Improved COVID Detection with GANs

Yacine Mohamed Bouaouni
mohamed.bouaouni@ens-paris-saclay.fr

Fayçal Rekbi
faycal.rekbi@ens-paris-saclay.fr

Abstract

Coronaviruses are a type of viruses that can be detected using chest X-ray images. The large success of deep learning models in medical imaging and their excellent preformances motivate the use of such models to detect the presence of the virus in X-ray images. However, due to the lack of data in the medical field and the fast propagation of the virus, the task of training a neural network and obtain good performance becomes more complex. In our project, we propose to generate chest X-ray images of patients affected by Covid-19 using Auxiliary Classifier Generative Adversarial Network (AC-GAN). We also show that combining the original dataset with synthetic X-ray images generated with the AC-GAN improves the performance of the CNN by more than 3% on the test set. We argue that this approach could lead to improve the detection and diagnosis of viruses in early stages when it's hard to gather sufficient data.

1. Introduction

Coronaviruses are a large family of viruses, which cause illnesses ranging from the common cold (some seasonal viruses are Coronaviruses) to more severe conditions such as MERS or SARS. The virus identified in January 2020 in China is a new Coronavirus. The disease caused by this Coronavirus has been named COVID-19 by the World Health Organisation (WHO). COVID-19 can be diagnosed from a chest radiograph and it's the fastest methods results are achieved within minutes. However, it looks very similar to other viral and bacterial pneumonias on chest radiographs, which makes it difficult to diagnose. There are different rules that can help radiologists to distinguish COVID-19 from other types of infections. The developed model can then be used as a support for doctors in the detection of COVID-19.

1.1. Motivation

One of the biggest challenges in the medical imaging domain is its data quantity limitations (see [9],[11]). A CNN requires a large learning base in order to correctly learn a model. As the epidemic is recent, it is very difficult to collect a large number of radiographic images of people with COVID-19. Moreover, CNNs can easily overfit on small datasets because of the large number of parameters, therefore, the efficiency of generalization is proportional to the size of the labeled data. Usually for small datasets transfer learning is used, however, the substantial differences between natural and medical images may compromise the effectiveness of such knowledge transfer. Thus the need for data augmentation techniques and the main question that can be asked is what data augmentation technique is well adapted to medical images?

1.2. Related works

Data augmentation techniques can be used to improve the generalization performances of the model, but a central question that is often asked in the literature is: Is it better to use basic image manipulations (e.g., geometric transforms) or use deep learning approaches (e.g., GANs) to augment medical images? According to [10], applying a geometric data augmentation must be domain dependent and without a clinical consideration it yields the model to perform poorly on the validation dataset. The second approach of data augmentation using deep learning techniques is becoming more popular (see [12],[7]). It has proven that it can improve the classifiers performances to help detect viruses in X-ray images.

Generative Adversarial Network are becoming more popular in the field of medical imaging for various applications. Forward and Backward GANs are used to generate synthetique images for lung-nodules classification in [13]. Progressively grown GANs are used for medical image synthesis of fundus pictures with premature retinopathic vascular pathology (see [3]). Patch-based GANs are used by [8] to convert brain CT images to the corresponding MRI. Finaly, Auxiliary Classifier GANs are used for generating chest x-ray images (see [12].

1.3. Problem defintion

In our project, we present a method to generate synthetic chest Xray (CXR) images using two types of GANs : (1) Auxiliary Classifier GAN (AC-GAN) and Deep Convolutional GAN (DCGAN). The generated images using these two GANs are compared by visual inspection and quantitative metric (frechet-inception distance) to assess the quality of the generated images compared to the original dataset. The generated data will then be provided to train a classifier to detect covid-19 Xrays and compare its performance with a classifier trained only on the original dataset.

Formally, the problem can be formulated as follows: given a dataset of labeled images $\{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$. Generate N new images $\{g_1, g_2, ..., g_n\}$ using GANs (AC-GAN/DCGAN) such that the distance between the two distributions is minimized :

$$\underset{\theta_g}{argmin} \quad d(\mathbf{P}(\mathbf{x}), \mathbf{P}(\mathbf{g}; \theta_{\mathbf{g}}))$$

Then, these the two sets of images are used to train the classifier to maximize the accuracy metric:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

where TP = True positive; FP = False positive; TN = True negative; FN = False negative

1.4. Dataset.

The dataset consists of chest X-ray and CT images of patients which are positive or suspected of COVID-19 and negative patients. In order to compare our approach with the literature, evaluating the model on the same dataset is necessary. For this, three publicly accessible datasets used to evaluate the model in [12] are used: 1) IEEE Covid Chest X-ray dataset [4] 2) COVID-19 Radiography Database [1] and 3) COVID-19 Chest X-ray Dataset Initiative [2].

The choice of the datasets was influenced by the fact that they are collected and shared from reliable sources and they are all open-source and freely accessible to the general public and research communities.

The three datasets contain some duplicate images that need to be removed using a hashing algorithm.

2. Contributions

Addressing this problem is summarized in the following steps:

- First, we will start with a data preparation step to remove the duplicates, since we are using multiple datasets. This is done using Hash algorithms, by detecting the images that have the same hash code and removing their duplicates.

- Second, We will Build two types of GANs: (1) Auxiliary Classifier GAN and (2) DC-GAN to create a synthetic dataset of COVID X-rays and augment the original dataset.

- Third, we compare the two GANs using the frechet-inception distance (FID) and select the best GANs that has the lower FID score to generate the data.

- Finally, train a CNN on the original dataset and the augmented dataset using synthetic data and conclude on the effectivness of our approach.

3. Models

3.1. GANs

Generative Adversarial Networks (GAN) are a type of generative modeling that uses deep learning methods (e.g., CNNs) to generate data.

GANs formulate the problem of generating data as a supervised learning problem with two sub-models: the generator and the discriminator. The generator model is trained to generate new samples from a random vector z (latent vector) and the discriminator model tries to classify the samples as either real (from the distribution) or fake (generated). The training consists of a zero-sum game (minmax game) where the generator tries to generate more realistic samples and the discriminator learns a mapping function to classify the inputs as real or fake samples. The Min-Max game means that when the generator fools the discriminator, the latter is penalized. While the generator is penalized and its model parameters are updated when the discriminator successfully indetifies the fake samples from the real ones. This is summarized in the following loss function:

$$\min_G \max_D V(D,G) =$$

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

where D and G are the discriminator model and the generator model , p_{data} is the distribution of the data, $p_{\mathbf{z}}$ is the distribution of the latent vector.

3.2. DC-GANs.

Deep Convolutional Generative Adversarial Networks (DCGANs) is a type of GANs based on convolutional layers that has some architectural constraints that assure a better convergence and demonstrate its capability of learning salient representations in both generator and discriminator. Some of these constraints are presented in section 4.1.3.

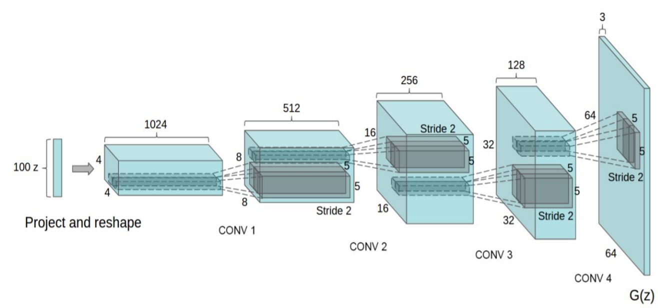


Figure 1: Generator architecture.

3.3. AC-GAN

The auxiliary classifier GAN is a type of conditional GAN where the discriminator predicts the class label of a given image and identifies if it's fake or real. This procedure has the effect of stabilizing the training process and helps the GAN to learn a representation for z independent on the class label. This architecture of GANs allows to train the generator and discriminator on a set of classes. Figure 2 illustrates the global scheme of the AC-GAN:

Each generated sample is linked to a class label $c \sim p(c)$ and a latent vector z. The discriminator is then trained to predict the conditional probabilities over sources and classes $P(s|X)$ and $P(c|X)$. The objective function can then be formulated as:

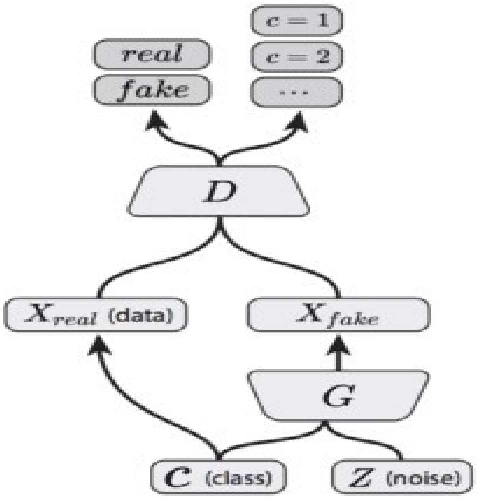


Figure 2: Auxiliary Classifier GAN Scheme.

$$L_S = E[\log P(S = \text{real} \mid X_{\text{real}})] + E\left[\log P\left(S = \text{ fake } \mid X_{\text{fake}}\right)\right]$$

$$L_C = E\left[\log P\left(C = c \mid X_{\text{real}} + E\left[\log P\left(C = c \mid X_{\text{fake}}\right)\right]\right.\right.$$

The discriminator is trained to maximize the sum of the two loss functions while the generator is trained to maximize the log-likelihood of the correct class and minimize the log-likelihood of the correct source (real/fake) to fool the discriminator.

4. Methodology

4.1. GAN Architecture.

4.1.1 Generator.

The input of the generator is a latent noise vector $z \sim \mathcal{N}(0, 0.02)$ and a class label. the latent vector has a dimentionality $z \in \mathbb{R}^{128}$ and the label is just one value that will be passed to an embedding layer of 50 dimensions before being passed through a dense layer and reshaped to a (7x7x1) tensor. On the other hand, a dense layer and activation function is applied to the noise vector to get a tensor that can be reshaped to (7x7x1024) and it's interpreted as having 1024 copies of low resolution (7x7) images. The noise tensor and the label tensor will then be concatenated before being fed to a set of transpose convolutional blocks to upsample the (7x7x1024) feature maps to (112x112x3). Each block is composed of a 2D transposed convolution, a batch normalization and a relu activation function, except the last layer that doesn't contain a batch normalization and its activation function is a hyperbolic tangent. The kernel size of the transpose convolutions is (5,5) with a (2,2) stride to upsample the generated image by a factor of 2 in each block. The number of feature maps in each block is $\{1024, 512, 256, 128, 3\}$ respectively.

4.1.2 Discriminator.

The discriminator is composed of 5 blocks, each containing a convolution with a (3,3) kernel, a batch normalization layer, a leaky relu activation with parameter $\alpha = 0.2$ and a dropout of 0.5. The first convolution layer has a (1,1) stride while all the succeeding layers have a (2,2) stride. The size of the feature maps in each block is respectively $\{3, 64, 128, 256, 512\}$. The discriminator has 2 outputs, the first is a dense layer with 1 neuron and sigmoid activation for source classification (real/fake) and the second is a dense layer with a number of neurons equal to the number of classes followed by a softmax activation.

4.1.3 Architecture Guidelines.

The architecture of the GAN and training procedure used respects the following tips presented in NIPS 2016 Workshop on Adversarial Training:

- **Avoid sparse gradients** by using strided convolutions and transpose convolutions instead of Max pooling and replace the relu by leaky relu in the discriminator.
- **Modified loss function:** use $max(log(D))$ instead of $min(log(1-D))$ because the latter formulation suffers from vanishing gradient. In practice, this can be achieved by switching the labels when training the GAN (fake ¡= real).

- **Discrete variables:** Embedding the input labels.

4.1.4 Training Procedure.

The generator and discriminator are stacked to form the GAN model and set the discriminator weights as non-trainable. First, the discriminator is trained on a batch of real and fake images generated by the generator. Then, sample a batch of noise vectors and class labels, invert the source labels of the samples (give them the label 'real') and train the GAN mode using these samples. Since the discriminator weights are non-trainable then only the generator is updated in this step. The optimizer used to train the model is Adam [6] with a learning rate 0.0002 and $\beta = 0.5$. The loss functions used for the training are the binary cross entropy for the classification of the source (real/fake) and sparse categorical crossentropy for multi-class classification. The model was trained for 500 epochs and we used a batch size equal to 64. The model takes around 9 hours to train on GPU.

5. Evaluation.

Generative adversarial networks lack an objective function, which makes it difficult to compare performance of different models.

While several measures have been introduced, as of yet, there is no consensus as to which measure best captures strengths and limitations of models and should be used for fair model comparison. [**?**]

- There is no objective function used when training GAN generator models, meaning models must be evaluated using the quality of the generated synthetic images.

- Visual examination of samples by humans is one of the common and most intuitive ways to evaluate GANs [**?**].

- Quantitative measures, such as the inception score and the Frechet inception distance, can be combined with qualitative assessment to provide a robust assessment of GAN models.

Visual Inspection is time consuming and requires the knowledge of the generated data.

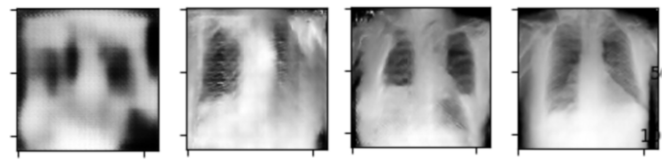


Figure 3: AC-GAN generated images after 100, 200, 300 and 400 epochs.

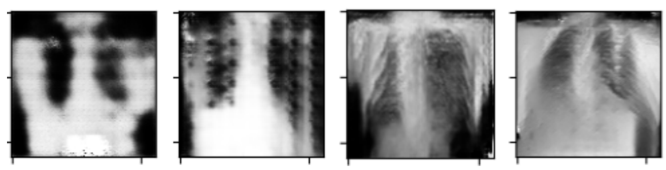


Figure 4: DC-GAN generated images after 100, 200, 300 and 400 epochs.

5.1. Frechet Inception Distance (FID)

The FID is a quantitative metric used to evaluate the quality of generated images and thus evaluate the performance of the GANs. This metric have been proposed as an improvement to the Inception score [5].

The FID uses the last pooling layer of the inception v3 model to compute the feature vector of a collection of real and generated images. The two sets of real and generated feature vectors are assumed to be multivariate gaussians distributions with mean and covariance matrix (μ_r, Σ_r) for real images and (μ_g, Σ_g) for the generated ones. Then, a distance between the two distributions is computed using the Wasserstein-2 distance (Frechet distance). The distance is expressed as follows:

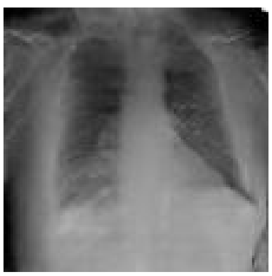
$$d^2 = ||\mu_r – \mu_g||^2 + Tr(\Sigma_r + \Sigma_g – 2 * \sqrt{\Sigma_r * \Sigma_g})$$

A low value of the FID indicates that the two distributions are close, which means that the generated images are more realistic and high quality images.

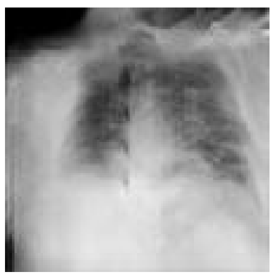
We have used this metric to compare X-ray images generated using AC-GAN and DCGAN.

GAN	Frechet-Inception distance
AC-GAN	280.33
DCGAN	357.917

Table 1: Comparison of AC-GAN and DCGAN using FID.



(a) AC-GAN



(b) DCGAN

Figure 5: Sample generated image using AC-GAN and DCGAN.

This metric represents a result computing using a hundred of generated images using both GANs. As the table shows, the AC-GAN clearly generates covid-xray images more close to the realistic ones than the DCGAN does. This can be attributed to the fact that the AC-GAN takes advantage of the available labels and using both Covid and Non-Covid images which increases drastically the size of the dataset used to train the GAN.

Finaly, we can see from Figure 5 that the image generated using AC-GAN is more realistic than the one generated with DCGAN. Based on this conclusion, we have used the AC-GAN to train the CNN to detect COVID Xray images.

5.2. Classification Performance.

The original problem we want to solve is the detection (classification) of Covid X-ray images. Thus, a natural way to assess the quality of the generated dataset is by evaluating a classification model (Convolutional neural networks) trained on: (1) Only the original collected dataset (2) On the original and generated dataset.

Table2 illustrates that our approach helped improving the test accuracy 3.5% and therefore validates the concept that using covid generated images can improve the detection of Covid. Note that the dataset used to train the CNN is balanced, thus we provided global accuracy results. Also, the test set is only formed of the original dataset and doesn't include any generated images.

Dataset	Original dataset	Augmented dataset
Train	99%	99%
Test	84%	87.5%

Table 2: Classification accuracy using VGG16.

6. Discussion and conclusion

In this project, we have developed an AC-GAN for generating COVID X-ray images, following some architecture guidelines to assure the stability of the training. We evaluated the quality of the generated images in two ways. First, by comparing the frechet-inception score of the generated images using AC-GAN with the ones generated using DCGAN which allowed us to confirm that using the labels in the training of the GAN can improve the quality of the generated images. Second, we trained the CNN model on both original dataset and augmented dataset using images generated from AC-GAN and we concluded that the data augmentation helps imrpoving the test accuracy by 3.5%. Although the results are promising, this analysis has a major limitation: the generated labels cannot be evaluated without medical assistance and clinical testing. Meaning that the this approach can be used to enhance COVID detection but with some supervision of medical experts to check the reliability of the system.

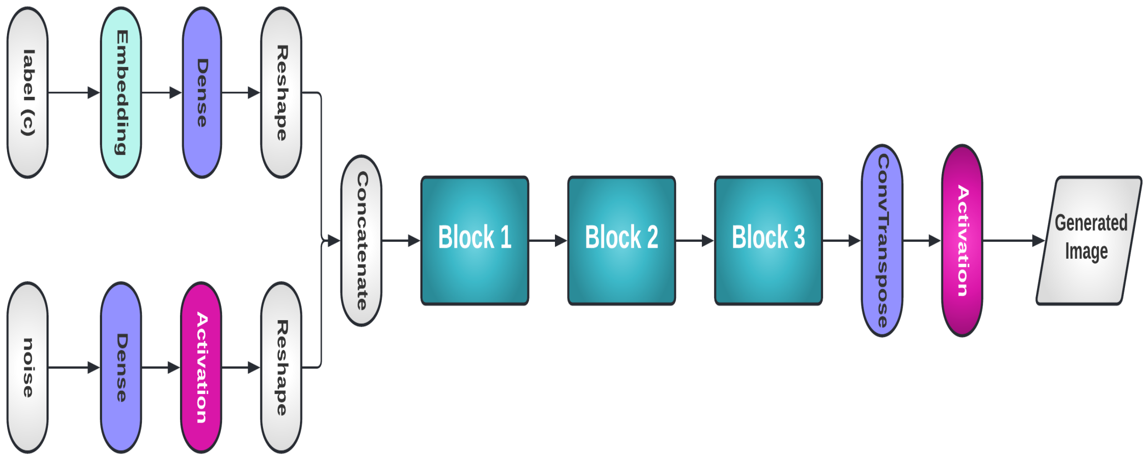


Figure 6: AC-GAN generator architecture.

References

[1] Covid19 radiography database., 2021.

[2] Agchung. Covid-19 chest x-ray dataset initiative.

[3] A. Beers, J. Brown, K. Chang, J. P. Campbell, S. Ostmo, M. F. Chiang, and J. Kalpathy-Cramer. High-resolution medical image synthesis using progressively grown generative adversarial networks, 2018.

[4] J. P. Cohen, P. Morrison, L. Dao, K. Roth, T. Q. Duong, and M. Ghassemi. Covid-19 image data collection: Prospective predictions are the future. *arXiv 2006.11988*, 2020.

[5] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium, 2018.

[6] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, 2017.

[7] M. Loey, F. Smarandache, and N. E. M. Khalifa. Within the lack of chest covid-19 x-ray dataset: A novel detection model based on gan and deep transfer learning, 2020.

[8] D. Nie, R. Trullo, J. Lian, C. Petitjean, S. Ruan, Q. Wang, and D. Shen. Medical image synthesis with context-aware generative adversarial networks. *Medical image computing and computer-assisted intervention : MICCAI ... International Conference on Medical Image Computing and Computer-Assisted Intervention*, 10435:417–425, Sep 2017. 30009283[pmid].

[9] H. R. Roth, L. Lu, J. Liu, J. Yao, A. Seff, K. Cherry, L. Kim, and R. M. Summers. Improving computer-aided detection using convolutional neural networks and random view aggregation. *IEEE Transactions on Medical Imaging*, 35(5):1170–1181, 2016.

[10] C. Shorten and T. M. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1), July 2019.

[11] N. Tajbakhsh, J. Y. Shin, S. R. Gurudu, R. T. Hurst, C. B. Kendall, M. B. Gotway, and J. Liang. Convolutional neural networks for medical image analysis: Full training or fine tuning? *IEEE Transactions on Medical Imaging*, 35(5):1299–1312, 2016.

[12] A. Waheed, M. Goyal, D. Gupta, A. Khanna, F. Al-Turjman, and P. R. Pinheiro. Covidgan: Data augmentation using auxiliary classifier gan for improved covid-19 detection. *IEEE Access*, 8:91916–91923, 2020.

[13] D. Zhao, D. Zhu, J. Lu, Y. Luo, and G. Zhang. Synthetic medical images using famp;bgan for improved lung nodules classification by multi-scale vgg16. *Symmetry*, 10(10), 2018.