GANs based Conditional Aerial Images Generation for Imbalanced Learning

Itzel Belderbos[1], Tim de Jong[2], and Mirela Popa[1]

[1] Department of Data Science and Knowledge Engineering, Faculty of Science and Engineering, Maastricht University, Maastricht 6200 MD, The Netherlands,
itzelbelderbos@hotmail.com,
mirela.popa@maastrichtuniversity.nl
[2] Statistics Netherlands, Heerlen 6401 CZ, The Netherlands tja.dejong@cbs.nl

Abstract. In this paper, we examine whether we can use Generative Adversarial Networks as an oversampling technique for a largely imbalanced remote sensing dataset containing solar panels, endeavoring a better generalization ability on another geographical location. To this cause, we first analyze the image data by using several clustering methods on latent feature information extracted by a fine-tuned VGG16 network. After that, we use the cluster assignments as auxiliary input for training the GANs. In our experiments we have used three types of GANs: (1) conditional vanilla GANs, (2) conditional Wasserstein GANs, and (3) conditional Self-Attention GANs. The synthetic data generated by each of these GANs is evaluated by both the Fréchet Inception Distance and a comparison of a VGG11-based classification model with and without adding the generated positive images to the original source set. We show that all models are able to generate realistic outputs as well as improving the target performance. Furthermore, using the clusters as a GAN input showed to give a more diversified feature representation, improving stability of learning and lowering the risk of mode collapse.

Keywords: generative adversarial networks · imbalanced learning · deep learning.

1 Introduction

In recent years, remote sensing data has become increasingly accessible and therefore prominent as a source of fine-grained information about the globe. Recent improvements in supervised classification of remote sensing images have also made this type of data more interesting for a wide range of application fields—official statistics in particular. In this paper, we focus on the improvement of CNNs trained for classification of roof-top solar panel installations in aerial images. A well-know problem in remote sensing is the generalization ability of deep learning models across various geographical areas. One of the frequently occurring underlying causes is the class imbalance problem: the binary classification model is biased towards the over-represented class when the class distribution is imbalanced [3]. The bias towards the majority class is even more severe in the case of high-dimensional data, such as images. Additionally, an imbalance is notably prevalent in the field of solar panel recognition, as the number of rooftops without solar panels largely outnumbers the number of rooftops with a solar panel, making it more expensive to obtain positive samples for an aerial image dataset. Acquiring more labeled data for the training set in order to improve this balance is expensive and time-consuming, which leads to the need for semi-supervised or unsupervised learning techniques. There are various techniques which aim to address this issue, such as oversampling, undersampling and ensemble learning. Oversampling techniques, and in particular Synthetic Minority Oversampling Technique (SMOTE), are most widely used. SMOTE identifies the k nearest neighbors for every minority sample based on the Euclidean distance. Nonetheless, for high-dimensional data such as RGB images, too many dimensions lead to every sample to appear equidistant. As these techniques are more focused on this local neighborhood information, these techniques may not be suitable for synthetic image generation [8]. Recently, Generative Adversarial Networks (GANs) have gained attention as an oversampling and data augmentation technique for high-dimensional data, as research has demonstrated that synthetic GAN images are effective as additional minority samples in order to improve the resulting classification performance. The vanilla GAN [5] consists of two neural networks with contradicting objectives: a generator and a discriminator. GANs are able to model and imitate the real data distribution in an antagonistic manner. Douzas et al. [4] successfully compared the performance of GANs with other oversampling techniques for binary class data on 71 datasets and found that GANs outperform the other methods. However, generating the under-represented class with exclusively these minority samples might be difficult, as there might not be sufficient minority data to train a GAN. A conditional GAN, which is able to condition the image generation on a specific class, would be more suitable for this oversampling task as it makes use of all data—positives and negatives. In this way, the GAN can learn from a larger feature space to generate samples for the minority class [10]. Moreover, Deep Convolutional GANs (DGGANs) [12] are an extension of the vanilla GAN, where the fully connected blocks are replaced by convolutional blocks. Another improvement in the GAN architecture is the Wasserstein GAN (WGAN) [2], which uses another loss function instead of the binary cross-entropy loss. With the addition of gradient penalty (WGAN-GP) to stabilize learning, it diminishes the vanishing gradient problem the original GAN has [6]. Various researchers have tried to investigate combinations of different models to tackle the imbalance problem. Shamsolmoali et al. [13] have integrated Capsule Neural Networks and GANs to generate the minority class. A more end-to-end framework is proposed by Mullick et al. [11], which came up with a framework consisting of a discriminator, generator and classifier to address class imbalance. The generator learns to generate the minority class which are misclassified, whereas the classifier learns to classify samples as minority or majority class. However, the above mentioned works are highly focused on solving the class imbalance in empirical datasets, such as MNIST and ImageNet, while we make use of a real-world remote sensing dataset, which is created by Statistics

Netherlands for analysis of sustainability indicators in The Netherlands. To our knowledge, oversampling the minority class in an imbalanced case with GANs has not been performed before in the field of solar panel classification. Although GANs have been used to address the lack of annotated remote sensing data (e.g. MARTA GAN [7], SiftingGAN [9]), these remote sensing GAN frameworks are not aimed at solar panel image generation and classification. Also, they are not designed to generate data of the specific under-represented class in case of class imbalance, but rather add more labeled data from every class when the dataset is small. Therefore, we propose a performance comparison of several new GAN frameworks to oversample the under-represented class in aerial solar panel datasets, such that a better classification performance on another geographical location is attained. Instead of directly making use of the positive class as input for the conditional GAN, we subdivide the classes based on their encoded feature information. This division is retrieved by clustering on feature embeddings extracted by a fine-tuned VGG16 network, and these clusters are corroborated by visualizing their samples in a lower dimension. Accordingly, we are able to further ensure heterogeneity of image generation, provide a better feature representation and decrease the risk that the generator only generates one single mode. To our knowledge, tackling the class imbalance with GANs conditioned on this type of auxiliary information has not been performed before in the field of remote sensing and/or solar panel data. Also, we avoid constructing architectures which are overly complex, such that relative computational efficiency is assured, making the proposed models realistically applicable in real-world challenges.

2 Methods

2.1 Datasets

The *source* dataset covers aerial images around Heerlen area (The Netherlands) and has a resolution of 200×200 pixels. It contains 23 847 labeled images, with 4672 positives (20%) and 19 175 negatives (80%). The *target* set covers aerial images from the South of province Limburg (The Netherlands), consisting of 39 506 labeled images, with 36 693 negatives (93%) and 2813 positives (7%). For this imbalanced ratio, a VGG11-based classification model fine-tuned on the source set gives an accuracy, precision and recall of 87.19%, 94.81% respectively 57.10% on the Zuid-Limburg target set. Even though 42.90% of the positives are misclassified as negative, the target accuracy is still high due to only consisting of 7% positives. The model does not have difficulty with classifying the negatives, derived from the high precision value, appearing to be highly biased towards the negative class, while at the same time being overly prudent with classifying samples as positive. Hence, for the comparison between classification performances, we will focus on improving the recall on the Zuid-Limburg target set as our performance metric. For computational efficiency purposes, a sample of 10 000 images is taken from the source dataset to cluster the images and train the GANs. This new set contains all the 4672 positives (47%) and 5328 (53%) negatives. Since we will not generate negatives with the GAN, this omission of negatives is not significant. In contrary, when training the classifier to evaluate the GANs performance, we use the complete source dataset of 23 847 labeled images.

Preprocessing: The research procedure consists of three parts: (1) a preprocessing step where the data is clustered based on its class and features, followed by (2) the generation of synthetic images by means of different types of GANs, and (3) an evaluation of the classification performance after adding the generated data. This is summarized in Figure 1. First, a VGG16 network, fine-tuned on the Heerlen source samples, is used as a feature extractor. The output of the last flatten layer is used, providing latent vectors of 12800 dimensions. These embeddings are the input for clustering methods, which divide the samples into multiple groups. This information is used as the conditional input for the GAN, which generates the fakes images.

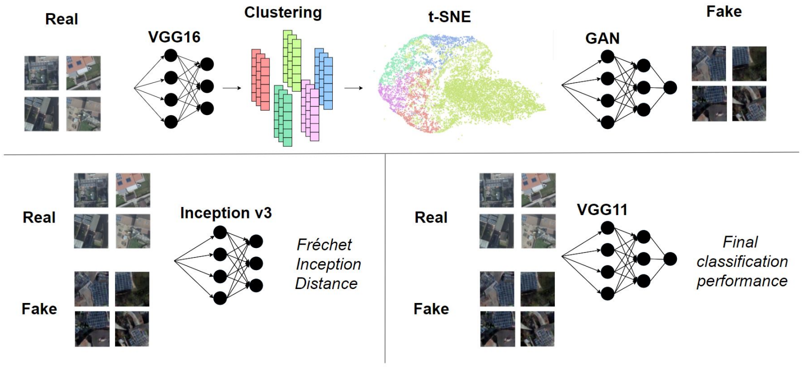

Fig. 1: Experimental design of research procedure

Clustering: The clustering methods are performed on the feature embeddings generated with the fine-tuned VGG16 network. These resulting clusters are provided as auxiliary information for the GANs. The clusterings are computed with KMeans and hierarchical clustering, with Euclidean distance and cosine similarity as distance measures. For the evaluation, The Silhouette score and Calinski-Harabasz (CH) score are used, as these metrics do not require access to the ground truth labels. The Silhouette score is bounded between -1 and $+1$, and a score of -1 implies poor clustering, while a score of $+1$ is optimal. The CH score is not bounded and is desired to be as high as possible.

2.2 Generative Adversarial Networks

Generative Adversarial Network (GAN) is a deep learning based technique that consists of two antagonistic neural networks which have opposite objectives.

Eventually, the goal of the GAN is to generate images which are indistinguishable from the real images, as the fake distribution gradually simulates the real distribution. In the original GAN, the discriminator is a classifier which determines whether a given image originated from the real or generative distribution, by evaluating the conditional probability of the class (real/fake) Y, given the features X. In contrary, the generator aims to produce samples which are close to the real data by modeling its feature distribution, evaluating the probability of the features/data X. It effectuates this imitation by learning a mapping between a latent input vector and the space of the data. Since the discriminator makes a distinction between two classes, the binary cross-entropy (BCE) cost function is used to determine the GAN loss.

An improvement of the vanilla GAN is the Deep Convolutional GAN [12], which performs downsampling with the convolutional stride and upsampling with transposed convolutions, rather than using fully connected layers. Another advancement is the addition of an auxiliary class condition to the input of the GAN, which adds an extra condition to the input of both the generator and discriminator. For the generator, the noise vector is concatenated with a one-hot vector of the class Y. The discriminator receives this auxiliary information by an addition of the one-hot matrices to its input images.

Wasserstein GAN: One of the main disadvantages of the BCE loss is that it approaches zero when the discriminator outputs probabilities close to 0 or 1, which leads to vanishing gradients. This could induce mode collapse, which implies that the generator is stuck at generating one type of sample (mode) which has shown to trick the discriminator, as it has stopped learning. This problem is even more severe in unconditional GANs, since the generator is not compelled to generate multiple classes. To combat this inconvenience, Wasserstein GANs are introduced by Arjovsky, Chintala and Bottou [2] [1]. The WGAN considers a loss function based on the Earth Mover's Distance (EMD), also called Wasserstein-1 distance, which computes the minimum effort required to transform the real distribution to the generative distribution. The main benefit of the EMD compared to BCE loss is that its output values are unbounded, which diminishes its susceptibility to vanishing gradient problems. In this setting, the discriminator is known as a 'critic', as it evaluates a distance rather than probabilities of classes. However, the Wasserstein GAN has the stability requirement that the critic's gradient norm is at most 1 for every possible input, which is controlled by 1-Lipschitz continuity. There are multiple ways to enforce 1-L continuity on the Wasserstein GAN. The first way is weight clipping [2]: imposing the critic's weights to be between bounds, meaning that the weight values which are lower or higher than these bounds are 'clipped' to these bounds. However, the main risk of this method is that it inhibits the learning of the critic. Another method is adding a gradient penalty, a form of regularization of the critic's gradient introduced by Gulrajani et al. [6]. Using these considerations, a new Wasserstein loss function can be derived, by incorporating gradient penalty and a penalty weight λ:

$$\min_g \max_c \mathbb{E}(c(x)) - \mathbb{E}(c(g(z))) + \lambda \mathbb{E}(\|\nabla c(\hat{x})\|_2 - 1)^2 \tag{1}$$

Self-Attention GAN: Although Deep Convolutional GANs are good in generating images with geometric structures, DCGANs sometimes fail to generate the complete object accurately. This is due to the convolutional filters in the DCGAN, having receptive fields which might not be sufficiently large to discover non-local structures in the image. Therefore, the concept of self-attention applied to GANs is introduced by Zhang et al [14], which improves the evaluation of which part of the feature map should receive more attention. This is particularly interesting in the field of solar panel recognition, as the solar panels cover a small fraction of the image. The concept of self-attention uses three representation matrices: query (Q), key (K) and value (V) matrices. Conceptually, the query matrix entails the representation of every position related to itself; the key matrix contains the representation of every position with respect to other positions; the value gives a weight for the attention at every position. The importance of two specific positions relative to each other is computed by the dot product of the query (Q) and key (K) matrices, which is called dot product attention. This dot product is converted to a probability distribution by means of a softmax. The computation and all the other details of Self-Attention GAN are described in [14].

3 Evaluation

Since we are generating images, we can subjectively observe how realistic the samples are. For the evaluation of GANs, two characteristics of the images are important: fidelity and diversity. The fidelity is defined as the quality (e.g. blurriness or realism) of the generated images. The diversity refers to the variety/heterogeneity of the images. However, we will also quantitatively evaluate the images by means of the Fréchet Inception Distance and classification performance increase. *The Fréchet Inception Distance (FID)* is an evaluation metric which computes a high-level feature distance between the real and fake images and is considered as a quantitative indicator of diversity and fidelity. The ImageNet pre-trained Inception-v3 network is widely used as the state-of-the-art feature extractor for FID evaluation. The output of the model is defined by the last pooling layer, which gives an encoded representation of the features in the image of 2048 dimensions. In our experiments, we used a subsample of 512 feature vectors for both real and fake images to compute the FID scores.

A second quantitative metric is assessing whether adding synthetic positives to the source dataset leads to a *classification performance increase in the target set.* For training, we split the source dataset into a train, validation and test set in the ratio (70:20:10). It is made sure that the percentage of positives is 20% in all datasets, analogous to the original source dataset. The architecture of this model is based on VGG11, where the last 4 convolutional blocks are retrained, while freezing the first 4 convolutional blocks. After adding synthetic positives, the classifier with the same hyperparameter settings is trained on this new dataset, and its performance metrics are compared with the baseline model. For every

Table 1: Evaluation metrics clustering

dataset	**method**	**linkage**	**distance**	**silhouette score**	**calinski-harabasz score**
source set	kmeans		euclidean	**0.51**	**1675.70**
	hierarchical	ward	euclidean	0.43	1207.40

experiment, we run the classification model three times and report the average performance.

Influencing the generation: (a) *Subset of clusters* Due to the usage of the clusters, we could have more control over the generation of positives by selecting a subset of clusters we would like to generate. (b) *Truncation trick* The generation can also be influenced by adjusting the noise vector. The values within this vector are sampled from a Gaussian distribution, which implies that values closer to zero occur more frequently in the generator's input during GAN training. These values will generally also lead to higher quality images, as the generator has seen these noise values more regularly during training. The downside is that these images will be less diverse. With this trick, we can influence the generation of images by truncating the values in the noise vectors provided to the trained GAN.

4 Results and Discussion

4.1 Data Preprocessing

In Table 1, the performance metrics for both KMeans and hierarchical clustering are shown. KMeans is evaluated with the Euclidean distance and the optimum number of clusters is $k = 5$, value based on experimentation. As hierarchical clustering has several parameters to tune (e.g. number of clusters, affinity, linkage type), only the best experimental combinations are shown in the table, while we explored all combinations. We can see that KMeans has the best Silhouette and CH score of 0.51 respectively 1675.70. Therefore, as a next step we evaluated the class distribution within the KMeans clusters. Four clusters (0, 2, 3, and 4) consist of more than 99% positives, while cluster 1 contains roughly all the negatives. However, 25% of the samples within cluster 1 are positive instead of negative. Since we will use the positive clusters as conditional input for the GANs to generate new positives, this implies that the positives hidden in the negative cluster will not be included in generation. Further exploration shows that these positives in cluster 1 are the samples that are mostly misclassified by the classification model. Hence, it might be interesting to include these positives in GAN training, since adding more of these type of generated positives to the dataset might enable the classifier to improve on these samples. Therefore, a possible solution is to split the negatives from this cluster into cluster 1 (the split

Table 2: Performance comparison of vanilla GAN

model	input	output	FID	accuracy	recall	precision
baseline				0.8719	0.5710	0.9481
A	2 classes	1-dim	251.6	0.8728	0.5918	0.9237
B	5 clusters (original)	1-dim	220.6	0.8731	0.5971	0.9182
C	5 clusters (positives omitted)	1-dim	292.6	0.8736	0.5972	0.9227
D	6 clusters	1-dim	**198.85**	0.8747	**0.6001**	0.9208
E	6 clusters	8-dim	**185.58**	0.8747	**0.6037**	0.9180

positives) and a newly created cluster 5 (the negatives). We will compare the GAN performances achieved by considering four scenarios: training the GAN with the 2 original classes as input, the original 5 clusters, the 6 clusters and the 5 clusters where the positives are omitted from the 'negative' cluster.

4.2 Generation and Evaluation

We will test three different types of GANs: the conditional vanilla GAN, the conditional Wasserstein GAN and the conditional Self-Attention GAN. We will also condition the GANs on two types of auxiliary inputs: the original binary class and the generated clusters. For all training, a GPU (Tesla-V100-SXM2) of 32 GB and CUDA version 11.1 is used. For all models, the images are resized from 200×200×3 to 96×96×3, as experimentation showed that resizing the images to this resolution provided more realistic outputs as well as being more computationally efficient. We noticed that adding too many synthetic samples to the source set would mean that there are proportionally a lot more synthetic positives than real positives in the dataset, which could make the classification model focus more on the fake positives. Experimentation with several proportions of added synthetic positives showed that the percentage of positives in the source set increasing from 19.59% to 35.13% consistently gave the best results. This implies the new source set consists of 45.0% real positives and 55.0% fake positives out of all positives. Therefore, all experiments below make use of these proportions to enable an analogous and equitable comparison of results. As already mentioned, for every experiment we run the classification model three times and report the average performance.

Vanilla GAN: The optimal architecture for the vanilla conditional GAN with binary cross-entropy loss has a generator with 6 convolutional blocks and a discriminator with 4 convolutional blocks. The batch size is 256, noise dimension 64, learning rate 0.0002 and we used the Adam optimizer. It makes use of normal initialization for both the convolutional and batch normalization layers. The model was trained for 600-2000 epochs. **Comparing the generator input:** We test four different GAN inputs: the original 2 classes (A), the 5 original clusters

(B), 5 clusters with the positives in the 'negative' cluster completely omitted from the dataset (C), and 6 clusters with the positives in the 'negative' cluster separated (D). The results are shown in Table 2. The results show that the 6 clusters with the split positives gives the best FID (198.85) and recall increase (2.91%) compared to the baseline model. Figure 2 shows a random set of generated images, in which every row represents a cluster and the last row represents the negative cluster. The models which make use of the clusters as input (B and D) have a higher/worse FID value as well as higher observed fidelity and diversity than the model A, which uses the classes as input. However, completely omitting the positives from the negative cluster (model C) increases the FID, which could be due to not having sufficient data. **Comparing discriminator output:** The results show that the 8-dimensional output (model E) provides a better FID score (185.58) than the scalar prediction of model D (198.85). Additionally, model E gives the highest increase in recall, namely 3.27%. A grid of generated images is shown in Figure 3.

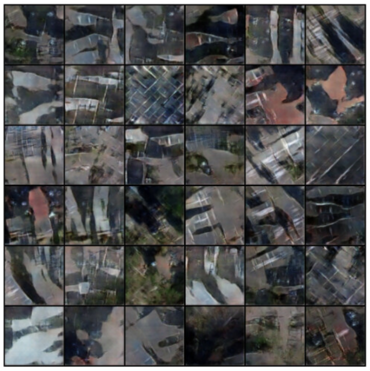

Fig. 2: Generated images Vanilla GAN (model D)

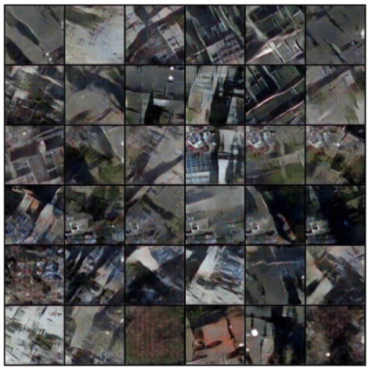

Fig. 3: Generated images Vanilla GAN (model E)

Wasserstein GAN: The second type of GAN we evaluate is the conditional Deep Convolutional WGAN. In order to ensure 1-Lipschitz continuity of the critic, we can impose gradient penalty or weight clipping. We evaluated several variations of the WGAN: WGAN without any 1-L enforcement, WGAN with gradient penalty and WGAN with gradient clipping. The optimal generator and critic architecture both consist of 5 blocks. The models are run for a varying number of epochs between 600-2000, with noise dimension 64, batch size 256 and learning rate 0.0002. We experiment with both optimizers Adam and RMSprop, as the authors of the vanilla WGAN [2] state that a momentum-based optimizer such as Adam may destabilize training, while the authors of the WGAN-GP [6] found that Adam outperforms RMSprop. Moreover, updating the critic multiple times before updating the generator is introduced for the WGAN-GP, as we want to avoid that the generator overpowers the critic due to not having a penalty. **WGAN with gradient penalty:** In Table 3, the results for some of the WGAN models with gradient penalty are shown. Model H is the WGAN without any penalty or 1-L enforcement. It generates good images until the 250th epoch, after which the model destabilizes and fails to generate realistic results. Hence, for its FID calculation, the model at the 250th epoch is chosen, which shows a relatively high diversity and medium realistic results and the highest recall increase of 3.52%. The other models (I until L) make use of gradient penalty. The WGAN with penalty weight 10, 2 updates for the critic before updating the generator and optimizer RMSprop (model I) shows the best FID of 193.51 and highest recall increase of 2.90%. A sample of generated images from model I is shown in Figure 4. The table also shows that increasing the critic repeats from 2 to 5 in model L worsens the performance drastically, as the critic might overpower the generator.

Table 3: Performance comparison of WGAN

model	**penalty weight**	**critic repeats**	**optim-izer**	**clip value**	**FID**	**accuracy**	**recall**	**precision**
baseline						0.8719	0.5710	0.9481
H	none	1	adam		232.6	0.8761	**0.6062**	0.9219
I	10	2	RMSprop		**193.51**	0.8741	0.6000	0.9201
J	15	1	RMSprop		200.88	0.8735	0.5995	0.9175
K	10	1	adam		251.3	0.8710	0.5703	0.9456
L	10	5	adam		332.1	0.8703	0.5698	0.9442
M			adam	0.01	**181.82**	0.8864	**0.6324**	0.9269
P			RMSprop	0.02	207.04	0.8735	0.6020	0.9240
Q			adam	0.03	210.06	0.8792	0.6144	0.9211

WGAN with weight clipping: For the WGAN with weight clipping, various clip values are examined. The results of some models with various parameters are shown in Table 3. Model M, P and Q show the results with clip value 0.01, 0.02 respectively 0.03. Model M showed the best FID value of 181.82. The model increases the target recall with 6.14%, while model Q with clip value 0.03 increases the metric with 4.34%.

Self-Attention GAN: The Self-Attention (SA) GAN paper states that spectral normalization in both the generator and discriminator improves the performance of the SA-GAN [14]. However, analogous to the results for the WGAN, spectral normalization destabilized the GAN. Moreover, both WGAN with weight clipping and WGAN with gradient penalty did not show stable results. However, the SA-GAN with BCE loss did showed more stable results. In Table 4, the results for the two best models are shown. Both models use the BCE loss. In model R, both the generator and discriminator consist of 5 convolutional blocks, while the attention block is placed after the second block. In model S, an extra attention

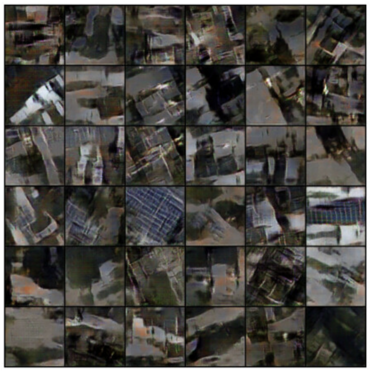

Fig. 4: Generated images WGAN with GP (model I)

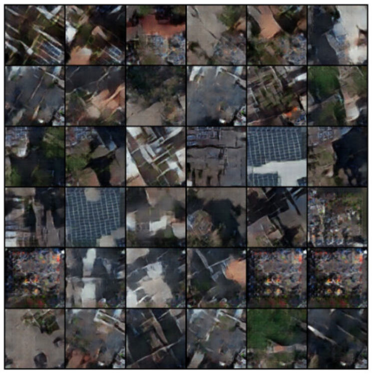

Fig. 5: Generated images SA-GAN (model R)

Table 4: SA-GAN comparison

model	**attention blocks**	**loss**	**FID**	**accuracy**	**recall**	**precision**
baseline				0.8719	0.5710	0.9481
R	1	BCE	**209.68**	0.8750	**0.6022**	0.9213
S	2	BCE	240.97	0.8739	0.5964	0.9232

block is placed after the second convolutional block. Table 4 shows that the model with one attention block delivered the highest FID of 209.68. Also, the recall rate improved with 3.12%. A sample of generated images is shown in Figure 5. The sample shows that the solar panels are visible, although it contains many images which seem more unrealistic and nonsensical than previous models.

5 Conclusion

This research was focused on finding a suitable oversampling method to address the class imbalance in an aerial image dataset, in order to improve the classification performance on a target set covering another geographical location. Originally, the model is biased towards the majority group, being overly prudent with classifying a sample as positive, resulting in a low recall rate of 57.10% in the target set. For the generation of synthetic minority samples, we made use of three different GAN architectures: the conditional vanilla GAN, conditional Wasserstein GAN with different 1-Lipschitz continuity enforcements and the Self-Attention GAN. Instead of directly making use of the positive class as condition input for the GAN, we subdivided the class based on clusters on fine-tuned VGG16-based feature embeddings. Accordingly, we were able to further ensure heterogeneity of image generation and guarantee a feature representation based on multiple modes. While all models led to an increase in recall for the target set, the Wasserstein

GAN with weight clipping provided the largest recall increase of 6.14%, while also achieving the best FID of 181.82. Nonetheless, the experiments have shown that these models are highly sensitive to hyper-parameter and architecture settings while also being computationally complex, making the process expensive. Moreover, one of the main challenges is the relative size of the solar panel in the image, as it covers a small part of the image. For future work we propose investigating whether these issues could be addressed, by further improving the GAN architecture with state-of-the-art techniques, such as injecting more random noise with Adaptive Instance Normalization, substituting the noise vector by a Noise Mapping Network, adding skip connections and/or progressively growing to speed up the training time.

References

1. Arjovsky, M., Bottou, L.: Towards principled methods for training generative adversarial networks. arXiv preprint arXiv:1701.04862 (2017)
2. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein gan. arXiv preprint arXiv:1701.07875 (2017)
3. Buda, M., Maki, A., Mazurowski, M.A.: A systematic study of the class imbalance problem in convolutional neural networks. Neural Networks **106**, 249–259 (2018)
4. Douzas, G., Bacao, F.: Effective data generation for imbalanced learning using conditional generative adversarial networks. Expert Systems with applications **91**, 464–471 (2018)
5. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in neural information processing systems. pp. 2672–2680 (2014)
6. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of wasserstein gans. In: Advances in neural information processing systems. pp. 5767–5777 (2017)
7. Lin, D., Fu, K., Wang, Y., Xu, G., Sun, X.: Marta gans: Unsupervised representation learning for remote sensing image classification. IEEE Geoscience and Remote Sensing Letters **14**(11), 2092–2096 (2017)
8. Lusa, L., et al.: Evaluation of smote for high-dimensional class-imbalanced microarray data. In: 11th Int. Conf. on Machine Learning and Applications (2012)
9. Ma, D., Tang, P., Zhao, L.: Siftinggan: Generating and sifting labeled samples to improve the remote sensing image scene classification baseline in vitro. IEEE Geoscience and Remote Sensing Letters **16**(7), 1046–1050 (2019)
10. Mariani, G., Scheidegger, F., Istrate, R., Bekas, C., Malossi, C.: Bagan: Data augmentation with balancing gan. arXiv preprint arXiv:1803.09655 (2018)
11. Mullick, S.S., Datta, S., Das, S.: Generative adversarial minority oversampling. In: Proc. of the IEEE/CVF Int. Conf. on Computer Vision. pp. 1695–1704 (2019)
12. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv:1511.06434 (2015)
13. Shamsolmoali, P., Zareapoor, M., Shen, L., Sadka, A.H., Yang, J.: Imbalanced data learning by minority class augmentation using capsule adversarial networks. Neurocomputing (2020)
14. Zhang, H., Goodfellow, I., Metaxas, D., Odena, A.: Self-attention generative adversarial networks. In: International conference on machine learning. pp. 7354–7363. PMLR (2019)