Parallel/Distributed Generative Adversarial Neural Networks for Data Augmentation of COVID-19 Training Images

Jamal Toutouh[1](✉), Mathias Esteban[2], and Sergio Nesmachnow[2]

[1] Massachusetts Institute of Technology, Cambridge, MA, USA
toutouh@mit.edu
[2] Universidad de la República, Montevideo, Uruguay
{mathias.esteban,sergion}@fing.edu.uy

Abstract. This article presents an approach using parallel/distributed generative adversarial networks for image data augmentation, applied to generate COVID-19 training samples for computational intelligence methods. This is a relevant problem nowadays, considering the recent COVID-19 pandemic. Computational intelligence and learning methods are useful tools to assist physicians in the process of diagnosing diseases and acquire valuable medical knowledge. A specific generative adversarial network approach trained using a co-evolutionary algorithm is implemented, including a three-level parallel approach combining distributed memory and fine-grained parallelization using CPU and GPU. The experimental evaluation of the proposed method was performed on the high performance computing infrastructure provided by National Supercomputing Center, Uruguay. The main experimental results indicate that the proposed model is able to generate accurate images and the 3×3 version of the distributed GAN has better robustness properties of its training process, allowing to generate better and more diverse images.

Keywords: Computational intelligence · Learning · Generative Adversarial Networks · Data augmentation · COVID-19

1 Introduction

Computational intelligence and automated learning have been successful tools for a wide range of applications [4]. In particular, generative models are powerful methods for learning and gaining knowledge about data, data distributions, and other valuable information. Generational models have been one of the most versatile unsupervised learning techniques in recent years [22,23].

Generative Adversarial Networks (GANs) [5] are powerful methods originally proposed to train generative models by using unsupervised learning. Nowadays, they have been extended to consider other approaches, including semi-supervised and fully supervised learning, and reinforcement learning. GANs propose a method for learning and estimate the distribution of data from a training set, to produce new information units that approximate the original set. Two artificial neural networks (ANN) are used (a generator and a discriminator), and adversarial learning is applied to optimize the learning process and the resulting outcome. GANs have been successfully applied to many problems, especially those concerning multimedia information (e.g., images, sound, and video), in science, design, art, games, and other areas [13]. Even complex tasks such as autonomous driving [20] and medical assistance [9] have been addressed using GANs. In fact, medical assistance provides a very interesting application area for GANs, since they can provide new insights to the interpretation process of medical information stored in different media (radiography, ultrasound, etc.).

In this line of work, this article presents a parallel/distributed GAN approach for image data augmentation, applied to generate COVID-19 training samples for computational intelligence methods. This is a relevant problem nowadays, considering the recent COVID-19 pandemic. The proposed approach is based on a two-level parallel model for training and configuration, and co-evolutionary training considering a dataset of real X-rays chest images. The proposed model is implemented and executed in the high performance computing infrastructure provided by National Supercomputing Center (Cluster-UY), Uruguay.

The main contribution of the research reported in this article include: i) a proposal for applying GANs to the data augmentation problem for medical images, to assist COVID-19 diagnosis; ii) a two-level parallel model developed to take advantage of high performance computing facilities; and iii) the experimental evaluation of the proposed distributed parallel/distributed GAN approach considering a dataset of real X-rays images.

The article is organized as follows. Section 2 describes the data augmentation problem to assist COVID-19 detection and reviews related works. Section 3 describes the proposed parallel/distributed approach. The experimental evaluation and results are reported and discussed in Sect. 5. Finally, the conclusions and the main lines for future work are presented in Sect. 6.

2 Data Augmentation for Medical Images to Assist COVID-19 Detection

This section describes the problem of data augmentation for medical images, its application to assist COVID-19 detection and reviews related works.

2.1 Data Augmentation for Medical Images and COVID-19 Detection

The technique of medical imaging consists of creating visual representations of the human body for clinical analysis to help diagnosing and treating diseases. Using this technique, physicians and organizations gather useful information to build databases of both normal and pathological images to identify physiological anomalies. To that end, several imaging technologies are applied, including radiography (X-rays), magnetic resonance, ultrasound, thermography, and others.

In recent years, computational intelligence methods have been increasingly applied to assist medical diagnostics. This way, learning models have been applied to provide a new insight to the process of medical image interpretation, which was previously limited to radiologists. However, learning models require significantly large datasets for training. Within this context, an important problem arises: expanding the base of knowledge about medical conditions in order to perform a proper training of computational intelligence/learning methods.

Image data augmentation [17] allows overcoming the difficulties associated to having few data to build training datasets for learning methods. The augmentation technique is useful to expand the size of a given training dataset, by creating modified versions of existing images. Then, learning models trained with the expanded dataset are able to produce better models and improve detection results, considering that more information can be extracted from the original dataset through augmentations. Useful, transfer learning and generalization are applied to broaden the applicability of built models.

Traditional approaches for image data augmentation are based on applying image operations, like point processing and frame transformations, geometric (resize, crop) and color transformations, merging, equalization, etc. On the other hand, deep learning approaches include adversarial training, neural style transfer, and GAN-based data augmentation.

Nowadays, there is an increasing interest on learning approaches to help dealing with outbreaks, like the recent COVID-19 pandemic.

2.2 Related Works

Computational intelligence have been applied for medical image processing, to address disease detection, help and guide diagnostics, and to improve medical imaging research [11].

Kovalev and Kazlouski [9] studied the generation of artificial biomedical images to be used as a substitute for real image datasets, focusing on generating realistic chest X-ray images using Deep Convolutional GAN and Progressive Growing GAN (growing both generator and discriminator continuously). A benchmark classification problem was solved using real and synthetic images for the problem of detecting breast tumor, using data from the lymph node database (image size: 256×256). The classification accuracy dropped between 2.2%–3.5% (considered as "acceptable for practical applications"), improving between two and four times the results of Loopy Belief Propagation and Random Forest.

The application of GANs in radiology, specifically for detecting congestive cardiac failure on chest radiographies, was studied by Seah et al. [16]. A generative model (Generative Visual Rationales, GVR) was trained on an unlabeled subset of a frontal chest radiographies set and a traditional ANN encoder was trained on a labeled subset. The experimental evaluation considered a custom overfitted model developed for comparison and classification by experts. The main results allowed to conclude that features by ANN can be identified using GVR, thus allowing detection of bias and overfitted models. The preprint article by Khalifa et al. [8] also explored the use of GANs and transfer learning, but in this cases focusing on a more general pneumonia disease. Several deep transfer learning models were studied to detect the pneumonia from images, using a training set of just 10% of real data and the other 90% generated using a GAN. No computational efficiency evaluation was reported, but authors worked under the assumption of including few layers in the underlying GANs, to reduce memory consumption and execution time. The main results showed that using GANs as augmentation technique allows improving the robustness of the proposed model, making it less prone to overfitting. Thus, using the proposed method, better images can be generated.

Regarding data augmentation of medical images, Bhagat and Bhaumik [2] proposed a method for generating synthetic chest X-ray images of patients with pneumonia, using GANs. Generator and discriminator started with low-resolution images (4×4) and the resolution increased step-by-step when including more layers (up to 128×256 and 256×128). A deep convolutional neural network was trained using the generated images, to solve the classification problem. The prediction accuracy improved when considering the augmented database.

Several recent articles have extended previous approach to help addressing the problem of coronavirus disease detection. Loey et al. [10] applied GAN and deep transfer learning for COVID-19 detection considering chest X-ray images as input, extending the proposal by Khalifa et al. [8]. Deep transfer models were studied over a training dataset containing images of four classes (COVID-19, normal, pneumonia bacterial, and pneumonia virus). GANs were applied as subordinate method for expanding the training dataset to contribute improving the detection accuracy. Waheed et al. [21] developed an auxiliary classifier GAN to overcome the problem of limited images available for COVID-19 detection when using convolutional neural networks (CNNs). The main results demonstrate that using synthetic images produced by the auxiliary classifier GAN allows improving the COVID-19 detection accuracy of CNN from 85% to 95%.

On the other hand, some recent advances have been developed on applying parallel computing to speed-up the training process of GANs and their effectiveness. Im et al. [7] recognized the difficulties of GAN training in practice, and proposed the Generative Adversarial Parallelization (GAP) framework for the simultaneous training of several GANs that share their discriminators. This approach extends the two-player generative adversarial game into a multi-player game, thus transforming the training from being a tightly-coupled problem (between generator and discriminator) to a more loosely one. Multiple models are required, each one with its own parameters, which are structured in a bipartite layout for competence. Instead of applying a traditional data-parallel approach on the parameter space, GANs are trained randomly swapping different discriminators/generators to produce synergy. The model was implemented in Theano and executed in GPU, without transfers through host memory (to reduce communication overheads). Empirical results showed that the GAP model allows improving mode coverage, convergence, and quality. No efficiency or execution time analysis was reported. Up to now, no parallel GAN training for COVID-19 detection has been proposed.

The analysis of related works allows concluding that no previous proposals have explored the application of parallel computing to implement efficient and accurate IA models for COVID-19 detection. This article contributes in this line of research, by applying a two-level parallel model for GANs training and configuration, applied to data augmentation for medical images to contribute in COVID-19 research.

3 The Proposed Parallel/distributed GANs for COVID-19 Data Augmentation

This section describes the proposed approach applying parallel/distributed GANs for COVID-19 data augmentation.

3.1 Generative Adversarial Networks

GANs are computational intelligence methods that intends to learn the specific distribution of a given training dataset, to synthesize samples using the estimated distribution. GANs consist of two ANNs, a generator and a discriminator, that applies adversarial learning to optimize their parameters. The discriminator try to learn how to distinguish the natural/real samples from the artificial/fake samples produced by the generator. The generator is trained to transform its inputs from a random latent space into artificial/fake samples to deceive the

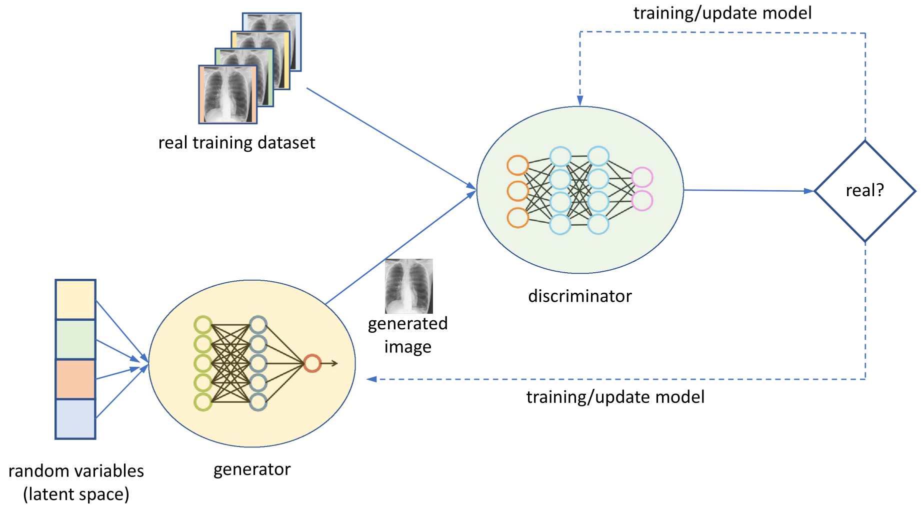


Fig. 1. General diagram of a generative adversarial network

discriminator (see Fig. 1). The GAN training problem is formulated as a minmax optimization problem by the definitions of generator and discriminator. In the last years, GANs have demonstrated to be efficient methods for learning [5].

3.2 Distributed GAN Training

The proposed approach applies the methodology introduced by Lipizzaner [15] and Mustangs [19]. A distributed GAN training is performed by applying co-evolutionary algorithms (coEA).

In the co-evolutionary distributed GAN training, two populations are evolved, one for generators and one for discriminators. These two populations are trained by competition between them. Individuals in each population are located in a spatial structure, an underlying toroidal grid. The concept of neighborhood is applied to define those individuals that participate in the training phase for both generators and discriminators.

The approach using distributed coEAs has shown to be effective overcoming the main pathologies in GAN training, namely modal collapse, vanish gradient, and non-convergence. Futhermore, cellular training allows implementing a data-parallel approach where each cell is trained using reduced subsets without affecting the overall quality of the implemented GANs, considering the difference between the probability distributions of generated samples and the original samples (or the similarity of generated images, when used with that goal). The data-parallel approach also allows improving the computational efficiency of the training process, since a smaller amount of training data batches are needed [18]. Figure 2 presents a diagram of the proposed distributed training for GANs implemented in Lipizzaner.

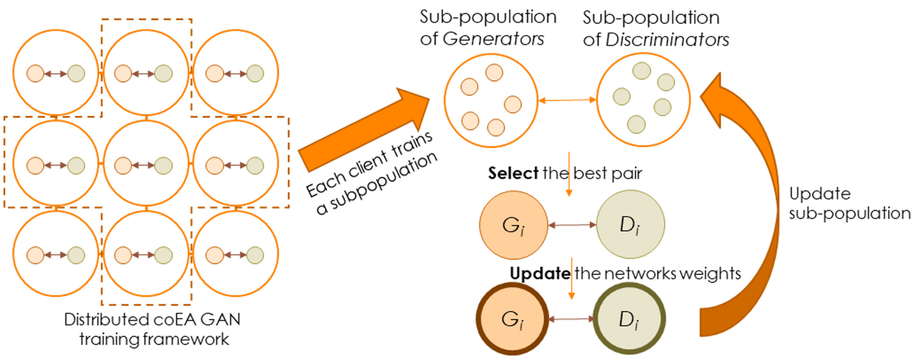

Distributed coEA GAN training framework
Each client trains a subpopulation
Sub-population of Generators
Sub-population of Discriminators
Select the best pair
G_i
D_i
Update the networks weights
G_i
D_i
Update sub-population

Fig. 2. Diagram of the distributed training for GANs proposed by Lipizzaner

4 Distributed GANs for COVID-19 Images Generation

This section describes the proposed approach for data augmentation of COVID-19 datasets using GANs.

4.1 Overall Description

The proposed approach for data augmentation of the COVID-19 image dataset applies the generative paradigm previously applied for generating new images of handwritten digits, small object photograph, and faces.

The approach consist in sampling from an existing database of real chest X-ray images to train the discriminator, while the generator uses a random variable from a Gaussian distribution, $z \sim \mathcal{N}(0, 1)$. Then, generator and discriminator are trained using the distributed co-evolutionary approach. Figure 3 shows two sample real chest X-ray images used for training.

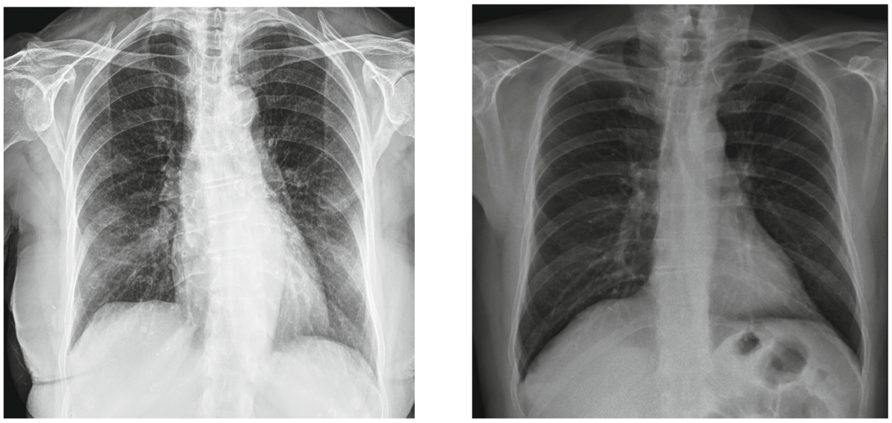


Fig. 3. Sample chest X-ray images used for GAN training

4.2 Implementation Details

Both the generator and discriminator models were implemented as Multilayer Perceptrons (MLP). MLP are one of the classical and most used types of neural network [6]. A MLP is comprised of perceptrons or neurons, organized on layers. At least two layers are used: input layer, which receives the problem data as input and output layer, and output layer, which produces the results. In between, one or more hidden layers can be included to provide different levels of abstraction to help with the learning goal. MLP are feedforward ANNs, meaning that connections between neurons do not form a cycle; information moves in only one direction (forward) from neurons on the input layer, through neurons on the hidden layers (if they exist) and finally to neurons in the output layer. Except for the input neurons, all other neurons use a nonlinear activation function, which separates MLP to simpler linear perceptrons. This feature makes MLP able to distinguish not linearly separable data. MLP are usful for dealing with structured data (e.g., tables), classification/prediction problems whith labeled inputs, regression problems, etc. Furthermore, MLP hava a high flexibility and applicability to learn any mapping function from inputs to outputs. Their flexible nature allows MLP to be applied to other types of data, e.g., pixels of an image, such as proposed for medical images in this article.

To deal with the proposed problem, the proposed approach explores the use of MLPs using four and five layers as underlying ANN architectures for both generators and discriminators. The generators and discriminators have the same input and output sizes. The generator input from the latent space has size 64 and the output has size 16384 (to encode 128×128 gray-scale images). The discriminator has input size of 16384 (an image) and output size of one to encode the truth value (real or fake). Both type of MLP use linear layers. Hidden layers apply the leaky version of a rectified linear unit (LeakyRelu) as activation function, the generators output layer the hyperbolic tangent (Tanh), and the discriminator output layer the sigmoid function. Table 1 report the main features of the MLP architectures in both generators and discriminators.

Table 1. Main features of the MLP architectures used in generators and discriminators of the proposed GAN for COVID-19 images generation

Layers	Four-layer MLP GAN		Five-layer MLP GAN	
	Generator	Discriminator	Generator	Discriminator
First hidden layer	64×256	16384×256	64×256	16384×512
Second hidden layer	256×256	256×256	256×256	512×256
Third hidden layer	–	–	256×512	256×256
Output	256×16384	256×1	256×16384	256×1

One of the key components of GANs is the `DataLoader` function, which allows performing the domain decomposition by dividing the training data in batches to be processed iteratively. In the proposed implementation, the Dataloader function reads the images from the training set and encodes each pixel with a real number in (0,1). Since gray-scale images are used, a single number is enough to provide the needed encoding (0 represents white and 1 represents black). After that, since images have different size because they come from different databases or have been possibly acquired using different devices, a resize transformation is applied in order to convert them to a unique resolution (set as an input parameter, in the reported research it is 128×128). The resulting vector of 16 384 positions is stored in memory. After all images are read, a tensor of $16\,384 \times \#TD$ is transferred to GPU, where $\#TD$ is the size of the training dataset.

4.3 Parallel Model

A three-level parallel model is applied in the proposed implementation, following the idea of the parallel/distributed implementation of cellular training for GANs proposed in our previous work [14].

The upper level applies a Multiple-Instruction-Multiple-Data (MIMD) parallel model to study one of the most important parameters of the model proposed by GANs: the architecture of the underlying ANN. A domain decomposition approach is applied on the space of candidate architectures, defined ad-hoc for the problem, and the space of relevant parameters (p_i) and a set of candidate values (v_i). A parallel master-slave model is followed, using a distributed memory paradigm implemented in the MPI for Python package. Considered architectures and parameter sets are assigned on-demand to a set of distributed processes, executed according to the availability of computational resources. A dynamic load balancing procedure is applied.

In turn, the medium level applies a distributed training approach, according to the parallel co-evolutionary model proposed by Lipizzaner. In the Lipizzaner algorithm, the population of generators and discriminators are distributed using a logical spatial grid and applying a cellular parallel model for EAs [1]. Parameters of the training process are explored and optimized competitively, in an asynchronous parallel execution of all cells in the grid defined by Lipizzaner. A master process performs the data distribution, by assigning populations to grid cells, and defines the communication channels according to the neighborhood topology, accounting for the grid size. Communications between processes are performed to exchange relevant information along the evolution.

Finally, the lower level applies the parallel training of the studied GANs in GPU, applied a Single-Instruction-Multiple-Data approach. This parallel training is implemented using the `PyTorch` open source machine learning library, widely employed for applications related to multimedia/image processing and computer vision. Unlike other libraries for ANN training like TensorFlow, PyTorch does not include a specific library for execution on GPU. Thus, the training dataset is loaded, a tensor is created (dimension #images $\times$ image width $\times$ image height $\times$ #channels). That tensor is transferred to GPU and processed in batches.

A schema of the proposed parallel model is presented in Fig. 4 (the sets of parameters and values on which the parameter sweep is performed could be different for different architectures).

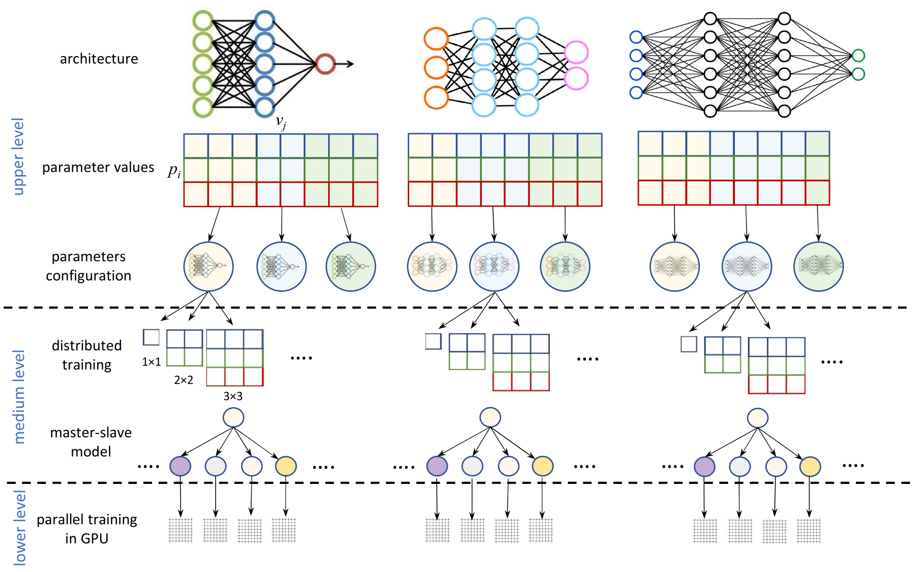


Fig. 4. Diagram of the three-levels parallel model for GANs training

5 Experimental Evaluation

This section describes the experimental evaluation of the proposed approach using GANs for COVID-19 images generation.

5.1 Evaluation Methodology, Training and Validation Instances

Methodology. The experimental evaluation is focused on analyzing the parameter configuration of the learning method and the quality of the generated images. The stop condition is 1000 *training epochs*. One training epoch is the number of iterations required to feed all batches in the training dataset to the generator.

Regarding the proposed approach using Lipizzaner, three configurations of the distributed learning method are studied: 1×1 grid, 2×2 grid, and 3×3 grid.

Metrics and Parameters. The metric considered in the evaluation is the inception score, which is commonly used for evaluating images generated by GANs. It aims at objectively assessing the quality of generated images via two relevant properties that are evaluated simultaneously: likeliness to a specific object to be generated and diversity. Inception score is within the range $(1.0, M)$, being M the highest inception score for the considered dataset. The computational efficiency of the proposed parallel model is also evaluated. In turn, the quality of generated images is also evaluated by examination of representative samples.

The studied parameters are the ANN architecture, the batch size, the Gaussian mutation probability (p_M), and the initial learning rate (l_0). All reported

Table 2. Inception score results for the studied configuration parameters.

Parameter	Value	Minimum	Median	Iqr	Max
Network architecture	Four layers perceptron	1.43	**1.68**	0.14	1.88
	Five layers perceptron	1.00	**1.68**	0.27	**2.25**
Batch size	50	1.00	**1.70**	0.14	2.05
	75	1.00	1.69	0.16	**2.25**
	100	1.00	1.64	0.18	2.07
p_M	0.3	1.00	1.66	0.18	1.95
	0.4	1.00	**1.69**	0.18	**2.25**
	0.5	1.00	1.68	0.19	2.07
l_0	0.00010	1.26	1.65	0.18	1.94
	0.00025	1.04	**1.73**	0.12	1.93
	0.00050	1.00	1.66	0.34	**2.25**

results correspond to 14 independent executions of the proposed GAN performed for each parameter configuration.

Training and Validation Instances. The training of the proposed model was performed using images from the open repository created by Cohen et al. [3], publicly available at https://github.com/ieee8023/covid-chestxray-dataset. The same dataset is considered for computing the inception score.

Development and Execution Platform. The proposed parallel/distributed GAN implementation was implemented in Python3 using pytorch (pytorch.org).

The experimental analysis was performed on National Supercomputing Center (Cluster-UY), Uruguay [12]. Cluster-UY offers up to 30 computing servers, each of them with Xeon Gold 6138 processors with 40 cores, Nvidia Tesla P100 GPUs (12 GB memory), 128 GB of RAM memory, and 300 GB of SSD storage for temporary files, interconnected by Ethernet at 10 Gbps.

5.2 Numerical Results

Quality of Generated Images: Inception Score. Table 2 reports the inception score values obtained for the studied configuration parameters. Results do not follow a normal distribution, thus values of median and interquartile range (Iqr.) are reported as relevant estimators. Figure 5 shows the corresponding boxplots for the parameters with more impact in the inception score values.

Computational Efficiency. Table 3 reports the execution time of the proposed GANs for the two configuration parameters that affect the most to the execution time. Execution parameters that impact the computational efficiency of the training process are the network architecture and the batch size. Other parameters impact on the result quality, but not on the efficiency. Results do not follow

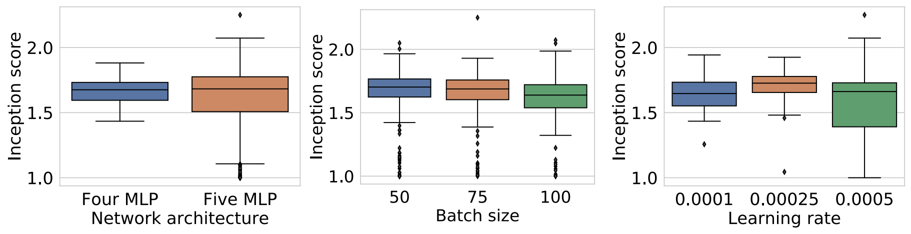


Fig. 5. Inception score boxplots for the studied configuration parameters

Table 3. Execution time for the studied network architectures and batch sizes.

Parameter	Value	Minimum	Median	Iqr	Max
Network architecture	Four layer perceptron	8.15	12.68	5.20	19.38
	Five layer perceptron	10.42	17.22	7.62	27.12
Batch size	50	9.42	15.68	3.73	27.12
	75	8.57	13.95	9.37	21.05
	100	8.15	12.53	4.52	19.38

a normal distribution, thus values of median and interquartile range (Iqr.) are reported as relevant estimators. The non-parametric Wilcoxon test was applied to analyze the results distributions and results confirmed that the differences are statistically significant (p-value < 0.01). Boxplots are presented in Fig. 6.

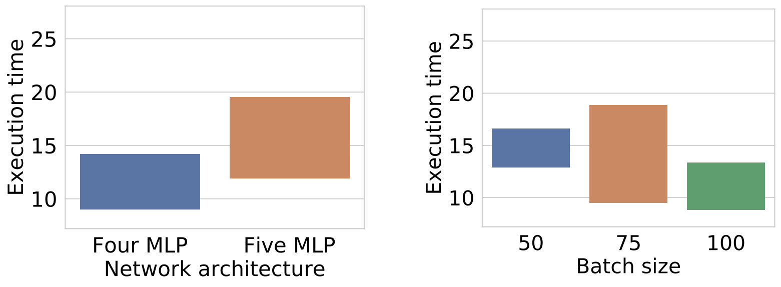


Fig. 6. Execution time boxplots for the studied values of network architecture and batch size parameters

Overall Analysis. Figure 7 presents a 2D graphical comparison of the inception score results and the execution time required for training. The well-known trade-off between results quality and execution time is observed. Considering that the time for a single execution of the proposed GAN model are very reasonable, the configurations that allowed computing the best inception score results were selected for execution of the distributed 2×2 and 3×3 GANs using Lipizzaner. The selected configurations are: *best median* (five layers perceptron, batch size 50, $p_M = 0.4$, and $l_0 = 0.00025$, whose results are marked with a red star in Fig. 7) and *best maximum* (five layers perceptron, batch size 75, $p_M = 0.4$, and $l_0 = 0.0005$, whose results are marked with a blue diamond). These configurations significantly outperformed the inception score results obtained by all others.

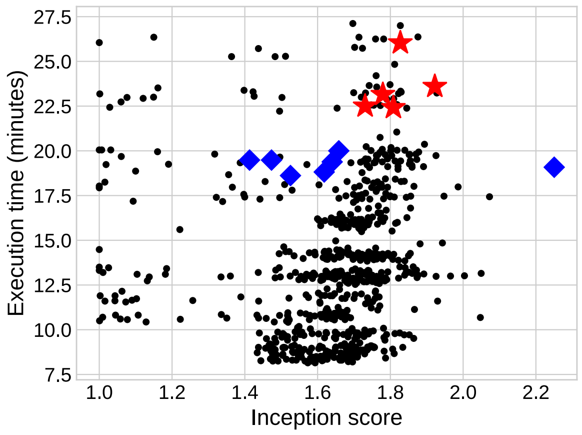
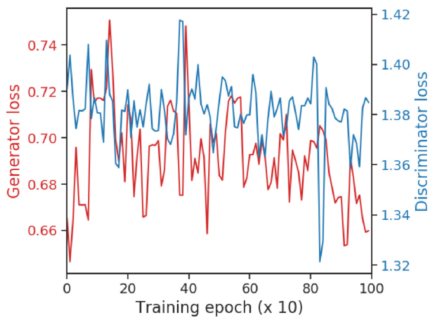


Fig. 7. Inception score vs. execution time (Color figure online)

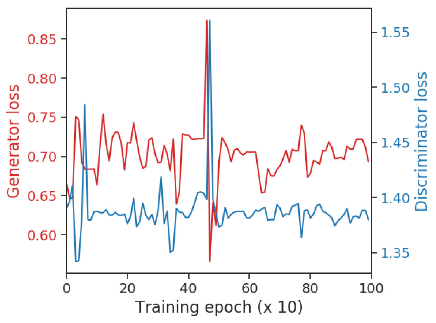
Distributed GANs. Inception score results for 2×2 and 3×3 grids are reported in Table 4. In turn, Fig. 8 presents the evolution of loss during the training of generator (red line) and discriminator (blue line) for one representative client of Lipizzaner, using 2×2 grid (left) and 3×3 grid (right).

Table 4. Inception score results for the different grid sizes

Grid	Best maximum configuration			Best median configuration		
	Minimum	Median	Maximum	Minimum	Median	Maximum
2×2	1.53	1.66	1.68	1.61	1.63	1.68
3×3	1.34	1.58	1.73	1.80	1.83	1.86
Overall	1.34	1.63	1.73	1.61	1.74	1.86



(a) 2×2 grid



(b) 3×3 grid

Fig. 8. Loss during the training for one representative client of Lipizzaner (Color figure online)

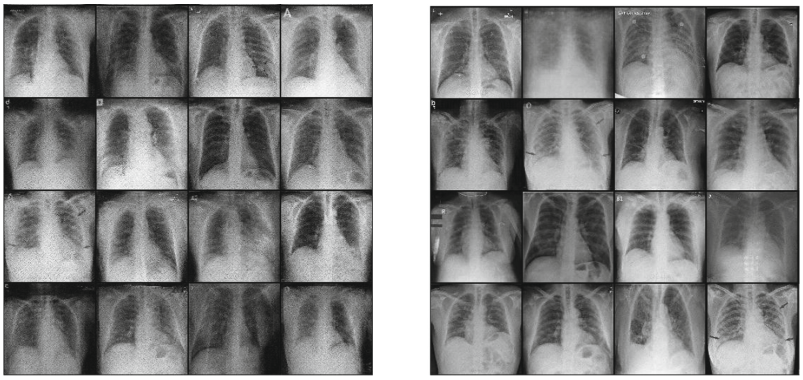


Fig. 9. Sample images generated by the proposed distributed GAN approach

Results reported in Table 4 indicate that the 3 × 3 grid trained better generators, able to create better samples (higher inception scores). Loss evolution in Fig. 8 show that the 3 × 3 grid also provides a more robust behavior of the resulting GAN, confirmed by a the loss function with fewer peaks and smoother variations, when compared with the one computed using the 2 × 2 grid. This result implies a better training, less prone to typical pathologies of GANs, thus producing better and more diverse images.

Sample Generated Images. Figure 9 shows two sample generated images using the proposed GAN approach.

6 Conclusions and Future Work

This article presented parallel/distributed GANs for image data augmentation, applied to the relevant problem of generating X-rays chest COVID-19 samples.

The proposed implementation applies a parallel model in three levels to combine Single-Program-Multiple-Data paralellism for parameters configuration, a masteer slave model to implement a distributed co-evolutionary training, and Single-Instruction-Multiple-Data using pytorch for training in GPU.

The experimental methodology was oriented to evaluate the image quality, considering the required execution time. The high performance computing infrastructure of National Supercomputing Center, Uruguay, was used.

The main results indicate that the proposed model is able to generate accurate images. The distributed GAN using a 3×3 neighborhood computed the best results, achieving a more robust training and generating better and more diverse images. These results allow concluding that the proposed method is useful for generation of synthetic COVID-19 images.

The main lines for future work are related to extend the evaluation of the proposed approach by studying larger training datasets, other synthetic methods for augmentation and different ANN architectures for both generator and discriminator in the proposed model.

Acknowledgment. The work of S. Nesmachnow is partly supported by ANII and PEDECIBA, Uruguay. J. Toutouh has been partially funded by EU Horizon 2020 research and innovation programme (Marie Skłodowska-Curie grant agreement No 799078), by the Spanish MINECO and FEDER projects TIN2017-88213-R and UMA18-FEDERJA-003, and the Systems that learn initiative at MIT CSAIL.

References

1. Alba, E., Luque, G., Nesmachnow, S.: Parallel metaheuristics: recent advances and new trends. Int. Trans. Oper. Res. **20**(1), 1–48 (2012)
2. Bhagat, V., Bhaumik, S.: Data augmentation using generative adversarial networks for pneumonia classification in chest Xrays. In: 5th International Conference on Image Information Processing (2019)
3. Cohen, J., Morrison, P., Dao, L.: COVID-19 Image Data Collection (2020). Preprint arXiv:2003.11597v1
4. Engelbrecht, A.: Computational Intelligence: An Introduction. Wiley, Hoboken (2007)
5. Goodfellow, I., et al.: Generative adversarial nets. In: Advances in Neural Information Processing Systems, pp. 2672–2680 (2014)
6. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning. Springer, New York (2009). https://doi.org/10.1007/978-0-387-84858-7
7. Im, D., Ma, H., Kim, C., Taylor, G.: Generative adversarial parallelization (2016). Preprint arXiv:1612.04021
8. Khalifa, N., Taha, M., Hassanien, A., Elghamrawy, S.: Detection of Coronavirus (COVID-19) Associated Pneumonia based on Generative Adversarial Networks and a Fine-Tuned Deep Transfer Learning Model using Chest X-ray Dataset (2020). arXiv preprint 2004.01184. Accessed June 2020
9. Kovalev, V., Kazlouski, S.: Examining the capability of GANs to replace real biomedical images in classification models training. In: Ablameyko, S.V., Krasnoproshin, V.V., Lukashevich, M.M. (eds.) PRIP 2019. CCIS, vol. 1055, pp. 98–107. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-35430-5_9

10. Loey, M., Smarandache, F., Khalifa, N.: Within the lack of chest COVID-19 x-ray dataset: a novel detection model based on GAN and deep transfer learning. Symmetry **12**(4), 651 (2020)
11. Morra, L., Delsanto, S., Correale, L.: Artificial Intelligence in Medical Imaging. CRC Press (2019)
12. Nesmachnow, S., Iturriaga, S.: Cluster-UY: collaborative scientific high performance computing in Uruguay. In: Torres, M., Klapp, J. (eds.) ISUM 2019. CCIS, vol. 1151, pp. 188–202. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-38043-4_16
13. Pan, Z., Yu, W., Yi, X., Khan, A., Yuan, F., Zheng, Y.: Recent progress on generative adversarial networks (GANs): a survey. IEEE Access **7**, 36322–36333 (2019)
14. Perez, E., Nesmachnow, S., Toutouh, J., Hemberg, E., O'Reily, U.: Parallel/distributed implementation of cellular training for generative adversarial neural networks. In: 10th IEEE Workshop on Parallel Distributed Combinatorics and Optimization (2020)
15. Schmiedlechner, T., Yong, I., Al-Dujaili, A., Hemberg, E., O'Reilly, U.: Lipizzaner: a system that scales robust generative adversarial network training. In: 32nd Conference on Neural Information Processing Systems (2018)
16. Seah, J., Tang, J., Kitchen, A., Gaillard, F., Dixon, A.: Chest radiographs in congestive heart failure: visualizing neural network learning. Radiology **290**(2), 514–522 (2019)
17. Shorten, C., Khoshgoftaar, T.M.: A survey on image data augmentation for deep learning. J. Big Data **6**(1) (2019)
18. Toutouh, J., Hemberg, E., O'Reilly, U.-M.: Data dieting in GAN training. In: Iba, H., Noman, N. (eds.) Deep Neural Evolution. NCS, pp. 379–400. Springer, Singapore (2020). https://doi.org/10.1007/978-981-15-3685-4_14
19. Toutouh, J., Hemberg, E., O'Reilly, U.M.: Spatial evolutionary generative adversarial networks. In: Genetic and Evolutionary Computation Conference, pp. 472–480 (2019)
20. Uřičář, M., Křížek, P., Hurych, D., Sobh, I., Yogamani, S., Denny, P.: Yes, we GAN: applying adversarial techniques for autonomous driving. Electron. Imaging **2019**(15), 48-1–48-17 (2019)
21. Waheed, A., Goyal, M., Gupta, D., Khanna, A., Al-Turjman, F., Pinheiro, P.R.: CovidGAN: data augmentation using auxiliary classifier GAN for improved Covid-19 detection. IEEE Access **8**, 91916–91923 (2020)
22. Wang, Z., She, Q., Ward, T.: Generative adversarial networks: a survey and taxonomy. preprint arXiv:1906.01529 (2019)
23. Wu, X., Xu, K., Hall, P.: A survey of image synthesis and editing with generative adversarial networks. Tsinghua Sci. Technol. **22**(6), 660–674 (2017)