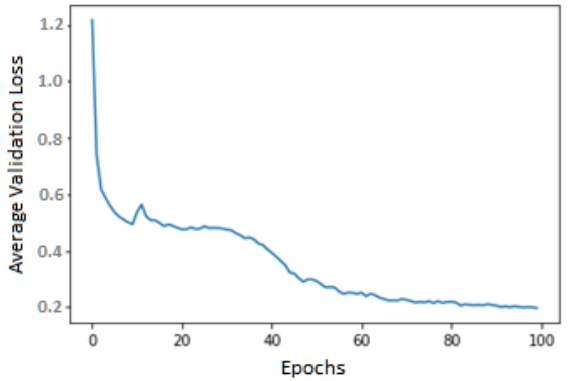


Figure 8. Average DCNN Training Validation Loss decrease

Experimental Results - Pixels split

A standard approach for hyperspectral pixel classifications implies splitting of the dataset in training set and test set. We have used six-pixel configurations for simulations (see Table 3):

TABLE 3. NUMBER OF PIXELS USED FOR EACH CLASS

Original Pixels for Training	Classical augmented pixels for training	GANs augmented pixels for training	Validation pixels	Test pixels
300	0	0	0	42,476
200	0	0	100	42,476
300	500	0	0	41,976
200	500	0	100	41,976
300	0	500	0	42,476
200	0	500	100	42,476

Experimental Results - GAN Pairs

For the pixel augmentation, the original training pixels have been split by class and original training set of each class has been used separately to train a *Generator* and a *Discriminator*.

As a result, for a dataset with 9 classes, we have used nine pairs of specialized *Generator – Discriminator* systems. For each class, the proposed GAN system has generated 500 pixels, and they are used for augmentation of the training dataset.

Experimental results – GAN Generated Elements

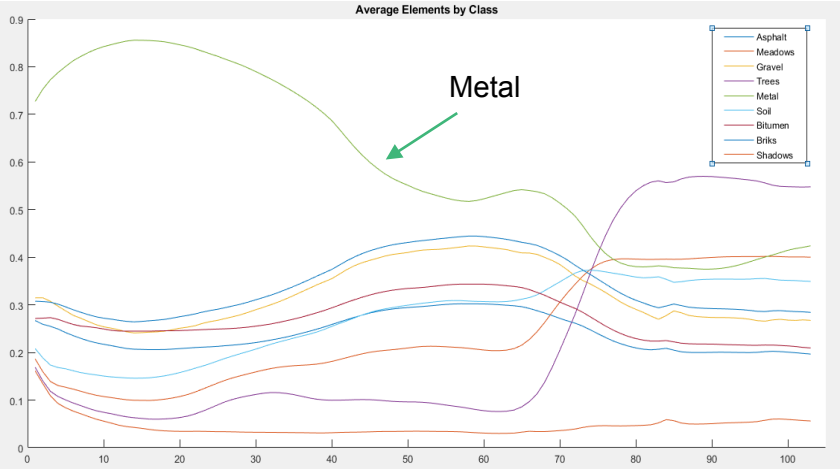


Figure 9. Average elements by Class

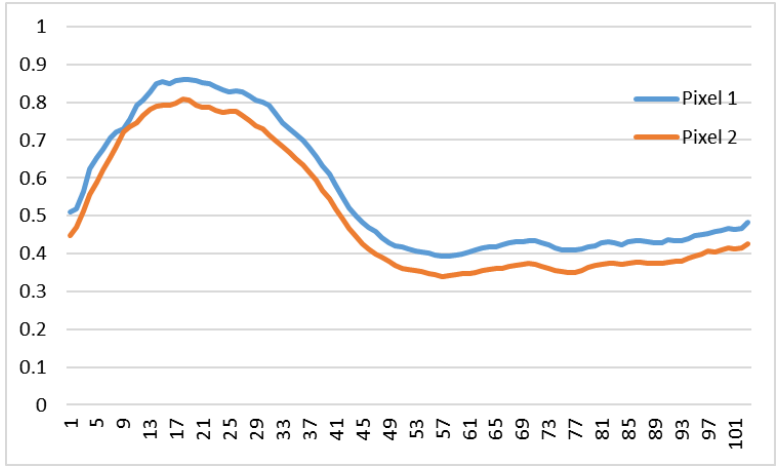


Figure 10. Two Examples of Original Metal Elements

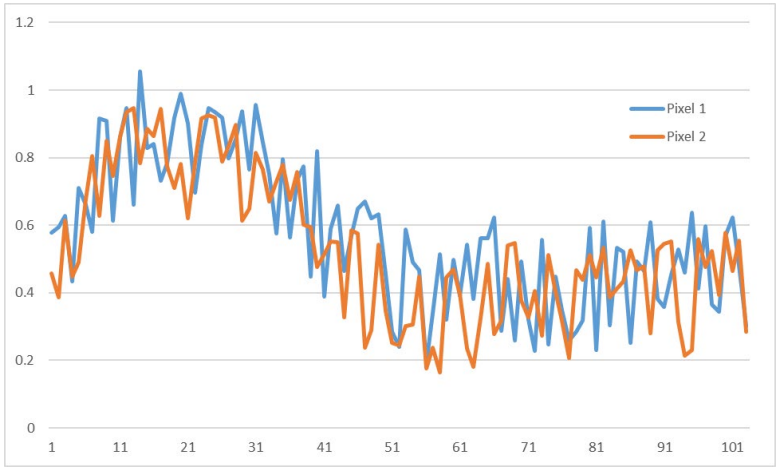


Figure 11. Two Examples of GAN Generated Metal Elements

Experimental results – GAN Generated Elements

Each GAN generation involves creating the entire parallelepiped of pixels, including all neighbor pixels around the meaningful pixel. This is happening because the GAN system is trained on classified training parallelepipeds.

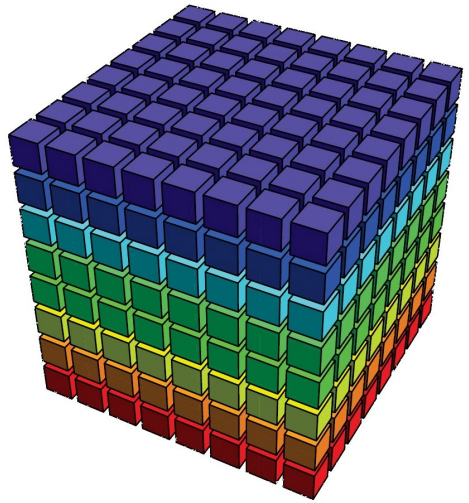


Figure 12.1. Original Element

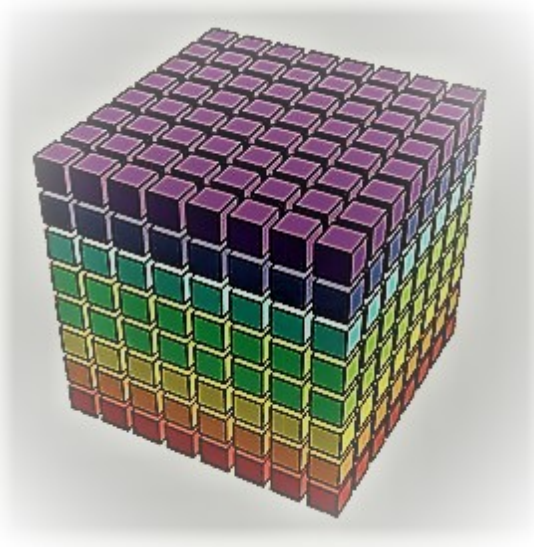


Figure 12.2. Generated Element

Experimental Results - GAN Convergence

For each of the 9 classes included by Pavia University dataset, the GAN system has been trained for a different number of epochs.

For example, the augmented pixels for class 4 have been generated in 21 epochs while the augmented pixels of class 8 have been generated in 150 epochs.

TABLE 4. NUMBER OF EPOCHS NEEDED TO REACH GAN CONVERGENCE

Class	Number of epochs to generate "good" virtual pixels
1	50
2	65
3	63
4	21
5	36
6	65
7	99
8	150
9	57

Experimental Results

TABLE 5. CLASSIFICATION RESULTS DEPENDING ON PIXEL ALLOCATION

Pixel Allocation for Augmentation and Validation	Classification Accuracy for Test Data Set
No Augmentation / No Validation	87.43%
No Augmentation / Validation	92.94%
Classical Augmentation / No Validation	84.14%
Classical Augmentation / Validation	86.76%
GAN Augmentation / No Validation	89.79%
GAN Augmentation / Validation	95.32%

Experimental Results

TABLE 6. CLASSIFICATION RESULTS PER CLASS

#	Class	Classification Accuracy for Test Data Set
1	Asphalt	91.62%
2	Meadows	96.72%
3	Gravel	93.45%
4	Trees	94.70%
5	Painted metal sheets	100%
6	Bare Soil	100%
7	Bitumen	100%
8	Self-Blocking Bricks	87.18%
9	Shadows	94.21%
	Average	95.32%

IV. Conclusions

Conclusions

The accuracy results regarding classification of the hyperspectral images prove the advantage of applying the proposed GAN augmentation for obtaining an improved training set.

These results have been obtained by using several of the latest deep learning techniques including Bayesian optimization of the hyperparameters, cyclical learning rates, reduced dropout value and the controlled termination of a simulation.

Conclusions

We have obtained a score of 95.32% correct classification using DCNN classifier with GANs versus the score of 92.94% of DCNN classifier without GANs on Pavia University hyperspectral image.

Other benefits from using GAN augmented training pixels can be a decrease in cost of acquiring labeled hyperspectral imaging and the relative high speed of training compared with other augmentation techniques.

References

References

1. T. Alipourfarda, H. Arefia, S. Mahmoudib, “A novel deep learning framework by combination of subspace-based feature extraction and convolutional neural networks for hyperspectral images classification”, Proc. Int. Geoscience Remote Sens. Symp. (IGARSS 2018), Valencia, Spain, pp.4784 – 4787, July 22-27, 2018.
2. P. Arun, K. Buddhiraju, A. Porwal, “Inversion of deep networks for modelling variations in spatial distributions of land cover classes across scales, Proc. Int. Geoscience Remote Sens. Symp. (IGARSS 2018), Valencia, Spain, pp.7133 – 7140, July 22-27, 2018.
3. J. Ball, P. Wei, “Deep learning hyperspectral image classification using multiple class–based denoising autoencoders, mixed pixel training augmentation, and morphological operations”, Proc. Int. Geoscience Remote Sens. Symp. (IGARSS 2018), Valencia, Spain, pp.6907 – 6910, July 22-27, 2018.
4. M. Bejiga, F. Melgani, “An adversarial approach to cross-sensor hyperspectral data classification”, Proc. Int. Geoscience Remote Sens. Symp. (IGARSS 2018), Valencia, Spain, pp. 3583-3586, July 22-27, 2018.
5. P. Diaconescu, V.E. Neagoe, “A highly configurable deep learning architecture for hyperspectral image classification”, Proc IEEE 13th International Symposium on Applied Computational Intelligence and Informatics (SACI 2019), Timisoara, Romania, pp. 197-200, in press.
6. S. Fang, D. Quan, S. Wang, L. Zhang, L. Zhou, “A two-branch network with semi-supervised learning for hyperspectral classification Proc. Int. Geoscience Remote Sens. Symp. (IGARSS 2018), Valencia, Spain, pp. 3868 – 3871, July 22-27, 2018.

References

7. M, Frid-Adar, I. Diamant, E. Klang, M. Amitai, J. Goldberger, H. Greenspan, “GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification", arXiv:1803.01229, 2018, unpublished.
8. Q. Gao, S. Lim, X. Jia, “Hyperspectral image classification using convolutional neural networks and multiple feature learning”, Remote Sens. 10, no. 2: 299.
9. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair et al. ,“Generative adversarial nets”, arXiv:1406.2661,
10. J. Li, L. Bruzzone, S. Liu, “Deep Feature Representation For Hyperspectral Image Classification”, Proc. Int. Geoscience Remote Sens.Symp. (IGARSS 2015), Milan, Italy, pp. 4951-4954, July 26-31, 2015.
11. K. Makantasis, K. Karantzalos, A. Doulamis, N. Doulamis, “Deep supervised learning for hyperspectral data classification through convolutional neural networks”, Proc. Int. Geoscience Remote Sens.Symp. (IGARSS 2015), Milan, Italy, pp.4959 – 4962, July 26-31, 2015.
12. V.E. Neagoe and A.D. Ciotec, "A new approach for accurate classification of hyperspectral images using virtual sample generation by concurrent self-organizing maps," Proc. Int. Geoscience Remote Sens. Symp. (IGARSS 2013), Melbourne, Australia, pp.1031-1034, July 21- 26, 2013.
13. V.E. Neagoe, A.D. Ciotec, L. Bruzzone, “A weakly-supervised change detection technique for SAR images based on deep learning and synthetic training data generated by an ensemble of self-organizing maps,” Proc. Int. Geoscience Remote Sens. Symp. (IGARSS 2019), Yokohamahyper-parameter, Japan, pp.1669-1672, July 28-August 2, 2019.

References

14. L. de Oliveira, M. Paganini, B. Nachman, “Learning particle physics by example: location-aware generative adversarial networks for physics synthesis”, arXiv:1701.05927
15. A. Radford, L. Metz, S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks”, arXiv:1511.06434v2 [cs.LG]
16. T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Redford, X. Chen, “Improved techniques for training GANs”, arXiv:1606.03498v1 [cs.LG]
17. L. Smith, “Cyclical learning rates for training neural networks”, arXiv:1506.01186v6 [cs.CV]
18. W. Song, L. Fang, S. Li, “Similarity-preserving deep features for hyperspectral image classification”, Proc. Int. Geoscience Remote Sens. Symp. (IGARSS 2018), Valencia, Spain, pp. 3603–3606, July 22-27, 2018.
19. Z. Tian, J. Ji, S. Mei, J. Hou, S. Wan, Q. Du, “Hyperspectral classification via spatial context exploration with multi-scale CNN”, Proc. Int. Geoscience Remote Sens. Symp. (IGARSS 2018), Valencia, Spain, pp. 2571 – 2574, July 22-27, 2018.
20. Z. Xiong, Y. Yuan, Q. Wang, AI-NET: “Attention inception neural networks for hyperspectral image classification, Proc. Int. Geoscience Remote Sens. Symp. (IGARSS 2018), Valencia, Spain, pp. 2655 – 2658, July 22-27, 2018.
21. Y. Xu, B. Du, L. Zhang, “Can we generate good samples for hyperspectral classification? --A generative adversarial network based method”, Proc. Int. Geoscience Remote Sens. Symp. (IGARSS 2018), Valencia, Spain, pp.5756 – 5759, July 22-27, 2028.
22. S. Zhao, W. Li, Q. Du, Q. Ran, “Hyperspectral classification based on siamese neural network using spectral-spatial feature”, Proc. Int. Geoscience Remote Sens. Symp. (IGARSS 2018), Valencia, Spain pp.2575 – 2582, July 22-27, 2018.

Thank you!