MARTA GANs: Unsupervised Representation Learning for Remote Sensing Image Classification

Daoyu Lin, Kun Fu, Yang Wang, Guangluan Xu, and Xian Sun

***Abstract*— With the development of deep learning, supervised learning has frequently been adopted to classify remotely sensed images using convolutional networks. However, due to the limited amount of labeled data available, supervised learning is often difficult to carry out. Therefore, we proposed an unsupervised model called multiple-layer feature-matching generative adversarial networks (MARTA GANs) to learn a representation using only unlabeled data. MARTA GANs consists of both a generative model *G* and a discriminative model *D*. We treat *D* as a feature extractor. To fit the complex properties of remote sensing data, we use a fusion layer to merge the mid-level and global features. *G* can produce numerous images that are similar to the training data; therefore, *D* can learn better representations of remotely sensed images using the training data provided by *G*. The classification results on two widely used remote sensing image databases show that the proposed method significantly improves the classification performance compared with other state-of-the-art methods.**

***Index Terms*— Generative adversarial networks (GANs), scene classification, unsupervised representation learning.**

I. Introduction

As SATELLITE imaging techniques improve, an ever-growing number of high-resolution satellite images provided by special satellite sensors have become available. It is urgent to be able to interpret these massive image repositories in automatic and accurate ways. In recent decades, scene classification has become a hot topic and is now a fundamental method for land-resource management and urban planning applications. Compared with other images, remote sensing images have several special features. For example, even in the same category, the objects we are interested in usually have different sizes, colors, and angles. Moreover, other materials around the target area cause high intraclass variance and low interclass variance. Therefore, learning robust and discriminative representations from remotely sensed images is difficult.

Previously, the bag of visual words (BoVW) [1] method was frequently adopted for remote sensing scene classification. BoVW includes the following three steps: feature detection, feature description, and codebook generation. To overcome the problems of the orderless bag of features image representation, the spatial pyramid matching model [2] was proposed, which works by partitioning the image into increasingly fine subregions and computing histograms of local features found inside each subregion. The above-mentioned methods have comprised the state of the art for several years in the remote sensing community [3], but they are based on handcrafted features, which are difficult, are time-consuming, and require domain expertise to produce.

Deep learning algorithms can learn high-level semantic features automatically rather than requiring handcrafted features. Some approaches [4], [5] based on convolutional neural networks (CNNs) [6] have achieved success in remote sensing scene classification, but those methods usually require an enormous amount of labeled training data or are fine-tuned from pretrained CNNs.

Several unsupervised representation learning algorithms have been based on the autoencoder [7], [8], which receives corrupted data as input and is trained to predict the original, uncorrupted input. Although training the autoencoder requires only unlabeled data, input reconstruction may not be the ideal metric for learning a general-purpose representation. The concept of generative adversarial networks (GANs) [9] is one of the most exciting unsupervised algorithm ideas to appear in recent years; its purpose is to learn a generative distribution of data through a two-player minimax game. In subsequent work, a deep convolutional GAN (DCGAN) [10] achieved a high level of performance on image synthesis tasks, showing that its latent representation space captures important variation factors.

GANs is a promising unsupervised learning method, yet thus far, it has rarely been applied in the remote sensing field. Due to the tremendous volume of remote sensing images, it would be prohibitively time-consuming and expensive to label all the data. To tackle this issue, GANs would be the excellent choice, because it is an unsupervised learning method in which the required quantities of training data would be provided by its generator. Therefore, in this letter, we propose a multiple-layer feature-matching GAN (MARTA GAN) model to learn the representation of remote sensing images using unlabeled data.

Although based on DCGAN, our approach is rather different in the following aspects: 1) DCGAN can, at most, produce

Manuscript received March 19, 2017; revised June 26, 2017 and August 24, 2017; accepted September 12, 2017. Date of publication October 5, 2017; date of current version October 25, 2017. This work was supported by the National Natural Science Foundation of China under Grant 41501485 and Grant 61331017. *(Corresponding author: Kun Fu.)*

The authors are with the Key Laboratory of Technology in Geospatial Information Processing and Application System, Institute of Electronics, Chinese Academy of Sciences, Beijing 100190, China (e-mail: lindaoyu15@mails.ucas.ac.cn; fukun@mail.ie.ac.cn; primular@163.com; gluanxu@mail.ie.ac.cn; sunxian@mail.ie.ac.cn).

Color versions of one or more of the figures in this letter are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LGRS.2017.2752750

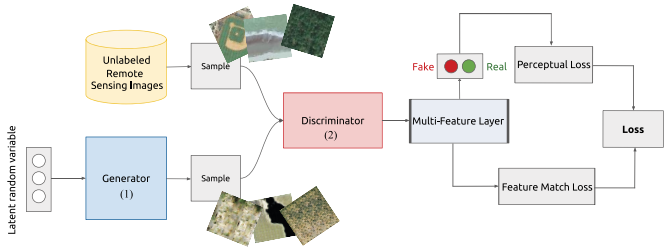


Fig. 1. Overview of the proposed approach. The discriminator (2) learns to make classifications between real and synthesized images, while the generator (1) learns to fool the discriminator.

images with a 64×64 resolution, while our approach can produce remote sensing images with a resolution of 256×256 by adding two deconvolutional layers in the generator; 2) to avoid the problem of such deconvolutional layers producing checkerboard artifacts, the kernel sizes of our networks are 4×4, while those of DCGAN are 5×5; 3) we propose a multifeature layer to aggregate the mid- and high-level information; and 4) we combine both the perceptual loss and feature-matching loss to produce more accurate fake images. Based on the improvements above, our method can realize the better representation of remote sensing images among all methods. Fig. 1 shows the overall model.

The contributions of this letter are the following.

1) To the best of our knowledge, this is the first time that GANs have been applied to classify unsupervised remote sensing images.
2) The results of experiments on the UC-Merced Land-use and Brazilian Coffee Scenes data sets showed that the proposed algorithm outperforms the state-of-the-art unsupervised algorithms in terms of overall classification accuracy.
3) We propose a multifeature layer by combining perceptual loss and loss of feature matching to learn better image representations.

II. Method

A GAN is most straightforward to apply when the involved models are both multilayer perceptrons; however, to apply a GAN to remote sensing images, we used CNNs for both the generator and the discriminator in this letter. The generator network directly produces samples $x = G(z; \theta_g)$ with parameters θ_g and z, where z obeys a prior noise distribution $p_z(z)$. Its adversary, the discriminator network, attempts to distinguish between samples drawn from the training data and samples created by the generator. The discriminator emits a probability value denoted by $D(x; \theta_d)$ with parameters θ_d, indicating the probability that x is a real training example rather than a fake sample drawn from the generator. During the classification task, the discriminative model D is regarded as the feature extractor. Then, additional training data so that the discriminator can learn a better representation is provided by the generative model G.

A. Training the Discriminator

When training the discriminator, the weights of the generator are fixed. The goals of training the discriminator $D(x)$ are as follows.

1) Maximize $D(x)$ for every image from the real training examples.
2) Minimize $D(x)$ for every image from the fake samples drawn from the generator.

Therefore, the objective function of training discriminator is to maximize

$$\mathbb{E}_{x\sim p_{\text{data}}(x)} \log D(x) + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))]. \quad (1)$$

B. Training the Generator

When training the generator, the weights of the discriminator are fixed. The goal of training the generator $G(z)$ is to produce samples that fool D. The output of the generator is an image that can be used as the input for the discriminator. Therefore, the generator wants to maximize $D(G(z))$ [or equivalently, minimize $1 - D(G(z))$], because D is a probability estimate that ranges only between 0 and 1. We call this concept perceptual loss; it encourages the reconstructed image to be similar to the samples drawn from the training set by minimizing the perceptual loss

$$\ell_{\text{perceptual}} = \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))]. \quad (2)$$

In summary, the discriminator D is shown an image produced from the generator G and adjusts its parameters to make its output, $D(G(Z))$, larger. But $G(Z)$ will train itself to produce images that fool D into thinking they are real. It does this by getting the gradient of D with respect to each sample it produces. In other words, the G is trying to minimize the output, while D is trying to maximize it; consequently, it is a minimax game that is defined as follows:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x\sim p_{\text{data}}(x)} \log D(x) + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))]. \quad (3)$$

To make the images generated by generator more similar to the real images, we train the generator to match the expected values of the features in the multifeature layer of the discriminator. Letting $f(x)$ denote activations on the multifeature layer of the discriminator, the loss of feature matching for the generator is defined as follows:

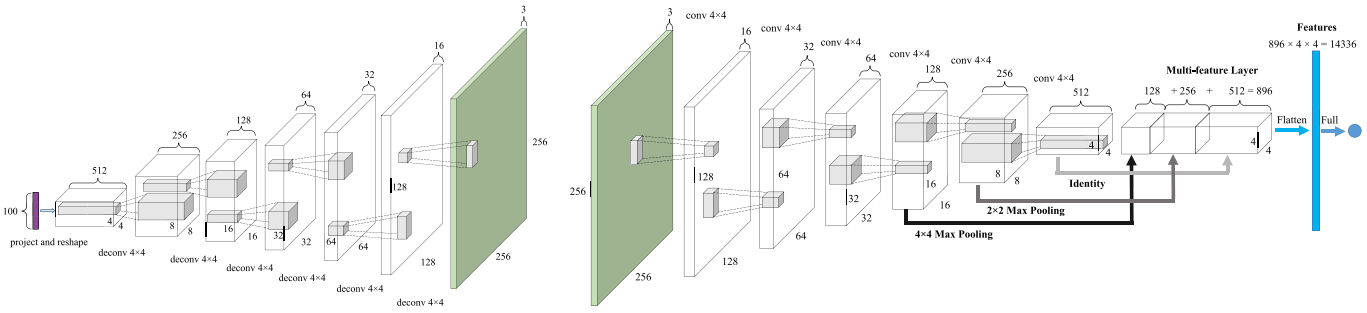
$$\ell_{\text{feature\_match}} = ||\mathbb{E}_{x\sim p_{\text{data}}(x)} f(x) + \mathbb{E}_{z\sim p_z(z)} f(G(z))||_2^2. \quad (4)$$

Therefore, our final object [the combination of (2) and (4)] for training the generator is to minimize

$$\ell_{\text{final}} = \ell_{\text{perceptual}} + \ell_{\text{feature\_matching}}. \quad (5)$$

C. Network Architectures

The details of the generator and the discriminator in MARTA GANs are as follows. The generator takes 100 random numbers drawn from a uniform distribution as input. Then, the result is reshaped into a 4-D tensor. We used six deconvolutional layers in our generator to learn its own spatial upsampling and upsample the 4×4 feature maps to 256×256 remote sensing images. Fig. 2(a) shows a visualization of the generator.



(a) (b)

Fig. 2. Network architectures of a generator and a discriminator. (a) MARTA GANs generator is used for the UC-Merced Land-use data set. The input is a 100-D uniform distribution $p_z(z)$ and the output is a 256×256-pixel RGB image. (b) MARTA GANs discriminator is used for the UC-Merced Land-use data set. The discriminator is treated as a feature extractor to extract features from the multifeature layer.

(a) (b)

Fig. 3. Part of exemplary images. (a) Ten random images (real images) from UC-Merced data set. (b) Exemplary images (fake images) produced by the generator trained on UC-Merced using the ℓ_{final} (5) objective.

For the discriminator, the first layer takes input images, including both real and synthesized images. We use convolutions in our discriminator, which allows it to learn its own spatial downsampling. As shown in Fig. 2(b), by performing 4×4 max pooling, 2×2 max pooling, and the identity function separately in the last three convolutional layers, we can produce feature maps that have the same spatial size, 4×4. Then, we concatenate the 4×4 feature maps through channel dimension in the multifeature layer. Finally, the multifeature layer is flattened and fed into a single sigmoid output. The multifeature layer includes two functions: 1) the features used for classification are extracted from the flattened multifeature layer and 2) when training the generator, we use feature matching loss (4) to evaluate the similarities of the features between the fake and real images in the flattened multifeature layer.

We set the kernel sizes to 4×4 and the stride to 2 in all the convolutional and deconvolutional layers, because the deconvolutional layers can avoid uneven overlap when the kernel size is divisible by the stride [11]. In the generator, all layers use ReLU activation except for the output layer, which uses the tanh function. We use LeakyReLU activation in the discriminator for all the convolutional layers; the slope of the leak was set to 0.2. We used batch normalization in both the generator and the discriminator, and the decay factor was 0.9.

III. Experiments

To verify the effectiveness of the proposed method, we trained MARTA GANs on two data sets: the UC-Merced Land-Use data set [12] and the Brazilian Coffee Scenes data set [4]. We carried out experiments on both data sets using a fivefold cross-validation protocol and a regularized linear L2 support vector machine as a classifier. We implemented MARTA GANs in TensorLayer,[1] a deep learning and reinforcement learning library extended from Google TensorFlow [13]. We scaled the input image to the range of $[-1, 1]$ before training. All the models were trained by stochastic gradient descent with a batch size of 64, and we used the Adam optimizer with a learning rate of 0.0002 and a momentum term β_1 of 0.5.

A. UC-Merced Data Set

This data set consists of images of 21 land-use classes (hundered 256×256-pixel images for each class). Some of the images from this data set are shown in Fig. 3(a). We used a moderate data augmentation in this data set by flipping images horizontally and vertically and rotating them by 90° to increase the effective training set size. Training takes approximately 4 h on a single NVIDIA GTX 1080 GPU.

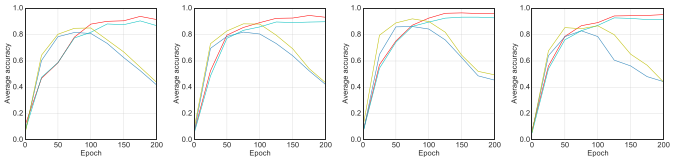
To evaluate the quality of the representations learned by the multifeature layer, we trained on the UC-Merced data and extracted the features from different multifeature layers. To improve the clarity of the expression, we use f_1 to denote the features from the last convolutional layer, f_2 to denote features combined from the last two convolutional layers' features, and so on. Based on the results shown in Fig. 4, we found that f_3 achieved the highest accuracy. These results can be explained by two reasons. First, f_3 has the same high-level information as f_1 and f_2, but it has more mid-level information compared with f_1 and f_2. However, f_4 has too much low-level information, which leads to the "curse of

[1] http://tensorlayer.readthedocs.io/en/latest/

TABLE I

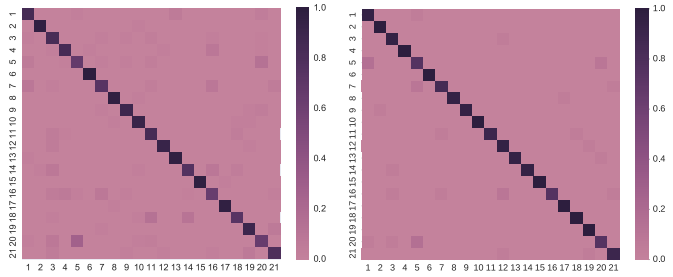
CLASSIFICATION ACCURACY (%) IN THE FORM OF THE MEANS ± STANDARD DEVIATION BARS OF DCGAN AND MARTA GAN FOR EVERY CLASS. THE CLASS LABELS ARE AS FOLLOWS. 1 : MOBILE HOME PARK. 2 : BEACH. 3 : TENNIS COURTS. 4 : AIRPLANE. 5 : DENSE RESIDENTIAL. 6 : HARBOR. 7 : BUILDINGS. 8 : FOREST. 9: INTERSECTION. 10: RIVER. 11: SPARSE RESIDENTIAL. 12: RUNWAY. 13: PARKING LOT. 14: BASEBALL DIAMOND. 15: AGRICULTURAL. 16: STORAGE TANKS. 17: CHAPARRAL. 18: GOLF COURSE. 19: FREEWAY. 20: MEDIUM RESIDENTIAL. 21: OVERPASS

Class	1	2	3	4	5	6	7	8	9	10	11
DCGAN	85 ± 5.0	94 ± 2.2	89 ± 4.2	95 ± 3.5	82 ± 2.7	91 ± 2.2	78 ± 2.7	83 ± 2.7	88 ± 2.7	90 ± 0.0	79 ± 2.2
MARTA GAN	95 ± 3.5	100 ± 0.0	96 ± 4.2	100 ± 0.0	89 ± 4.2	99 ± 2.2	86 ± 6.5	97 ± 2.7	98 ± 2.7	94 ± 2.2	89 ± 2.2
Class	12	13	14	15	16	17	18	19	20	21	
DCGAN	89 ± 4.2	88 ± 2.7	95 ± 3.5	78 ± 4.5	93 ± 2.7	88 ± 2.7	97 ± 2.7	77 ± 2.7	95 ± 5.0	89 ± 4.2	
MARTA GAN	94 ± 4.2	98 ± 2.7	100 ± 0.0	85 ± 5.0	100 ± 0.0	93 ± 2.7	100 ± 0.0	87 ± 5.7	97 ± 5.5	95 ± 5.0	



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

Fig. 4. Performance comparison uses different features. (a) f_1. (b) f_2. (c) f_3. (d) f_4. Red curves: training with ℓ_{final} and with data augmentation. Cyan curves: training with $\ell_{\text{perceptual}}$ and with data augmentation. Yellow curves: training with ℓ_{final} and without data augmentation. Green curves: training with $\ell_{\text{perceptual}}$ and without data augmentation.



(a) (b)

Fig. 5. Confusion matrix of (a) DCGAN and (b) MARTA GAN. The class labels are the same as Table I.

dimensionality." Therefore, the features extracted from the last three convolutional layers in the discriminator resulted in the highest accuracy. As shown in Fig. 4, data augmentation is an effective way to reduce overfitting when training a large deep network. Augmentation generates more training image samples by rotating and flipping patches from original images. We also evaluated the performance between two types of loss: $\ell_{\text{perceptual}}$ [see (2)] and ℓ_{final} [see (5)] and found that using ℓ_{final} achieved the best performance. Synthesized remote sensing images when using ℓ_{final} are shown in Fig. 3(b).

Fig. 5 depicts the confusion matrix of classification results for the two GAN architectures, DCGAN and MARTA GAN. DCGAN and MARTA GAN reached an overall accuracy of $87.76 \pm 0.64\%$ and $94.86 \pm 0.80\%$, respectively. MARTA GAN is approximately 7% better, because it used the multifeature layer to merge the mid-level and global features. To improve the comparison, the accuracy classification performances of the methods for each class are shown in Table I. Compared with DCGAN, MARTA GAN achieves 100.00% accuracy in some scene categories (e.g., Beach, Airplane, and so on). Moreover, MARTA GAN also achieves higher accuracy in some very close classes, such as dense residential, building, medium residential, and sparse residential.

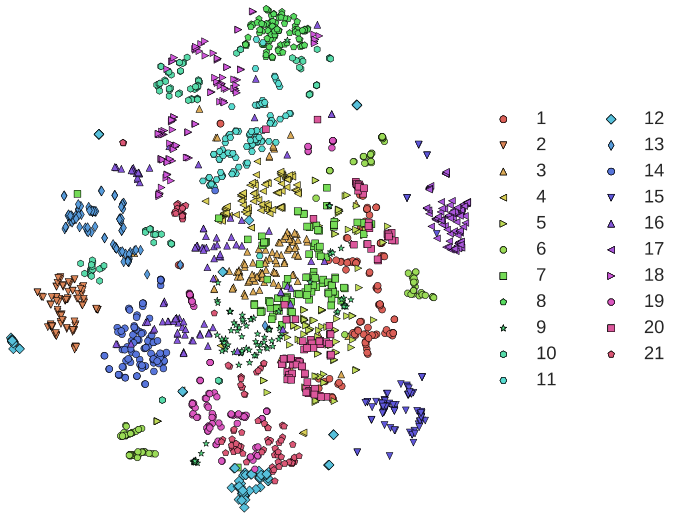


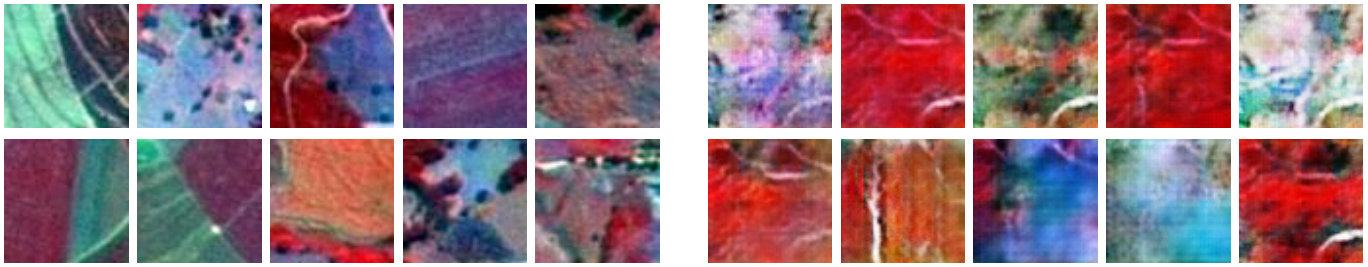
Fig. 6. Two-dimensional feature visualization of image global representations of the UC-Merced data set. The class labels are the same as Table I.

In addition, we visualized the global image representations encoded via MARTA GANs features of the UC-Merced data set. We computed the features for all the scenes of the data set and then used the t-distributed stochastic neighbor embedding algorithm to embed the high-dimensional features in 2-D space. The final results are shown in Fig. 6. This visualization shows that features extracted from the multifeature layer contain abstract semantic information because those close classes are also very close in 2-D space.

Compared with the results of other tested methods, the method proposed in this letter achieves the highest classification accuracy among the unsupervised methods. As shown in Table II, our method outperforms the a sparse coding-based multiple-feature fusion method (SCMF) [14] by 3.82%. When the classification accuracy of our method is compared with logistic regression based feature fusion (LRFF) [15] (an improved unsupervised feature learning algorithm based on spectral clustering), our method outperforms LRFF by more than 4%. Although some of the supervised methods [4], [5] achieved an accuracy above 99%, these methods are fine-tuned from pretrained models, which are usually trained with a large amount of labeled data (such as ImageNet). Compared with those methods, our unsupervised method requires fewer parameters.

B. Brazilian Coffee Scenes Data Set

To evaluate the generalization power of our model, we also performed experiments using the Brazilian Coffee Scenes



(a) (b)

Fig. 7. Parts of exemplary images. (a) Ten random images (real images) from the Brazilian Coffee Scenes data set. (b) Exemplary images (fake images) produced by a generator trained on the Brazilian Coffee Scenes data set using the ℓ_{final} (5) objective.

TABLE II
OVERALL CLASSIFICATION ACCURACY (%) OF REFERENCE AND PROPOSED METHODS ON THE UC-MERCED DATA SET AND COFFEE SCENES DATA SET. OUR RESULT IS IN BOLD

DataSet	Method	Description	Parameters	Accuracy
UC-Merced	SCMF [14]	Unsupervised	-	91.03 ± 0.48
	UFL-SC [15]	Unsupervised	-	90.26 ± 1.51
	OverFeat$_L$ + Caffe [4]	Supervised	205M	99.43 ± 0.27
	GoogLeNet [5]	Supervised	5M	99.47 ± 0.50
	MARTA GANs	**Unsupervised**	**2.8M**	94.86 ± 0.80
Coffee	BIC [4]	Unsupervised	-	87.03 ± 1.07
	OverFeat$_L$+OverFeat$_S$ [4]	Supervised	289M	83.04 ± 2.00
	CaffeNet [5]	Supervised	60M	94.45 ± 1.20
	MARTA GANs	**Unsupervised**	**0.18M**	89.86 ± 0.98

data set [4], which is a composition of scenes taken by the Satellite Pour l'Observation de la Terre (SPOT) sensor in the green, red, and near-infrared bands. This data set has 2876 multispectral high-resolution scenes. It includes 1438 tiles of coffee and 1438 tiles of noncoffee with a 64×64-pixel resolution. Fig. 7(a) shows some examples of this data set. We did not use data augmentation on this data set because it contains sufficient data to train the network.

Table II shows the results obtained with the proposed method. In general, the results are significantly worse than those on the UC-Merced data set, despite reducing the classification from a 21-class to a 2-class problem. Brazilian Coffee Scenes is a challenging data set because of the high intraclass variability caused by different crop management techniques, different plant ages, and spectral distortions and shadows. Nevertheless, our results are better than that of border-interior pixel classification [4].

IV. CONCLUSION

This letter introduced a representation learning algorithm called MARTA GANs. In contrast to previous approaches that require supervision, MARTA GANs are completely unsupervised; it can learn interpretable representations even from challenging remote sensing data sets. In addition, MARTA GANs introduce a new multiple-feature-matching layer that learns multiscale spatial information for high-resolution remote sensing. Other possible future extensions to the work described in this letter include: producing high-quality samples of remote sensing images using the generator and classifying remote sensing images in a semisupervised manner to improve classification accuracy.

REFERENCES

[1] J. Sivic *et al.*, "Video Google: A text retrieval approach to object matching in videos," in *Proc 9th IEEE Int. Conf. Comput. Vis.*, Oct. 2003, pp. 1470–1477.

[2] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2. Jun. 2006, pp. 2169–2178.

[3] M. Lienou, H. Maitre, and M. Datcu, "Semantic annotation of satellite images using latent Dirichlet allocation," *IEEE Geosci. Remote Sens. Lett.*, vol. 7, no. 1, pp. 28–32, Jan. 2010.

[4] O. A. B. Penatti, K. Nogueira, and J. A. dos Santos, "Do deep features generalize from everyday objects to remote sensing and aerial scenes domains?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2015, pp. 44–51.

[5] K. Nogueira, O. A. B. Penatti, and J. A. dos Santos, "Towards better exploiting convolutional neural networks for remote sensing scene classification," *Pattern Recognit.*, vol. 61, pp. 539–556, Jan. 2017.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[7] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, no. 12, pp. 3371–3408, Dec. 2010.

[8] A. Makhzani and B. Frey, "*k*-sparse autoencoders." Unpublished paper, 2013. [Online]. Available: https://arxiv.org/abs/1312.5663

[9] I. J. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[10] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," Unpublished paper, 2015. [Online]. Available: https://arxiv.org/abs/1511.06434

[11] A. Odena, V. Dumoulin, and C. Olah, "Deconvolution and checkerboard artifacts," *Distill*, vol. 1, no. 10, Oct. 2016. [Online]. Available: http://distill.pub/2016/deconv-checkerboard

[12] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *Proc. ACM 18th SIGSPATIAL Int. Conf. Adv. Geogr. Inf. Syst.*, 2010, pp. 270–279.

[13] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous distributed systems." Unpublished paper, 2016. [Online]. Available: https://arxiv.org/abs/1603.04467

[14] G. Sheng, W. Yang, T. Xu, and H. Sun, "High-resolution satellite scene classification using a sparse coding based multiple feature combination," *Int. J. Remote Sens.*, vol. 33, no. 8, pp. 2395–2412, 2012.

[15] F. Hu, G.-S. Xia, Z. Wang, X. Huang, L. Zhang, and H. Sun, "Unsupervised feature learning via spectral clustering of multidimensional patches for remotely sensed scene classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 5, pp. 2015–2030, May 2015.