Journal of Integrated and Advanced Engineering (JIAE)
Vol. 2, No. 2, September 2022: 107-112
http://asasijournal.id/index.php/jiae
http://doi.org/10.51662/jiae.v2i2.80

Generating images for Supervised Hyperspectral Image Classification with Generative Adversarial Nets

Hassan Abdalla Abdelkarim Osman, Norsinnira Zainul Azlan*
Department of Mechatronics, International Islamic University Malaysia, Malaysia

Abstract
With the advancement of remote sensing technologies, hyperspectral imagery has garnered significant interest in the remote sensing community. These developments have inspired improvement in various hyperspectral images (HSI) classification applications, such as land cover mapping, amongst other earth observation applications. Deep Neural Networks have revolutionized image classification tasks in areas of computer vision. However, in the domain of hyperspectral images, insufficient training samples have been earmarked as a significant bottleneck for supervised HSI classification. Moreover, acquiring HSI from satellites and other remote sensors is expensive. Thus, researchers have turned to generative models to leverage the existing data to increase training samples, such as particularly generative adversarial networks (GAN). This paper explores the use of a vanilla GAN to generate synthetic data. The network employed in this paper was built using a deep learning python package, PyTorch and tested on a popular HSI dataset called Indian Pines dataset. The network achieved an overall accuracy of 64%. While promising, there is still room for improvement.

Keywords:
Deep learning;
Generative Adversarial Networks;
Hyperspectral Imagery;
Remote Sensing;

Article History:
Received: August 22, 2022
Revised: September 16, 2022
Accepted: September 20, 2022
Published: September 24, 2022

Corresponding Author:
Norsinnira Zainul Azlan,
Department of Mechatronics,
International Islamic University,
Malaysia.
Email: sinnira@iium.edu.my

INTRODUCTION

Hyperspectral Imagery uses sensors to remotely capture an object's spectral and spatial properties [1]. A hyperspectral image (HSI) contains reflected radiance in hundreds of narrow or wide bands on the electromagnetic spectrum [2]. In an RGB image, a pixel has information in three channels, namely red, green and blue. However, in an HSI, a pixel includes spectral data in several bands being recorded, usually in the hundreds. This puts into perspective the rich information present in an HSI. The ability to capture rich data with HSI's garnered interest in many fields for classification tasks. For instance, in [3], drone-based hyperspectral imagery was used to remotely detect algal pigmentation at different water depths. Meanwhile, other works use satellite-based hyperspectral imagery for land cover classification and change detection [4][5].

The use of HSI's can also be seen in widespread real-world applications ranging from the medical field [6] to industries such as food [7][8] and agriculture [9][10]. Many supervised learning models have been explored in the space of hyperspectral image classification. Before the recent success of Deep Neural Network (DNN) seen in areas of computer vision such as image classification [11], traditional supervised learning algorithms such as K Nearest Neighbor (KNN) and maximum likelihood were used [1].

However, due to the nature of HSI being high in dimensionality, these algorithms weren't able to perform on them effectively. Support vector machines (SVMs) can handle high dimensionality and are shown to be good spectral classifiers [1]. However, SVM's fall short when handling spatial and spectral features in the data [12][13]. Deep learning models, particularly Convolutional Neural Networks (CNN), are better suited for extracting spatial-spectral features. This is the case for several reasons, for instance, their ability to learn latent features in a hierarchical manner comprising customizable layers. Each layer progressively retains more abstract features. For example, in [14], lower layers extract low-level features such as edges while higher layers learn higher-level features such as faces. In [13], the properties of CNNs were exploited for HSI classification, and their work showed promising results. Several other deep learning networks have also been proposed for HSI classification, such as Autoencoders (AE) [15] and Recurrent Neural Networks (RNN)[16].

DL models are known to be "data hungry"; they require a large amount of data. On the other hand, there is a limited number of training samples in HSI classification due to the cost and intensive labor required to gather and label data [12]. Although there has been progress in using the DL model for HSI classification, the lack of training data makes DL models prone to overfitting, that is, performing well on the training dataset but showing suboptimal performance in the test dataset [11]. Hence, it is imperative to devise ways to largen existing datasets as it is critical to the performance of DL models. Data augmentation is a popular method used in various domains of computer vision to improve a model's robustness to overfitting. For example, simple geometric operations such as rotating and shifting the image proved an effective strategy [17][18]. Another interesting approach to the problem becoming increasingly popular is using Generative Adversarial Networks (GAN) to generate synthetic data. GAN's are powerful in learning the distribution in a dataset and outputting desired samples [19].

This paper explores the use of GAN's on a popular dataset, i.e., Indian Pines, in the domain of hyperspectral image classification. The paper's contribution is to design a GAN model that generates real samples to increase training datasets. The rest of this paper is structured as follows. Section 2 briefly highlights the main theory behind GANs. Section 3 breakdowns the training method used, followed by the experimental setup and results and discussion in Section 4 and Section 5, respectively. Finally, Section 6 draws the conclusion and future perspectives.

MATERIAL AND METHOD

Generative Adversarial Networks Theory

GANs are a class of generative models comprising two neural networks: discriminator (D) and generator (G). The role of the generator network is to generate samples, G(z), by estimating the probability distribution in a dataset. These generated samples are then fed as input to the discriminator one input, while the other input is real samples from the dataset (x). The role of the discriminator network is to distinguish between fake and real images, illustrated in Figure 1.

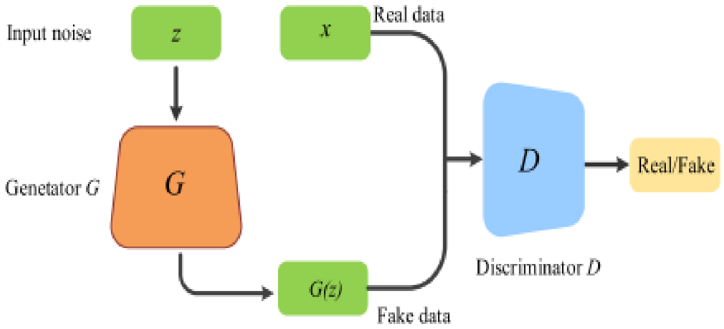

Figure 1. The architecture of GANs [1]

Essentially, the generator's objective is to fool the discriminator into thinking that the samples generated are real. These networks are trained in an adversarial manner, hence the term Generative Adversarial Networks. This groundbreaking model, introduced by [20][21], showed great potential in generating desired samples and was applied to various computer vision applications [19].

Training Generative Adversarial Networks

Training GAN's involves training two separate networks simultaneously, thus making it a more difficult task than training other machine learning models. To successfully train a GAN, both networks must be optimized to perform their respective task, i.e., the generator is optimized to fool the discriminator. In contrast, the discriminator is optimized to call the generator's bluff. The initial step in the training process is to feed the generator with a randomized input noise (z), as shown in Figure 1. This input noise is a fixed-length vector drawn from a gaussian distribution. The vector size corresponds to the size of real samples in the dataset. The generator produces a fake sample, G (z, θ_g), where θg are the network parameters. The fake sample, alongside a real sample, is fed to the discriminator network D (x, θd), where θd are the network parameters. The discriminator outputs a single scalar, D (x), which denotes the probability that x comes from the data rather than the generator G. Up until this point is the feed-forward part of the training process.

GAN's fall under the category of unsupervised learning. However, an attractive property of training them is that they are trained in a supervised manner. The next step after forward propagating the inputs (i.e., feed-forward) is backpropagation. Backpropagation is a method of calculating the gradient of the neural network to optimize a loss function. A loss function is a function that determines how well a network models the dataset. In the case of GAN's, backpropagation isn't straightforward as there are two networks to train simultaneously. Thus, there are two loss functions to optimize in one iteration. The discriminator D is trained to maximize the probability of assigning a correct label to both actual training data and fake examples generated by the generator G. The loss function is the binary cross entropy. Simultaneously, G is trained to minimize log (1 − D (G (z))). GAN's have a minimax objective function, that is,

$$\max_{(G)}\min_{(D)} V(D,G) = Ex{\sim}pdata(x)\,[\log D(x)] + Ez{\sim}pz(z)\,[\log(1 - D(G(z)))] \qquad (1)$$

where log D (x) is the cross-entropy between $[1\ 0]^T$ and $[D(x)\ 1 - D(x)]^T$, meanwhile log (1 − D (G (z))) is the cross-entropy between $[0\ 1]^T$ and $[D(G(z))\ 1 - D(G(z))]^T$.

The following are the steps to train GAN's:

1. Initialize the random vector, z, by sampling from a Gaussian distribution.
2. Input z into the generator. Generator outputs G (z, θ_g).
3. Input G (z, θ_g) and real sample into discriminator outputs D (x, θ_d).
4. Backpropagate using loss function for discriminator, only optimizing the discriminator.
5. Freeze layers of the discriminator, and backpropagate using the loss function of the generator to optimize the generator.
6. Repeat steps 2 to 6 until convergence.

Experimental Setup and Dataset

The dataset used in this experiment is the Indian Pines dataset. It contains 145x145 pixels and 224 spectral bands. In this paper, 200 spectral bands are adopted for analysis. The Indian Pines dataset contains 16 vegetation classes. The false-color image (bands 50, 27, 17) and its ground truth are shown in Figure 2a and Figure 2b, respectively. Figure 3 shows the number of classes along with the number of labelled samples.

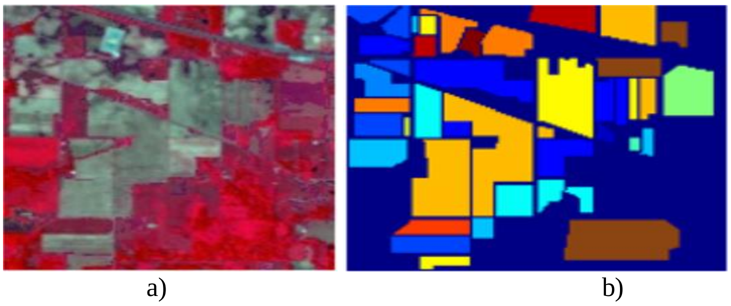

Figure 2. a) False color composite image of the Indian Pines Dataset [22] b) Ground Truth of Indian Pines dataset [22]

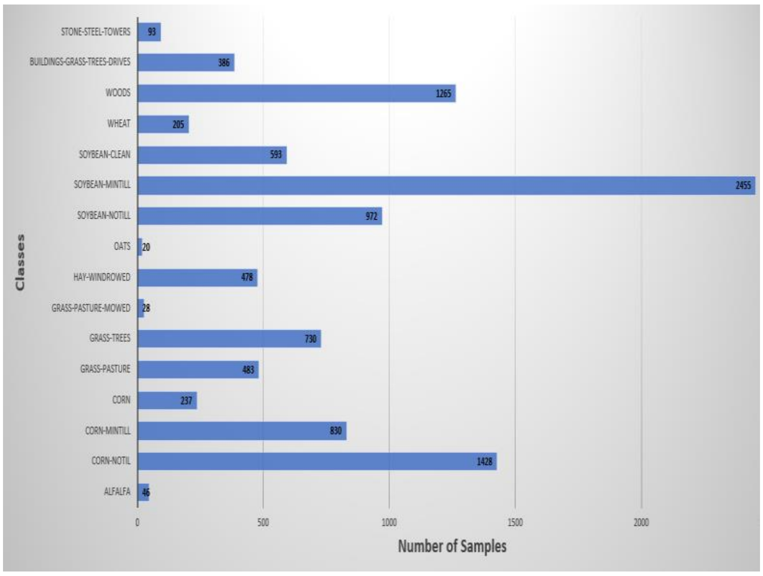

Figure 3. 16 classes and the number of samples per class

GAN Model

The network design modifies the 3D-GAN used in [1]. Notably, two main differences exist in the approach taken between the two designs. Firstly, in [1], PCA analysis was used to reduce the dimensions of the data. This allows for a lack of feature extraction in the spectral band. We deviate from using PCA and thus use a deeper layer to construct the generator and discriminator.

The model was built using PyTorch, a popular python deep learning framework and is implemented on google collab. At the time of writing, google collab offers a free GPU, namely Tesla T4, with 16 GPU of RAM.

RESULTS AND DISCUSSION

The classification results of the Indian Pines dataset compared to previous work are shown in Table 1. The three metrics used to evaluate the performance of the network are Overall Accuracy (OA), Average Accuracy (AA) and Kappa coefficient (Kappa). The overall accuracy is the ratio of the total samples correctly classified. At the same time, the kappa coefficient pertains to the number of samples correctly classified and the number of samples incorrectly classified. As can be seen, the best-performing algorithm reported is CA-GAN [23]. Perhaps the use of an attention mechanism and long-short term memory (LSTM) layer, such as in [24], can be used to retain spectral-spatial features. However, our model outperforms HSGAN on all metrics while outperforming 3D-GAN on average accuracy.

Another takeaway from the results is the use and effectiveness of Principal Component Analysis (PCA). In the benchmark algorithms, PCA was used to reduce the dimensionality of the dataset. However, in this research, PCA was not used. In order to compare the effectiveness of PCA, we compare the time used to train the model and benchmark it against the other models. This is shown in Table 2. As can be seen, all the other models take significantly less time to train than our GAN model. This is due to the high dimensionality present without using PCA. On the other hand, CA-GAN takes longer than the rest of the models. According to [24], this could be attributed to the number of parameters in the attention and convLSTM layers [25][26].

Table 1. Performance of GAN on Indian Pines dataset compared to another model [24]

	Model	**Overall Accuracy (OA)** %	**Average Accuracy (AA)** %	**Kappa** %
1	***GAN****	**92 ± 0.6**	**86 ± 1.8**	**88 ± 0.5**
2	HSGAN [26]	74.0±0.9	60.2±2.6	70.0±1.0
3	CA-GAN [24]	97.4±0.5	95.2±2.2	97.0±0.6
4	3D-GAN [1]	93.5±0.3	84.8±2.7	93.1±1.2

Table 2. Runtime of models. Results of other models adapted from [24]

	Model	**Training time** **(s)**
1	***GAN****	**3456 ± 0.2**
2	HSGAN [26]	444.7 ± 73.1
3	CA-GAN [24]	712.9 ± 3.1
4	3D-GAN [1]	597.67 ± 60.8

CONCLUSION

HSI image classification continues to draw interest in remote sensing due to its breakthrough in recent years. Insufficient labelled data has been one of the main bottlenecks in supervised learning using state-of-the-art deep learning methods. This paper used a vanilla GAN to generate new training samples and discussed the approach's components. It has shown promising results that proved that GAN's could be used to create synthesized data and thus help with limited training samples. However, there is still room for improvement. Future studies can be conducted to improve the generalization capabilities of GAN's to learn the deeper latent structure and truly exploit the capabilities of supervised learning deep learning models.

REFERENCES

[1] L. Zhu, Y. Chen, P. Ghamisi and J. A. Benediktsson, "Generative Adversarial Networks for Hyperspectral Image Classification," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 9, pp. 5046-5063, Sept. 2018, doi: 10.1109/TGRS.2018.2805286

[2] A. Ozdemir & K. Polat, "Deep learning applications for hyperspectral imaging: a systematic review,“ *Journal of the Institute of Electronics and Computer*, vol. 2, no., pp. 39-56, 2020, doi: 10.33969/JIEC.2020.21004

[3] Y. S. Kwon, J C. Pyo, Y-H. Kwon, H., Duan, K. H. Cho & Y. Park, "Drone-based hyperspectral remote sensing of cyanobacteria using vertical cumulative pigment concentration in a deep reservoir,“ *Remote Sensing of Environment*, vol. 236, ID: 111517, 2020, doi: 10.1016/j.rse.2019.111517

[4] M. Crowson, R. Hagensieker & B. Waske,"Mapping land cover change in northern Brazil with limited training data,“ *International Journal of Applied Earth Observation and Geoinformation*, vol. 78, pp. 202-214, 2019, doi: 10.1016/j.jag.2018.10.004

[5] G. Rousset, M. Despinoy, K. Schindler & M. Mangeas,"Assessment of deep learning techniques for land use land cover classification in southern New Caledonia,“ *Remote Sensing*, vol. 13, no. 12, ID: 2257, 2021, doi: 10.3390/rs13122257

[6] U. Khan, S. Paheding, C. P. Elkin and V. K. Devabhaktuni, "Trends in Deep Learning for Medical Hyperspectral Image Analysis," in *IEEE Access*, vol. 9, pp. 79534-79548, 2021

[7] R. Calvini, S. Michelini, V. Pizzamiglio, G. Foca & A. Ulrici, “Exploring the potential of NIR hyperspectral imaging for automated quantification of rind amount in grated Parmigiano Reggiano cheese,“ *Food Control*, vol. 112, ID: 107111, 2020, doi: 10.1016/j.foodcont.2020.107111

[8] L. Zhang, Z. Rao & H. Ji, “Hyperspectral imaging technology combined with multivariate data analysis to identify heat-damaged rice seeds,“ *Spectroscopy Letters*, vol.53, no. 3, pp. 207-221, 2020, doi: 10.1080/00387010.2020.1726402

[9] P. V. Manley, V. Sagan, F. B. Fritschi & J. G. Burken, “Remote sensing of explosives-induced stress in plants: Hyperspectral imaging analysis for remote detection of unexploded threats,“ *Remote Sensing*, vol. 11, no. 15, ID: 1827, 2019, doi: 10.3390/rs11151827

[10] J. Peón et al.,“Prediction of topsoil organic carbon using airborne and satellite hyperspectral imagery,“ *Remote Sensing*, vol. 9, no. 12, ID: 1211, 2017, doi: 10.3390/rs9121211

[11] D. L. Z. Astuti, S. Samsuryadi, & D. P. Rini,“ Real-time classification of facial expressions using a principal component analysis and convolutional neural network,“ *SINERGI*, vol. 23, no. 3, pp. 239-244, 2019, doi: 10.22441/sinergi.2019.3.008

[12] N. Wambugu et al., “Hyperspectral image classification on insufficient-sample and feature learning using deep neural networks: A review,“ *International Journal of Applied Earth Observation and Geoinformation*, vol. 105, ID: 102603, 2021, doi: 10.1016/j.jag.2021.102603

[13] Z. Zhong, J. Li, Z. Luo and M. Chapman, "Spectral–Spatial Residual Network for Hyperspectral Image Classification: A 3-D Deep Learning Framework," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 2, pp. 847-858, Feb. 2018, doi: 10.1109/TGRS.2017.2755542

[14] D. Salama, A. M. Almansori, M., Taha & E-S. Badr, “A deep facial recognition system using computational intelligent algorithms,“ *PLoS ONE*, vol. 15, no. 12, ID; e0242269, 2020, doi: 10.1371/journal.pone.0242269

[15] H. Xu, W. He, L. Zhang and H. Zhang, "Unsupervised Spectral–Spatial Semantic Feature Learning for Hyperspectral Image Classification," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1-14, 2022, Art no. 5526714, doi: 10.1109/TGRS.2022.3159789

[16] L. Mou, P. Ghamisi and X. X. Zhu, "Deep Recurrent Neural Networks for Hyperspectral Image Classification," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 7, pp. 3639-3655, July 2017, doi: 10.1109/TGRS.2016.2636241

[17] W. Wang, X. Liu & X Mou, “Data Augmentation and Spectral Structure Features for Limited Samples Hyperspectral Classification,“ *Remote Sensing*, vol. 13, no. 4, pp. 547, 2021, doi: 10.3390/rs13040547

[18] M. Ahmad, S. Shabbir, S. K. Roy, D. Hong, X. Wu, and J. Yao, “Hyperspectral Image Classification-Traditional to Deep Models: A Survey for Future Prospects,“ *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, pp. 1-20, 2021

[19] Z. Wang, Q. She & T. E. Ward, “Generative adversarial networks in computer vision: A survey and taxonomy,“ ACM Computing Surveys (CSUR), vol. 54, no. 2, pp. 1-38, 2021, doi: 10.1145/3439723

[20] I. Goodfellow et al., “Generative adversarial networks,“ *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020, doi: 10.1145/3422622

[21] T. Alipour-Fard and H. Arefi, "Structure Aware Generative Adversarial Networks for Hyperspectral Image Classification," in *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 5424-5438, 2020, doi: 10.1109/JSTARS.2020.3022781

[22] M. E. Paoletti, J. M. Haut, J. Plaza & A. Plaza, “Deep learning classifiers for hyperspectral imaging: A review,“ *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 158, pp. 279-317, 2019, doi: 10.1016/j.isprsjprs.2019.09.006

[23] W. Lv & X. Wang, “Overview of hyperspectral image classification,“ *Journal of Sensors*, vol. 2, pp. 1-13, 2020, doi: 10.1155/2020/4817234

[24] J. Feng, X. Feng, J. Chen, X. Cao, X., Zhang, L, Jiao & T. Yu, “Generative adversarial networks based on collaborative learning and attention mechanism for hyperspectral image classification,“ *Remote Sensing*, vol. 12, no. 7, ID: 1149, 2020, doi: 10.3390/rs12071149

[25] W. Li, G. Wu, F. Zhang and Q. Du, "Hyperspectral Image Classification Using Deep Pixel-Pair Features," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 2, pp. 844-853, Feb. 2017, doi: 10.1109/TGRS.2016.2616355

[26] Y. Zhan, D. Hu, Y. Wang and X. Yu, "Semisupervised Hyperspectral Image Classification Based on Generative Adversarial Networks," in *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 2, pp. 212-216, Feb. 2018, doi: 10.1109/LGRS.2017.2780890.