# Learning with a generative adversarial network from a positive unlabeled dataset for image classification

F Chiaroni, Mohamed-Cherif Rahal, Nicolas Hueber, Frédéric Dufaux

# LEARNING WITH A GENERATIVE ADVERSARIAL NETWORK FROM A POSITIVE UNLABELED DATASET FOR IMAGE CLASSIFICATION

*F. Chiaroni*[⋆‡] *M-C. Rahal*[⋆] *N. Hueber*[†] *F. Dufaux*[‡]

[⋆] VEDECOM Institute, Department of delegated driving (VEH08), Perception team,
{florent.chiaroni, mohamed.rahal}@vedecom.fr
[†] French-German Research Institute of Saint-Louis (ISL), ELSI team, nicolas.hueber@isl.eu
[‡] L2S-CNRS-CentraleSupelec-Univ Paris-Sud,
{florent.chiaroni, frederic.dufaux}@l2s.centralesupelec.fr

**ABSTRACT**

In this paper, we propose a new approach which addresses the Positive Unlabeled learning challenge for image classification. Its functioning is based on GAN abilities in order to generate fake images samples whose distribution gets closer to negative samples distribution included in the unlabeled dataset available, while being different to the distribution of the unlabeled positive samples. Then we train a CNN classifier with the positive samples and the fake generated samples, as it would be done with a classic Positive Negative dataset. The tests performed on three different image classification datasets show that the system is stable up to an acceptable fraction of positive samples present in the unlabeled dataset. Although very different, this method outperforms the state of the art PU learning on the RGB dataset CIFAR-10.

***Index Terms***— Deep Learning, Image classification, Positive Unlabeled Learning, Representation Learning, Generative Models

## 1. INTRODUCTION

Deep learning methods using convolutional kernel filters have demonstrated good prediction performances in the field of computer vision and especially for the task of image classification. To achieve such performance, large fully labeled datasets are required. Nowadays, multiple datasets can be merged to increase the generalization capacity of learning model as described in YOLO9000 [1].

Nevertheless, to mitigate this need of large labeled datasets, an idea is to focus mainly on data of interest. This is the case in One-Class Classification methods (OCC) [2], novelty detection [3] where they only use during the training the samples of the class of interest; the positive class. To our knowledge, OCC methods have a limited performance when applied to large data tensors such as images. Besides, it is often easy to acquire unlabeled samples that may contain relevant information especially about the counter-examples of the class of interest. In this way, we address a Positive Unlabeled learning problem. It turns out that PU learning methods have been applied recently to image data type such as the Rank Pruning method (RP) [4]. RP method consists in consecutively carrying out several trainings of the classifier on a noisy labeled dataset, by removing the least relevant samples after each training stages. Furthermore, according to [5], PU learning methods become competitive when the number of unlabeled samples in the dataset considerably increases. This is an advantage when we can easily get unlabeled data.

In addition, the generative adversarial networks (GANs) have drawn our attention because of their ability to generate fake samples $x_F$ that have a distribution $p_G(x_F)$ that tends towards the distribution $p_{data}(x_R)$ of the real samples $x_R$ used during its training. The original GAN [6] contains a generative model $G$ and a discriminative model $D$. Both models have a multilayer perceptron structure. A noise vector $z$ with a distribution $p_z(z)$, composed of continuous random variables, is placed at the input of $G$. $D$ is trained to distinguish real samples from fake samples generated by $G$, while the latter is trained to produce fake samples that seem as real as possible. This adversarial training consists in using a minimax function value $V(G, D)$:

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

When $D$ can no longer distinguish real samples from fake samples, we have the following property for its scalar predicted output $y_D$:

$$p_G(x_F) \xrightarrow[y_D \to 0.5]{} p_{data}(x_R). \quad (2)$$

Other variants of GAN have emerged such as the DCGAN [7], which adapts its structure to image processing by incorporating convolutional layers. The Wasserstein GAN (WGAN) [8] integrates the distance $Earth-Mover$ $(EM)$ into its cost function in order to rectify the instability problem of these early versions of GANs.

Because of their ability to learn relevant semantic representations, we decided to exploit their advantages for a PU learning application. The approach [9] also appeared to answer the same problematic by the use of a GAN learning model. But their study stops at the functional description of their model. Moreover, the latter requires to train simultaneously five neural networks against only two in our method. Here, our proposed approach that we called Positive-GAN (hereafter "PGAN") has been tested on three different datasets and whose results are very promising in terms of prediction performance and robustness. It outperforms the state of the art on the most challenging image dataset that we tested.

The paper is organized as follows. In the next section we present the method. The experimentations and results are presented in the third section. In the end, a conclusion is drawn on our approach and future research directions are suggested.

## 2. POSITIVE-GAN LEARNING METHOD

In this section, we describe our PU learning framework as generically as possible and focus the description on the training method. The Positive-GAN learning method (PGAN) consists in substituting the absence of labeled negative samples $x_N$ with fake samples $x_F$ generated by our GAN, and so that whose distribution is as close as possible to that of $x_N$, while being different from that of positive samples $x_P$. Fig.1 illustrates the structure of the system.

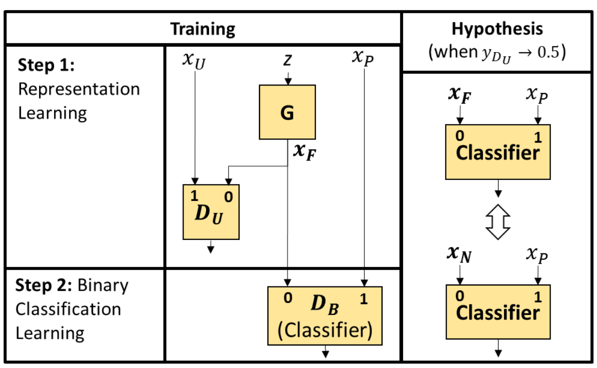



**Fig. 1**. Proposed Positive Unlabeled system: Positive-GAN Learning model.

During the Step 1, the GAN is trained with the unlabeled samples $x_U$ from the PU training dataset that contains a fraction $\pi$ of positive samples and a fraction $1-\pi$ of negative samples $x_N$. The Positive Unlabeled framework includes three convolutional neural network models with different roles:

- The discriminative model $D_U$ is trained to distinguish real unlabeled samples $x_U$ from fake generated unlabeled samples $x_F$, with $y_{D_U} \in (0,1)$ its scalar output prediction.
- The generative model $G$ takes in input a noise vector $z$ of continuous random variables with a uniform distribution in this case and outputs, in the same format as $x_U$, the fake image samples $x_F(\Leftrightarrow G(z))$ which can be either positive $x_{FP}$ or negative $x_{FN}$. $G$ is trained in an adversarial way with $D_U$ in order to generate fake samples such that their distribution $p(x_F)$ tends towards $p(x_U)$. At the same time it gets away from positive labeled samples distribution $p(x_P)$ as explained below.
- In Step 2, once the GAN training is completed, the convolutional classifier $D_B$, designed for binary image classification task, is trained to distinguish the real positive samples $x_P$ from fake samples $x_F$.

We remember that the untagged dataset is composed of a fraction $\pi$ of positive samples $x_P$ and a fraction $1-\pi$ of negative samples $x_N$. So if the GAN is correctly trained on the unlabeled samples $x_U$, we can deduce that:

$$p(x_F) \xrightarrow[y_{D_U}\to 0.5]{} p(x_U) \tag{3}$$

$$\Leftrightarrow p(x_{FP}), p(x_{FN}) \xrightarrow[y_{D_U}\to 0.5]{} p(x_P), p(x_N). \tag{4}$$

It is also known that a GAN is not perfect in its operation when it is applied to high dimensional data, therefore:

$$p(x_{FP}) \neq p(x_P), \text{ and } p(x_{FN}) \neq p(x_N). \tag{5}$$

Thus it is possible to estimate the non-zero distance $d$ computed into the cost function of $D_B$ such that:

$$d(p(x_P), p(x_F)) \;\Leftrightarrow \begin{cases} d(p(x_P), p(x_{FP})) \\ d(p(x_P), p(x_{FN})). \end{cases} \tag{6}$$

But, the distance $d(p(x_P), p(x_{FP}))$ will not be exploited for the final application where we treat only real samples with the classifier. This means that when $p(x_{FN}) \xrightarrow[y_{D_U}\to 0.5]{} p(x_N)$, we get the equivalence:

$$d(p(x_P), p(x_{FN})) \Leftrightarrow d(p(x_P), p(x_N)). \tag{7}$$

We are thus able to calculate the distance that interests us. By transposing this reasoning in the PU framework, this amounts to asserting the following equivalences at the output loss function $L_{D_B}$ of the classifier $D_B$ when $y_{D_U} \to 0.5$:

$$\begin{aligned} L_{D_B} &= \mathbb{E}_{x_P\sim p(x_P)}[log D_B(x_P)] \\ &+\mathbb{E}_{z\sim p_z(z)}[log(1-D_B(G(z)))] \end{aligned} \tag{8}$$

$$\begin{aligned} \Leftrightarrow L_{D_B} &= \mathbb{E}_{x_P\sim p(x_P)}[log D_B(x_P)] \\ &+\mathbb{E}_{x_N\sim p(x_N)}[log(1-D_B(x_N))]. \end{aligned} \tag{9}$$

Thus, from the assumptions made above we can assume that the PGAN method is getting closer to a Positive Negative training while moving away from a Positive Unlabeled training despite the fact that the training dataset we have contains only positive labeled samples and unlabeled samples. However, two risks can occur with this method:

- If the untagged samples $x_U$ contain mostly positive samples $x_P$, then it is possible that $G$ no longer generates enough fake samples similar to the samples $x_N$.
- If $G$ generates false samples with a distribution equal to that of the real samples, unlike in the inequality 5, then PGAN would become equivalent in terms of performance to a classical Positive Unlabeled training. But when the dimensionality of images to be processed is large, this risk disappears.

## 3. EXPERIMENTS

### 3.1. Settings

Experiments have been realized on the three datasets MNIST [10], Fashion-MNIST [11] and CIFAR-10 [12]. We have compared our approach to RP [4], which is the best PU learning method to the best of our knowledge. Author's implementation is available [1]. We also report the performance of the classifier trained on the entire Positive Negative initial training dataset, and we call naturally this method PN. We also compare our method to a training named PU which is equivalent to PN but with the PU dataset.

For these experiments, PN, PU, PGAN (ours) and RP methods are tested with exactly the same CNN classifier in order to stay impartial. The classifier has the same structure as in [2] to remain generic. It contains two convolutional layers, two corresponding maxpooling steps, and then two consecutive fully connected layers. Activation function after each layer is ReLU except the last one where softmax is applied. We only changed its last layer from 10 to 2 neurons to adapt it for binary classification. The classifier is trained on 20 epochs iterations. For the CIFAR-10 dataset images 32*32*3, the input and output tensors of the two convolutional layers are adapted and the depth of the first convolutional kernel filters is 3 to correspond to the 3 channels of the RGB images. But the number of kernel filters and their remaining height*width remain unchanged.

Because of the WGAN [8] abilities, we use its training method for these experiments. The training duration for the generative model depends on the dataset complexity: 10 epochs for MNIST, 20 for Fashion-MNIST, and 100 for CIFAR-10. For the latter, we do the same modifications in the structures of $D_U$ and $G$ as explained before for the classifier.

Regarding the PU training dataset, $\rho$ corresponds to the fraction of positive samples $P$ from the total of positive samples in the initial training dataset which contains $n_P$ samples. These collected samples are then introduced into the $U_{train}$ unlabeled dataset, which initially contains only negative samples $N$ whose total number is $n_N$. $\pi$ is the fraction of positive samples $P$ present in the unlabeled training dataset $U_{train}$. $U_{train}$ then contains both negative and positive samples according to the $\rho$ and $\pi$ parameters. We establish that if $\pi \in [\frac{1}{\frac{n_N}{\rho \times n_P}+1}, 1)$, and $\rho \in (0, 1)$, then we can obtain consecutively the two following training sets:

$$P_{train} = \{(1-\rho)n_P\ P\ ;\ 0\ N\}, \tag{10}$$

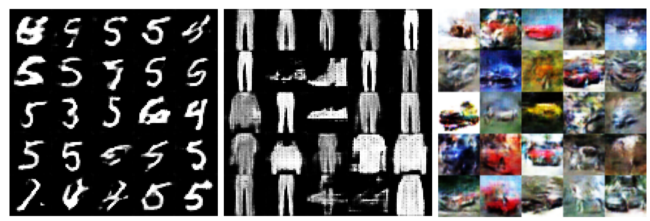
with $P_{train}$ the set of positive training samples, and

$$U_{train} = \{\rho \times n_P\ P\ ;\ \frac{1-\pi}{\pi}\rho \times n_P\ N\}. \tag{11}$$

The results presented below are all performed with $\rho = 0.5$ and for several values of $\pi$.

### 3.2. Results

In Fig. 2, we present some of the fake images generated, respectively from MNIST, Fashion-MNIST and CIFAR-10. We can notice that the images generated by $G$ seem visually real, which indicates from a qualitative point of view the proper functioning of the generative model. In order to get such a rendering, the more complex and large the images are, the more the generative model requires a large number of training epochs.



(a) (b) (c)

**Fig. 2**. Images generated by $G$ trained with $\rho = 0.5$ and $\pi = 0.5$ after 10 epoch itterations on MNIST(a), 20 on Fashion-MNIST(b), and 100 on CIFAR-10(c). Respective positive classes are "5", "$trouser$" and "$automobile$".

To compute the F1-Scores, the ArgMax function is applied to the two output neurons of the classifier. Thus, if the index of the first neuron is returned by ArgMax, then the treated sample is considered as negative. Otherwise, the sample is considered as positive. Table 1 shows some of the average F1-Score [3] comparative results.

In Fig. 3, it can be observed that the PN method is a good reference for the MNIST and Fashion-MNIST datasets. We find that the efficiency of the PGAN learning method is equivalent to that of the RP method up to $\pi = 0.5$ on MNIST and $\pi = 0.3$ on Fashion-MNIST. Its efficiency then declines a little bit faster than for RP, while keeping a correct F1-Score. On CIFAR-10 from end to end, the F1-Score is better for PGAN than for RP. Note that the PGAN method

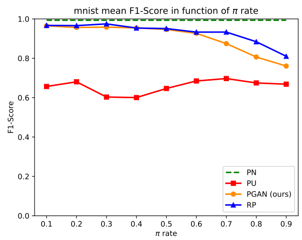
[1] $https://github.com/cgnorthcutt/rankpruning$

[2] $https://github.com/tensorflow/tensorflow/blob/master/tensorflow/examples/tutorials/mnist/mnist\_softmax.py$

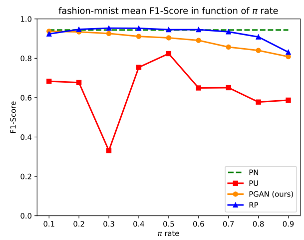
[3] Average F1-Score represents the mean of F1-Score mesured for each class in a given dataset.

**Table 1**. F1-Score averages results comparisons on MNIST, Fashion-MNIST and CIFAR-10 after 20 epochs of the classifier.

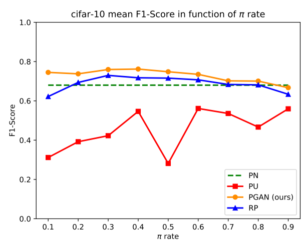
| | PU | PGAN | RP | PU | PGAN | RP | PU | PGAN | RP | PU | PGAN | RP |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Dataset | $\rho = 0.5,\ \pi = 0.1$ | | | $\rho = 0.5,\ \pi = 0.3$ | | | $\rho = 0.5,\ \pi = 0.5$ | | | $\rho = 0.5,\ \pi = 0.7$ | | |
| $AVG_{\text{MNIST}}$ | 0.66 | **0.97** | **0.97** | 0.60 | 0.96 | **0.97** | 0.65 | **0.95** | **0.95** | 0.70 | 0.87 | **0.93** |
| $AVG_{\text{Fashion-MNIST}}$ | 0.68 | **0.94** | 0.92 | 0.33 | 0.93 | **0.95** | 0.82 | 0.90 | **0.95** | 0.65 | 0.86 | **0.94** |
| $AVG_{\text{CIFAR-10}}$ | 0.31 | **0.75** | 0.62 | 0.42 | **0.76** | 0.73 | 0.28 | **0.75** | 0.72 | 0.54 | **0.70** | 0.68 |




(a)

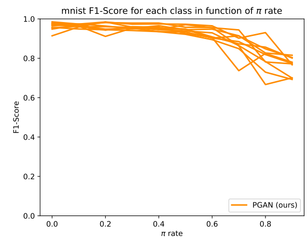



(b)

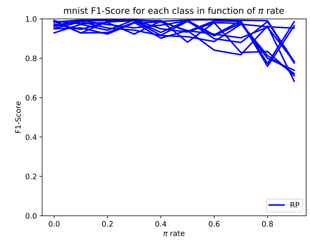



(c)

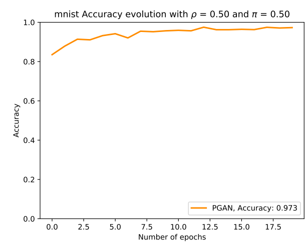
**Fig. 3**. Average F1-Scores after 20 training epoch iterations of the classifier in fonction of the rate $\pi$ that is varied between 0 and 1 with a step of 0.1, for PN (green), PU (red), RP (blue) and PGAN (orange) on MNIST (a), Fashion-MNIST (b) and CIFAR-10 (c).



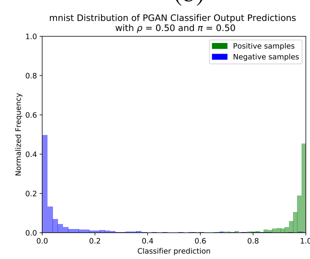

(a)




(b)




(c)




(d)

**Fig. 4**. Robustness analysis on MNIST. F1-Score evolution for each class as a function of $\pi$ for PGAN (a), and for RP [4] (b). (c) shows the Accuracy evolution during the PGAN training with the positive class "5" and $\pi = 0.5$. (d) is the histogram of the output distributions of the classifier at its 20th epoch iteration of (c) for positive (green) and negative (blue) test samples.

also presents better results than the reference PN up to $\pi = 0.8$, which is quite surprising. This is probably because the generated samples represent a larger field of negative sample distributions than the real negative samples present in the initial Positive Negative dataset. Moreover, the PGAN F1-Score is consistently higher than the PU method on the three datasets.

Figure 4 presents the study of the robustness of PGAN approach. Figures 4.a and 4.b show that PGAN method has a comparatively more robust behavior such that we can predict more easily the F1-Score evolution as a function of $\pi$ for each dataset class. Figure 4.c shows us that the classifier stabilizes and converges after about 10 training epochs. To make the histogram in Fig. 4.d, we have retrieved the scalar of the second output neuron of the classifier which corresponds to the predicted probability for a input image of belonging to the positive class. It can be seen that the distribution of the negative and positive test samples processed by the PGAN is of the Gaussian type, which is an interesting characteristic for real applications.

## 4. CONCLUSION

Thereby, we demonstrated that the proposed PU learning approach provides state of the art prediction performances and has a steady behavior on small image datasets up to an acceptable fraction $\pi$ of positive samples in the unlabeled training dataset. In addition, our method outperforms the state of the art on challenging RGB images of CIFAR-10. System optimization can be carried on testing other generative models instead of the WGAN [8], like BEGAN [13], WGAN-GP [14]. Another orientation can be to exploit the $z$ latent space of GANs to perform linear arithmetic operations, as in [15], in order to generate more relevant fake samples.

Considering the promising performances obtained on the CIFAR-10 dataset, a future direction is to extend this method to the analysis of larger images and thus to allow the realization of more complex tasks such as object detection [16], [17], [18] or semantic segmentation [19].

## 5. REFERENCES


[1] Joseph Redmon and Ali Farhadi, "YOLO9000: Better, Faster, Stronger," *arXiv preprint arXiv:1612.08242*, 2016.

[2] Shehroz S. Khan and Michael G. Madden, "One-class classification: taxonomy of study and review of techniques," *The Knowledge Engineering Review*, vol. 29, no. 3, pp. 345–374, 2014.

[3] Marco AF Pimentel, David A. Clifton, Lei Clifton, and Lionel Tarassenko, "A review of novelty detection," *Signal Processing*, vol. 99, pp. 215–249, 2014.

[4] Curtis G. Northcutt, Tailin Wu, and Isaac L. Chuang, "Learning with confident examples: Rank pruning for robust classification with noisy labels," in *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence*. 2017, UAI'17, AUAI Press.

[5] Gang Niu, Marthinus Christoffel du Plessis, Tomoya Sakai, Yao Ma, and Masashi Sugiyama, "Theoretical comparisons of positive-unlabeled learning against positive-negative learning," in *Advances in Neural Information Processing Systems*, 2016, pp. 1199–1207.

[6] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[7] Alec Radford, Luke Metz, and Soumith Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[8] Martin Arjovsky, Soumith Chintala, and Léon Bottou, "Wasserstein generative adversarial networks," in *International Conference on Machine Learning*, 2017, pp. 214–223.

[9] Ming Hou, Qibin Zhao, Chao Li, and Brahim Chaib-draa, "A generative adversarial framework for positive-unlabeled classification," *arXiv preprint arXiv:1711.08054*, 2017.

[10] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[11] Han Xiao, Kashif Rasul, and Roland Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," 2017.

[12] Alex Krizhevsky and Geoffrey Hinton, "Learning multiple layers of features from tiny images," 2009.

[13] David Berthelot, Tom Schumm, and Luke Metz, "Began: Boundary equilibrium generative adversarial networks," *arXiv preprint arXiv:1703.10717*, 2017.

[14] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C. Courville, "Improved training of wasserstein gans," in *Advances in Neural Information Processing Systems*, 2017, pp. 5769–5779.

[15] Piotr Bojanowski, Armand Joulin, David Lopez-Paz, and Arthur Szlam, "Optimizing the latent space of generative networks," *arXiv preprint arXiv:1707.05776*, 2017.

[16] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[17] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg, "SSD: Single shot multibox detector," in *European Conference on Computer Vision*. 2016, pp. 21–37, Springer.

[18] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[19] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.