A COMPARATIVE ANALYSIS OF UNSUPERVISED AND SEMI-SUPERVISED REPRESENTATION LEARNING FOR REMOTE SENSING IMAGE CATEGORIZATION

P. J. Soto[1], J. D. Bermudez[1], P. N. Happ[1], R. Q. Feitosa[1,2]

[1] Dept. of Electrical Engineering, Pontifical Catholic University of Rio de Janeiro - (psoto, bermudez, patrick, raul)@ele.puc-rio.br
[2] Rio de Janeiro State University.

ICWG II/III: Pattern Analysis in Remote Sensing

KEY WORDS: Generative Adversarial Networks, Deep Learning, Semi-supervised Learning, Representation Learning

ABSTRACT:

This work aims at investigating unsupervised and semi-supervised representation learning methods based on generative adversarial networks for remote sensing scene classification. The work introduces a novel approach, which consists in a semi-supervised extension of a prior unsupervised method, known as MARTA-GAN. The proposed approach was compared experimentally with two baselines upon two public datasets, *UC-MERCED* and *NWPU-RESISC45*. The experiments assessed the performance of each approach under different amounts of labeled data. The impact of fine-tuning was also investigated. The proposed method delivered in our analysis the best overall accuracy under scarce labeled samples, both in terms of absolute value and in terms of variability across multiple runs.

1. INTRODUCTION

Over the last decades, much of the effort involved in deploying automatic image classification algorithms has been invested in designing and manually selecting custom features for a target application. In this sense, the use of *Bag-of-Visual-Words* (BoVW) was one of the first attempts in the field (Yang , Newsam, 2010), followed later by different classifiers like Random Forest (RF) and Support Vector Machines (SVM) (Helber et al., 2017). Recently, Deep Learning (DL) techniques have become the dominant trend in image classification (Simonyan , Zisserman, 2014, Szegedy et al., 2015, Cheng et al., 2018), mainly due to their ability to automatic learn discriminative features directly from data (LeCun et al., 2015, Krizhevsky et al., 2012, Penatti et al., 2015, Nogueira et al., 2017), when labeled samples are abundant.

Although recent years have witnessed an increase of Earth observation data, remote sensing labeled data still falls short of the demands imposed by DL-based techniques. Mainly because of the high costs involved in field survey and the required labor-intensive visual interpretation.

In this sense, transfer learning (Pan , Yang, 2010, Weiss et al., 2016) and unsupervised deep learning techniques, such as Stacked Denoising Autoencoders, Convolutional Autoencoders and Deep Belief Networks (Liang et al., 2017, Romero et al., 2016, Zou et al., 2015), emerged as attractive alternatives. In transfer learning, networks already trained using huge data-sets are reused in problems where the labeled data is limited by performing a fine tuning (Nogueira et al., 2017) of certain layers. On the other hand, unsupervised methods do not require any labeled data for the learning process.

In the last few years, Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) have been catching the community attention due to their ability to learn data distributions through an unsupervised two-player min-max game performed by two different networks: a generator and a discriminator.

Considering the power of GANs for unsupervised learning, Lin *et al* (Lin et al., 2017) proposed a Multiple-Layer Feature-Matching GANs architecture (MARTA-GANs) for feature learning. In short, MARTA-GANs capture latent features from the discriminator network, which can be later used as input to a classifier. This method presented substantial improvements in comparison with others unsupervised feature learning models.

Aiming to exploit cases where few labeled samples are available, (Springenberg, 2015) proposed to work with semi-supervised GAN (SS-GAN) algorithms. More specifically, they introduced the categorical generative adversarial networks (CatGANs) for image classification. This model was extended in (Salimans et al., 2016) to improve its convergence. Specifically, they proposed the feature matching term and the mini-batch discrimination concept among others modifications. Later, the SS-GAN approach was adapted to remote sensing data applications, such as object detection (Chen et al., 2018) and pixel-wise PolSAR (Liu et al., 2018) and hyperspectral (He et al., 2017, Zhan et al., 2018) image classification. However, despite the efforts of (Salimans et al., 2016), SS-GANs still present some convergence problems, mostly when the number of unlabeled samples is much larger than the labeled ones.

Motivated by this scenario, we introduce in this paper a Semi-Supervised Representation Learning GAN (SSRL-GAN), which, although conceptually similar to SS-GANs, presents a different training strategy and adaptations in architecture. In short, SSRL-GANs present an external classifier allowing the use of binary cross-entropy cost functions for supervised and unsupervised stages. With these changes, we observed an improvement in the convergence of the model and in the classification performance, mainly when less labeled samples were used to train the model.

We further analyze and compare different alternatives for remote sensing image categorization when a limited number

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-IV-2-W7-167-2019

of labeled samples is available. First, we take the MARTA-GAN (Lin et al., 2017) as baseline, which is an unsupervised learning method. Then, we compare it with two semi-supervised approaches: the Semi-Supervised GANs, as presented in (Salimans et al., 2016), and the Semi-Supervised Representation Learning GAN proposed in this work. Additionally, we evaluate how these methods behave when more labeled samples are added in the training set. And finally, we adopt a classic fine tuning approach, using only labeled data, to investigate if their performance can still be enhanced.

The rest of this paper is organized as follows. Section 2 briefly describes the fundamentals underlying GANs. A detailed description of each assessed method is the subject of Section 3. The experimental protocol is reported in Section 4, while Section 5 shows the results obtained by the experiments. Finally, Section 6 summarizes the main conclusions and indicates future directions.

2. GENERATIVE ADVERSARIAL NETWORKS (GANS)

GANs, introduced by (Goodfellow et al., 2014), constitute a class of unsupervised machine learning models composed by two neural networks: the generator, which synthesizes realistic images and the discriminator, which tries to correctly discern between synthesized and real images.

A min-max game procedure is used to train these neural networks. The Generator learns a function $\mathcal{G}$ that maps samples of a known random distribution $p(z)$ into samples of a distribution $p_{model}(x)$, which the Discriminator $\mathcal{D}$ can hardly distinguish from a sample of a given data distribution $p_{data}(x)$. The Discriminator, in turn, is trained to learn a function $\mathcal{D}$ that distinguishes whether a sample comes from $p_{data}(x)$ or $p_{model}(x)$. The optimal mapping function $\mathcal{G}^*$ can be found by solving the following equation:

$$\mathcal{G}^* = \arg\min_{\mathcal{G}} \max_{\mathcal{D}} \mathcal{L}(\mathcal{G}, \mathcal{D}), \tag{1}$$

where $\mathcal{L}(\mathcal{G}, \mathcal{D})$ is the GAN loss function defined by,

$$\begin{aligned}\mathcal{L}(\mathcal{G}, \mathcal{D}) = &\mathbb{E}_{x\sim p_{data}(x)}[\log(\mathcal{D}(x))] \\ &+ \mathbb{E}_{z\sim p_{(z)}}[\log(1-\mathcal{D}(\mathcal{G}(z)))],\end{aligned} \tag{2}$$

where $\mathbb{E}$ and $\log$ are the expectation and logarithmic operators, respectively, and z is a random noise vector, which follows a known noise distribution $p(z)$, typically uniform or Gaussian.

3. EVALUATED METHODS

This section presents the four methods assessed in this paper for remote sensing image categorization with few labeled samples available. In the following, we describe the unsupervised MARTA-GAN, the Semi-Supervised GAN, the Semi-Supervised Representation Learning GAN, and the Fine Tuning applied in the Discriminator of all methods.

3.1 Multiple-Layer Feature Matching GANs (MARTA-GANs)

MARTA-GAN (Lin et al., 2017) is an unsupervised representation learning algorithm that relies on the same GAN's min-max game to learn discriminative features $f(x)$. Like Deep Convolutional GANs (Radford et al., 2015), the Generator and the Discriminator are convolution networks trained to minimize a modified loss function $\mathcal{L}(\mathcal{G}, \mathcal{D})$ given by the equation:

$$\begin{aligned}\mathcal{L}(\mathcal{G}, \mathcal{D}) = &\mathbb{E}_{x\sim p_{data}(x)}[\log(\mathcal{D}(x))] \\ &+ \mathbb{E}_{z\sim p_{(z)}}[\log(1-\mathcal{D}(\mathcal{G}(z)))] \\ &+ ||\mathbb{E}_{x\sim p_{data}(x)}[f(x)] - \mathbb{E}_{z\sim p_{(z)}}[f(\mathcal{G}(z))]||_2^2\end{aligned} \tag{3}$$

The third term, called *feature matching loss*, is added to the GAN loss function to favor similarity between the generated and real images. The learned features $f(x)$, named in (Lin et al., 2017) *multi-feature layer*, result from concatenating the outputs of the three last convolutional layers of the discriminator network.

3.2 Semi-Supervised GANs (SS-GANs)

SS-GANs (Salimans et al., 2016) exploit the available labeled data together with the unlabeled data to perform a semi-supervised learning. The Discriminator output is changed from 1 neuron to $K+1$ neurons, where the first K neurons are used to classify the real labeled samples into one out of the K classes present in the data-set and the $(K+1)-th$ neuron computes the probability that the input sample is real or fake, i.e. synthesized by the GAN. The training function for the SS-GANs becomes:

$$\mathcal{L}(\mathcal{G}, \mathcal{D}) = \mathcal{L}_{supervised} + \mathcal{L}_{unsupervised}, \tag{4}$$

where:

$$\mathcal{L}_{supervised} = -\mathbb{E}_{x\sim p_{data}(x)}[log(\mathcal{D}(x, y|y < K+1))] \tag{5}$$

and

$$\begin{aligned}\mathcal{L}_{unsupervised} = &-\{\mathbb{E}_{x\sim p_{data}(x)}[\log(1-\mathcal{D}(x, y|y = K+1))] \\ &+ \mathbb{E}_{z\sim p_{(z)}}[\log(\mathcal{D}(\mathcal{G}(z, y|y = K+1)))]\} \\ &+ ||\mathbb{E}_{x\sim p_{data}(x)}[f(x)] - \mathbb{E}_{z\sim p_{(z)}}[f(\mathcal{G}(z))]||_2^2\end{aligned} \tag{6}$$

Observe that, $\mathcal{L}(\mathcal{G}, \mathcal{D})$ is a composition of the standard supervised loss function $\mathcal{L}_{supervised}$ with the unsupervised loss $\mathcal{L}_{unsupervised}$, which actually represents the standard GAN min-max game, including the well known *feature matching loss*. The optimal solution can be found by minimizing these two losses jointly.

3.3 Semi-Supervised Representation Learning GANs (SSRL-GANs)

The proposed SSRL-GANs differs from the SS-GANs by an auxiliary classifier not embedded in the Discriminator. Thus, the Discriminator is responsible for verifying if the input sample is real or fake, whereas the Classifier evaluates how good are the features at the *multi-feature layer* for the classification of the available labeled samples. The architecture of the SSLR-GAN is shown in Figure 1 and involves three networks: Generator, Discriminator and Classifier.

The training process is divided into two consecutive stages, unsupervised and supervised, depending on whether the

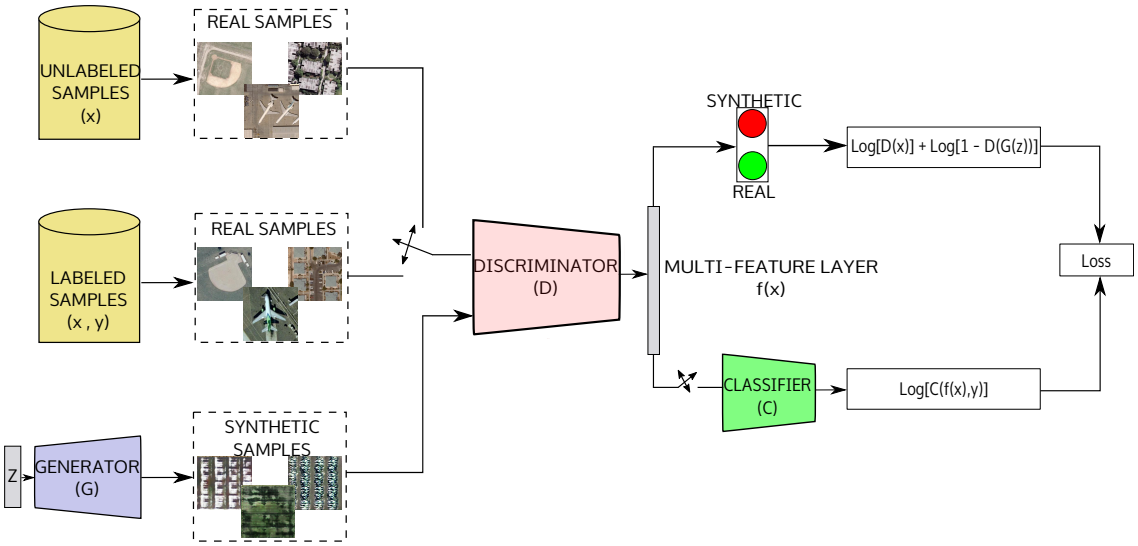


Figure 1. Overview of the SSRL-GAN method. The Generator ($\mathcal{G}$) learns to synthesize images to fool the Discriminator ($\mathcal{D}$), which learns to distinguish between real and synthesized images. The semi-supervised procedure is performed by switching between unlabeled and labeled real images. When labeled images are used, features $f(x)$ are extracted from the multi-feature layer and used as input to the Classifier ($\mathcal{C}$) which will influence the GAN objective function.

training data is labeled or not. In the first, pure unlabeled data is used in each mini-batch while in the second only labeled samples are employed. The Generator is trained in the same way for both stages, since it does not rely on labels. Thus, while the parameters of the Discriminator are fixed, the parameters of the Generator are updated to synthesize images realistic enough to fool the Discriminator. Formally, it is about minimizing the following cost function which also includes the *feature matching loss* term:

$$\begin{aligned}\mathcal{L}_{\mathcal{G}} &= \mathbb{E}_{z\sim p_{(z)}}[log(1-\mathcal{D}(\mathcal{G}(z)))] \\ &+ ||\mathbb{E}_{x\sim p_{data}(x)}[f(x)] - \mathbb{E}_{z\sim p_{(z)}}[f(\mathcal{G}(z))]||_2^2\end{aligned} \tag{7}$$

Analogously, while the Discriminator is being trained, the Generator parameters are kept fixed. Thus, in the unsupervised stage, the Discriminator parameters are updated so that the function $\mathcal{L}_{\mathcal{D}}$ is maximized for real samples and minimized for synthetic ones, as stated below:

$$\begin{aligned}\mathcal{L}_{\mathcal{D}} &= \mathbb{E}_{x\sim p_{data}(x)}[\log(\mathcal{D}(x))] \\ &+ \mathbb{E}_{z\sim p_{(z)}}[\log(1-\mathcal{D}(\mathcal{G}(z)))]\end{aligned} \tag{8}$$

In the supervised stage, the function $\mathcal{L}_{\mathcal{D}}$ is modified to include a new term that tries to maximize the probabilities $\mathcal{C}(f(x), y)$ assigned by the Classifier to the real class y of each sample x, as shown in Equation 9.

$$\begin{aligned}\mathcal{L}_{\mathcal{D}} &= \mathbb{E}_{x\sim p_{data}(x)}[\log(\mathcal{D}(x))] \\ &+ \mathbb{E}_{z\sim p_{(z)}}[\log(1-\mathcal{D}(\mathcal{G}(z)))] \\ &+ \mathbb{E}_{x\sim p_{data}(x)}[\log(\mathcal{C}(f(x),y))]\end{aligned} \tag{9}$$

Aiming to minimize this expression, the Discriminator will tend to produce more discriminative and representative features. Since the Classifier network requires label information for training, it is not used in the unsupervised stage. In the supervised stage, it is trained using the features $f(x)$ learned by the Discriminator considering only the real labeled data. In summary, the whole method can be mathematically described as:

$$\mathcal{G}^* = \arg\min_{\mathcal{G}} \max_{\mathcal{D}} \max_{\mathcal{C}} \mathcal{L}(\mathcal{G}, \mathcal{D}, \mathcal{C}) \tag{10}$$

where $\mathcal{L}(\mathcal{G}, \mathcal{D}, \mathcal{C})$ is the GAN objective function defined by,

$$\begin{aligned}\mathcal{L}(\mathcal{G}, \mathcal{D}, \mathcal{C}) &= \mathbb{E}_{x\sim p_{data}(x)}[\log(\mathcal{D}(x))] \\ &+ \mathbb{E}_{z\sim p_{(z)}}[\log(1-\mathcal{D}(\mathcal{G}(z)))] \\ &+ ||\mathbb{E}_{x\sim p_{data}(x)}[f(x)] - \mathbb{E}_{z\sim p_{(z)}}[f(\mathcal{G}(z))]||_2^2 \\ &+ \mathbb{E}_{x\sim p_{data}(x)}[\log(\mathcal{C}(f(x),y))]\end{aligned} \tag{11}$$

3.4 Fine Tuning

We further tested if the features learned by the aforementioned methods could be improved by a subsequent fine-tuning step. For MARTA-GAN and SSRL-GAN the original classification layer was replaced by a *softmax* multiclass classification layer. For SS-GAN, we kept the first K neurons of the Discriminator output layer. Then, a new supervised training was carried out using the available labelled samples.

4. EXPERIMENTAL ANALYSIS

The experiments performed in this work aimed to evaluate the representations learned by the methods described above, specifically: MARTA-GAN, SS-GAN, SSRL-GAN and the fine tuned version of these algorithms.

Once the methods were trained, we took the features extracted from their respective *multi-feature layers* for image categorization. As in (Lin et al., 2017), we used a Support Vector Machine (SVM) (Hearst et al., 1998) for this purpose. The SVM was trained on the same labeled samples available on the training set.

4.1 Datasets

We assessed the methods using two public datasets for remote sensing image categorization.

Agricultural Airplanes Baseball diamond Beach

(a) UC-MERCED

Basketball court Bridge Church Commercial area

(b) NWPU-RESISC45

Figure 2. Examples of images taken randomly from both datasets. (a) *UC-MERCED* and (b) *NWPU-RESISC45*.

Layer	Out shape
Input	(14336,1)
$\text{FcA}_1(.)$	(512,1)
Fc(.)	(N. Classes,1)
softmax(.)	(N. Classes,1)

Table 1. Architecture of the SSRL-GAN Classifier.

The first dataset was the *UC MERCED Land Use Dataset*[1] (Yang , Newsam, 2010). It comprises 21 land-use classes. Each 256×256 pixel image has a spatial resolution of 0.3 m per pixel. For each class, 100 images were manually extracted from large images downloaded from the USGS National Map of different urban areas around the United States. Some image samples of this data-set are shown in Figure 2(a).

The second dataset used in our experiments was the *NWPU-RESISC45* (Cheng et al., 2017). This dataset[2] contains 31500 remote sensing images of size 256×256 pixels and spatial resolution from about 30 m to 0.2 m per pixel for most classes. A total of 45 scene classes are represented in the dataset. For each class, 700 images were extracted from Google Earth by experts in the remote sensing field. Figure 2(b) shows samples of these images.

4.2 Network Architectures

The architecture of the Generator and Discriminator networks were essentially the same as that of the MARTA-GAN (Lin et al., 2017). The Classifier, used only in the SSRL-GANs, was a Multi-Layer Perceptron (MLP) network, which took as input the feature vector at the *multi-feature layer* of the Discriminator and propagated it into a hidden layer with 512 units empirically chosen and using a rectified linear unit (ReLU) as activation function. Its output layer implemented a *softmax* function and had as many units as the number of classes in the dataset.

Layer	Out shape
Input	(100,1)
Fc(.)	(8192,1)
Reshape(.)	(4,4,512)
$\text{BA}_1(.)$	(4,4,512)
$\text{DBA}_1(256,4,2)$	(8,8,256)
$\text{DBA}_1(128,4,2)$	(16,16,128)
$\text{DBA}_1(64,4,2)$	(32,32,64)
$\text{DBA}_1(32,4,2)$	(64,64,32)
$\text{DBA}_1(16,4,2)$	(128,128,16)
D (3, 4 , 2)	(256,256,3)
tanh(.)	(256,256,3)

Table 2. Architecture of the Generator for the three methods.

The three network architectures (Classifier, Generator, and Discriminator) are described in more details in Tables 1, 2 and 3. The symbols denote for each layer, convolution (C), deconvolution (D), batch normalization (B), ReLU (A_1), Leaky ReLU (A_2), MaxPooling (P), Flatten (F) and Fully Connected (Fc). The number of filters, filter's dimension and the convolution stride are indicated in parenthesis. All filters were square and the stride was equal in horizontal and vertical directions. The *multi-feature layer* resulted from the concatenation of F_1, F_2 and F_3 which were the product of a flattening operation over feature maps at different scales in the network.

4.3 Experimental Protocol

We assessed the methods using different amounts of available labeled samples. To this end, we used two public datasets for remote sensing image categorization.

[1] http://weegee.vision.ucmerced.edu/datasets/landuse.html
[2] http://www.escience.cn/people/JunweiHan/NWPU-RESISC45.html

Layer	Out shape
Input	(256,256,3)
$\mathrm{CA}_2(16, 4, 2)$	(128,128,16)
$\mathrm{CBA}_2(32, 4, 2)$	(64,64,32)
$\mathrm{CBA}_2(64, 4, 2)$	(32,32,64)
$\mathrm{CBA}_2(128, 4, 2)$	(16,16,128)
$\mathrm{PF}_1(4, 4)$	(2048,1)
$\mathrm{CBA}_2(256, 4, 2)$	(8,8,256)
$\mathrm{PF}_2(2, 2)$	(4096,1)
$\mathrm{CBA}_2(512, 4, 2)$	(4,4,512)
$\mathrm{F}_3(.)$	(8192,1)
Feature$[F_1, F_2, F_3]$	(14336,1)
MARTA-GANs: sigmoid(.)	(1,1)
SSRL-GANs: sigmoid(.)	(1,1)
SS-GANs: softmax(.)	(K + 1, 1)

Table 3. Architecture of the Discriminator for the three methods.

Each database was divided into three sets: *Train*, *Test* and *Aux*, corresponding to 76%, 5%, 19% of all samples for *UC-MERCED* and of 70%, 20% and 10% for *NWPU-RESISC45*, respectively.

All methods were trained with a batch size of 64 samples using the Adam optimizer (Kingma , Ba, 2014), which parameters learning rate and momentum β_1 were set to 0.0002 and 0.5, respectively. The α parameter in the Leaky ReLU activation function was set to 0.2. The terms that make up the cost functions of all methods had the same relevance, been setting each importance coefficient to one. As in (Lin et al., 2017), we scaled the input images in the range of $[-1, 1]$ before training and testing. Also, we applied the early stopping regularization procedure to avoid overfitting. The patience parameter, which controls the number of epochs without improvements in the validation loss, was set to 10. Each experiment was executed 5 times in order to evaluate the sensitivity of the methods to the initial solution of trainable parameters.

To verify the influence of the number of labeled samples in the performance of each method, our experiments were carried out in two different protocols. We used the same *Train* set in both protocols in the unsupervised learning stage. The protocols differed in the number of labeled samples used for the supervised training stage of SS-GANs and SSRL-GANs, and also for training the SVM.

In Protocol 1, we used for the supervised stage the *Aux* set, as described before. In Protocol 2 we applied vertical and horizontal flips, rotations and data replication to augment the number of labeled samples. This way, the number of labeled samples in Protocol 2 was about seven times larger than in Protocol 1. The methods were implemented in TensorLayer[3] on a NVIDIA Titan XP GPU.

5. RESULTS

Figure 3 summarizes the results for the *UC-MERCED* and *NWPU-RESISC45* datasets in terms of *Overall Accuracy* (OA). The bar plots in Figure 3a to 3b refer to *UC-MERCED*, whereas Figure 3c to 3d relates to *NWPU-RESISC45*.

The results for the fine tuned version of the evaluated methods are presented in the Figure 3b and 3d for *UC-MERCED* and *NWPU-RESISC45*, respectively. In these figures the suffix FT denotes the results obtained after fine-tuning. Each bar group indicates the median OA over all runs for each method and protocol. The plots also show, in black, the highest and the lowest OA value recorded in our experiments in each case.

As expected, the augmentation of labeled data improved the accuracy, in some cases remarkably. This can be seen by comparing corresponding bars within each plot. Data augmentation affected favorably even the MARTA-GAN results, an unsupervised representation learning method. The improvement for this method came from the SVM classifier, which profited from the extra labeled samples. The gain brought by labeled data augmentation ranged from 4.6% for MARTA-GAN on *UC-MERCED*, to 19.2% for SS-GAN FT on *NWPU-RESISC45*.

A comparison of plots related to the same dataset reveals that fine-tuning also improved the accuracy consistently. Also the variability of the results across multiple runs reduced thanks to fine-tuning. The improvement in terms of OA ranged from 0.3%, for SS-GAN in Protocol 1, to 4,8% for MARTA-GAN in Protocol 2.

However, the key issue in this analysis is the comparison of the three methods in each scenario. The proposed method, SSRL-GAN, was consistently superior to MARTA-GAN in all experiments. Data augmentation and fine-tuning affected MARTA-GAN and SSRL-GAN performance similarly on both datasets. Even so, the proposed method always outperformed its unsupervised counterpart. Thus, the exploitation of labeled samples in MARTA-GAN as proposed in SSRL-GAN, was generally beneficial in all variants tested in our experiments.

SS-GAN presented a unique behavior. In all experiments conducted under Protocol 1 it presented the worst results among all methods, both in terms of absolute values and in terms of variability. However, when we increased the number of labeled samples, moving to Protocol 2, SS-GAN became consistently the best performing method.

These results indicate that SS-GAN was among all tested methods the most sensitive to the so-called, small sample size problem. In other words, the experiments indicated that under conditions of greater scarcity of labeled data the SSRL-GAN presented the best results among all analyzed methods on both databases. Additionally, in conditions of more abundance of labeled data the proposed method was overcome by SS-GAN.

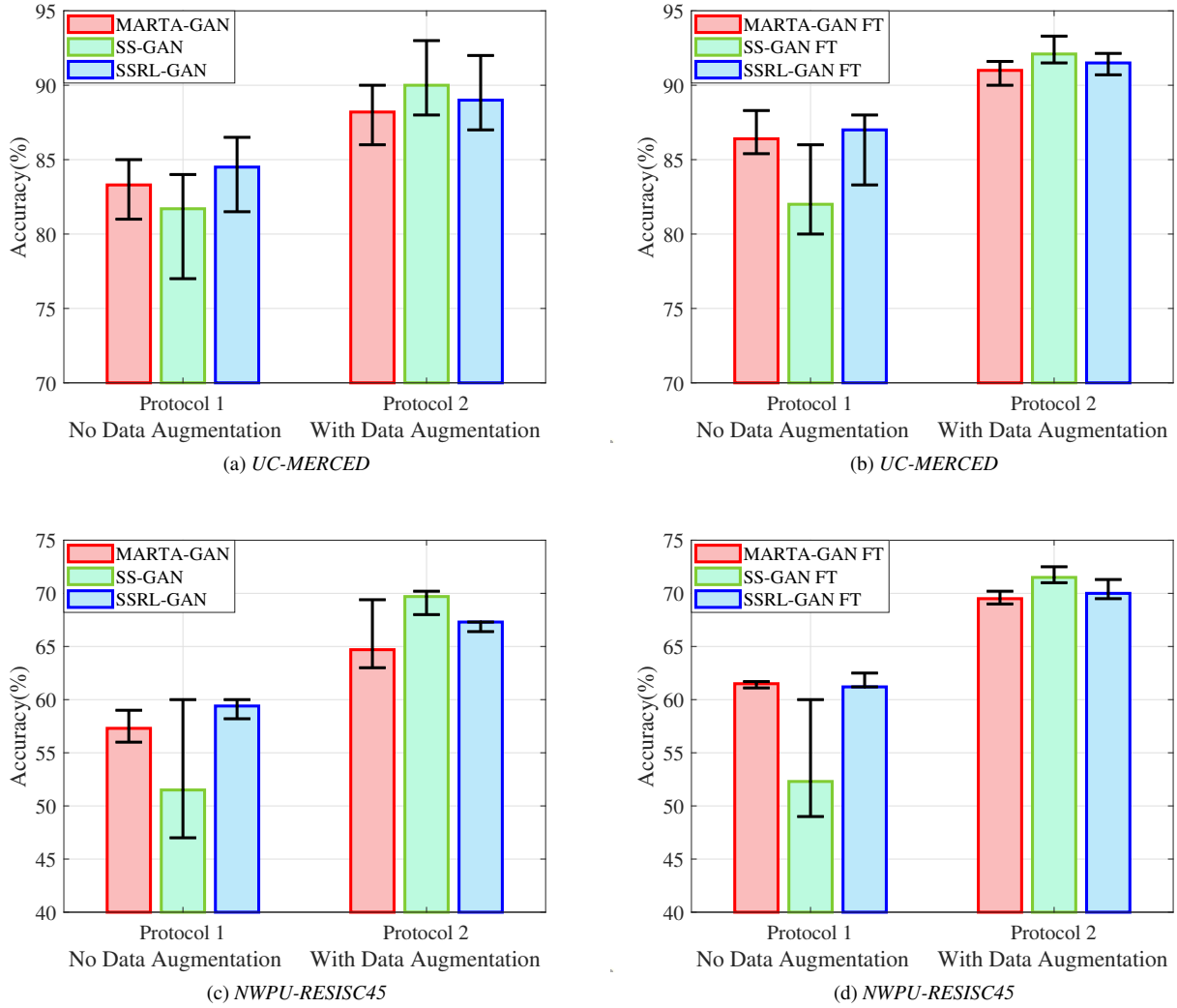
6. CONCLUSIONS

In this work, we performed a comparative analysis of semi-supervised representation learning methods for remote sensing scene classification. We further introduced a novel semi-supervised approach based on Generative Adversarial Networks (GANs).

The methods were evaluated on two public datasets. We took as baseline an unsupervised and a semi-supervised method, both based on GANs. The experimental analysis indicated that the features learned by the proposed method allowed to achieve better accuracy than the baselines when the amount of labeled data was small.

The experimental analysis also revealed that a fine-tuning step further improved results in all tested methods.

[3] https://tensorlayer.readthedocs.io/en/stable/



(a) *UC-MERCED* (b) *UC-MERCED*

(c) *NWPU-RESISC45* (d) *NWPU-RESISC45*

Figure 3. Overall Accuracy results in (%): FT in the plots on the right indicates fine-tuning

As a continuation of the present research, we intend to explore the conclusions drawn from this work for solutions based on GANs for other applications.

ACKNOWLEDGEMENTS

This work is supported by CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico), CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) and NVIDIA corporation.

REFERENCES

Chen, G., Liu, L., Hu, W., Pan, Z., 2018. Semi-supervised object detection in remote sensing images using generative adversarial networks. *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*, IEEE, 2503–2506.

Cheng, G., Han, J., Lu, X., 2017. Remote sensing image scene classification: benchmark and state of the art. *Proceedings of the IEEE*, 105, 1865–1883.

Cheng, G., Yang, C., Yao, X., Guo, L., Han, J., 2018. When deep learning meets metric learning: remote sensing image scene classification via learning discriminative CNNs. *IEEE Transactions on Geoscience and Remote Sensing*, 56, 2811–2821.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative adversarial nets. *Advances in Neural Information Processing Systems*, 2672–2680.

He, Z., Liu, H., Wang, Y., Hu, J., 2017. Generative Adversarial Networks-Based Semi-Supervised Learning for Hyperspectral Image Classification. *Remote Sensing, 1411.1784*, 10, 1042–1068.

Hearst, M. A., Dumais, S. T., Osuna, E., Platt, J., Scholkopf, B., 1998. Support vector machines. *IEEE Intelligent Systems and their Applications*, 13, 18–28.

Helber, Patrick, Bischke, Benjamin, Dengel, Andreas, Borth, Damian, 2017. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *arXiv preprint arXiv:1709.00029*.

Kingma, D. P., Ba, J., 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Krizhevsky, A., Sutskever, I., Hinton, G. E., 2012. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 1097–1105.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. *Nature*, 521, 436.

Liang, P., Shi, W., Zhang, X., 2017. Remote Sensing Image Classification Based on Stacked Denoising Autoencoder. *Remote Sensing*, 10, 16.

Lin, D., Fu, K., Wang, Y., Xu, G., Sun, X., 2017. MARTA GANs: Unsupervised representation learning for remote sensing image classification. *IEEE Geoscience and Remote Sensing Letters*, 14, 2092–2096.

Liu, M., Hu, Y., Wang, S., Guo, Y., Hou, B., Jiao, L., Hou, X., 2018. Fully convolutional semi-supervised gan for polsar classification. *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*, IEEE, 621–624.

Nogueira, K., Penatti, O. A. B., dos Santos, J. A., 2017. Towards better exploiting convolutional neural networks for remote sensing scene classification. *Pattern Recognition*, 61, 539–556.

Pan, S. J., Yang, Q., 2010. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22, 1345–1359.

Penatti, O. A. B., Nogueira, K., dos Santos, J. A., 2015. Do deep features generalize from everyday objects to remote sensing and aerial scenes domains? *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 44–51.

Radford, A., Metz, L., Chintala, S., 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.

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, 1349–1362.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, S., 2016. Improved techniques for training gans. *Advances in Neural Information Processing Systems*, 2226–2234.

Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Springenberg, J. T, 2015. Unsupervised and semi-supervised learning with categorical generative adversarial networks. *arXiv preprint arXiv:1511.06390*.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going deeper with convolutions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1–9.

Weiss, K., Khoshgoftaar, T. M., Wang, D., 2016. A survey of transfer learning. *Journal of Big Data*, 3, 9.

Yang, Y., Newsam, S., 2010. Bag-of-visual-words and spatial extensions for land-use classification. *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, ACM, 270–279.

Zhan, Y., Wu, K., Liu, W., Qin, J., Yang, Z., Medjadba, Y., Wang, G., Yu, X., 2018. Semi-supervised classification of hyperspectral data based on generative adversarial networks and neighborhood majority voting. *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*, IEEE, 5756–5759.

Zou, Q., Ni, L., Zhang, T., Wang, Q., 2015. Deep Learning Based Feature Selection for Remote Sensing Scene Classification. *IEEE Geosci. Remote Sensing Lett.*, 12, 2321–2325.