engineering proceedings

MDPI

Proceeding Paper

Generative Adversarial Networks for the Synthesis of Chest X-ray Images †

Mai Feng Ng and Carol Anne Hargreaves *

Department of Statistics and Data Science, Faculty of Science, National University of Singapore, Singapore 117546, Singapore; maifeng.ng@u.nus.edu

* Correspondence: carol.hargreaves@nus.edu.sg; Tel.: +65-65166623

† Presented at the 3rd International Electronic Conference on Applied Sciences, 1–15 December 2022; Available online: https://asec2022.sciforum.net/.

Abstract: One way to diagnose COVID-19 is to use the Polymerase Chain Reaction (PCR) test. However, this test is rather invasive. An alternative would be to use chest images of the patients to diagnose if the patient has COVID-19. These chest X-ray images have to be manually annotated by a medical professional such as a radiologist, and due to privacy concerns, getting access to readily available and annotated COVID-19 chest X-ray images is difficult. In order to train a deep learning model to perform image classification tasks, it is prudent to train the deep-learning model on a large enough dataset to avoid the problem of overfitting. In this paper, we explore using Generative Adversarial Networks (GANs) as a form of data augmentation technique to enlarge the training data for deep learning models. We first explored how the synthetic data generated by GANs are affected by the training size. Following which, we compared the performance of the two different GAN architectures, namely the Deep Convolutional Generative Adversarial Networks (DCGAN) and the Wasserstein Generative Adversarial Networks with Gradient Penalty (WGAN-GP). We successfully used GANs to generate synthetic COVID-19 chest X-ray images with a Fréchet Inception Distance (FID) score that was below 2.

Keywords: COVID-19; chest X-rays; Generative Adversarial Networks (GANs); synthetic images; Deep Convolutional Generative Adversarial Networks (DCGAN); Wasserstein Generative Adversarial Networks with gradient penalty (WGAN-GP)

Citation: Ng, M.F.; Hargreaves, C.A. Generative Adversarial Networks for the Synthesis of Chest X-ray Images. *Eng. Proc.* **2023**, *31*, 84. https://doi.org/10.3390/ASEC2022-13954

Academic Editor: Nunzio Cennamo

Published: 5 January 2023

1. Introduction

Medical data are often small in quantity and hard to come by due to data privacy reasons. For COVID-19 Chest X-ray analysis, specialized medical professionals such as a radiologist must physically and manually inspect the chest X-rays. As such, such medical data is costly and time-consuming to produce. Machine learning and deep learning models are excellent at solving computer vision problems such as image classification. However, the training models usually overfit a small dataset. Data augmentation is one of the most commonly used methods to aid deep learning models and prevent overfitting. Data augmentation is a valuable technique to enlarge the training dataset and provide more training data for the deep learning models. Data augmented using GANs allows new data that follows closely to the original distribution of the training data without adding too much noise. In this paper, we provide a deep learning approach to data augmentation through the application of Generative Adversarial Networks (GANs). In Section 2, we introduce related work in the generation of synthetic X-ray images using Generative Adversarial Networks (GANs), followed by Section 3, where we introduce our methodology. In Section 4, we show the results and make comparisons. Lastly, we draw our conclusions in Section 5 and provide some suggestions for future work in Section 6.

2. Related Work

Generative Adversarial Networks (GAN) is a framework that enables deep learning models to learn the data distribution from the training data to produce realistic synthetic data. Goodfella, I., J.; et al. [1] introduced the Generative Adversarial Network (GAN). A typical GAN framework consists of two separate deep learning models: a generator and a discriminator. In the GAN architecture, the generator's main objective was to produce synthetic data to fool the discriminator into thinking that the synthetic data was the actual data by learning the distribution of the training data. At the same time, the discriminator's main objective was to learn and distinguish between the actual training data and the generated data by the generator. Data augmentation is a technique to modify the data slightly by adding noise to increase the training dataset for deep learning models. In the following section, we present some relevant works on using Generative Adversarial Networks (GAN) to generate synthetic chest images. Researchers from Qassim University explored using GAN to expand all classes of chest X-rays with respiratory issues to achieve a more balanced dataset, as the number of typical chest X-ray images was significantly higher than that of those with respiratory issues. The study suggested that augmented data can increase the accuracy of image classifiers [2].

Further, researchers from Harrisburg University of Science and Technology explored using DCGAN to generate chest X-ray images, where the generated chest X-ray images achieved an Fréchet Inception Distance (FID) score of 1.289 and a CNN image classifier trained on the generated images achieved an accuracy of 95.5%, a 3.5% improvement over traditional data augmentation techniques [3].

Later, Sharmila, V.J., [4] explored the use of DCGAN to generate synthetic chest X-ray images, allowing the Convolutional Neural Network (CNN) image classifier to achieve higher accuracy than the pre- trained image classifier. In addition, researchers from Delhi Technological University explored using Deep Convolutional Generative Adversarial Networks (DCGAN) to generate COVID-19-positive chest CT images. They found that validating the COVID-19 chest images was challenging, as they needed a medical professional to validate them manually. Instead, they validated the generated COVID-19 chest images using a pre-trained DenseNet169 model, achieving an accuracy of 40% higher than the baseline model [5].

Later, researchers from the Massachusetts Institute of Technology [6] explored conditional GANs for data augmentation in chest X-ray classification. In their work, they found that the data augmentation with GAN provided the best improvement when applied to small training datasets, and the data augmentation from GAN even gave no or negative improvement for large training datasets. Rehman, N.U., et al. [7] used a combined dataset of X-ray modalities for chest disease detection by including a COVID-19 X-ray imaging dataset. To normalize the data, data augmentation was performed. This preprocessed data removed the data bias. To make the proposed study more promising, Rehman, N.U., et al. [7] used two methods of validation (5- and 10-fold). They observed from the results that more folding made the results more accurate. This means that the proposed study can be used to detect multi-class chest disease.

Often the images in databases have low contrast; therefore, Khan, M.A.; et al. [8] implemented a new hybrid method, and the contrast was improved. Improvement of the contrasts plays a key role in obtaining useful features. Khan, M.A., et al. [8] developed a new fully automated deep learning feature fusion-based method for the classification of chest CT images originating from COVID-19-infected and healthy subjects.

In the paper by Venu & Ravula, [3], they suggested one potential way to improve the DCGAN image quality was to replace the binary cross-entropy loss function in the DCGAN with Wasserstein's loss with gradient penalty. As such, in this paper, we synthesize chest X-ray images using the DCGAN and the WGAN-GP and compare the performances of the two different GANs using the FID score from the images generated. Later, Allioui, H., et al. [9] proposed a unique way to minimize human effort in the extraction of medical image masks by introducing an advanced version of the Deep Q-Network (DQN) architecture. As

such, Allioui, H., et al. [9] enhanced deep reinforcement learning to select optimal masks during the segmentation of medical images and provided a robust strategy for semantic segmentation that uses a deep reinforcement learning model.

3. Methodology

Goodfellow, I.J.; et al. [1] describes in their paper that the GAN model reaches equilibrium when the generator can produce synthetic data that looks completely real while the discriminator achieves an accuracy of 50% in distinguishing between actual and fake data.

$$min_G max_D V(D,\ G) = E_{X\ \sim\ \mathbb{P}_{data(x)}}[logD(x)] + E_{Z\ \sim\ \mathbb{P}_z(z)}[log\ (1 - D(G(Z)))] \quad (1)$$

The GAN model's objective function is described in Equation (1). $E_{X\ \sim\ \mathbb{P}_{data(x)}}$ represents the expected value of all the real instances from the training data and $E_{Z\ \sim\ \mathbb{P}(z)}$ is the expected value of all the fake instances from the generator. $\mathbb{P}_Z(Z)$ is the noise data sampled from a standard Gaussian distribution, $G(Z)$ represents the noise data being mapped to the distribution of the training data x by the generator. $D(x)$ is the probability of the discriminator correctly identifying the training data as real while $D(G(Z))$ represents the probability of the discriminator identifying the generated data as real data.

During training, the GAN model will aim to maximize $logD(x)$ which is the logarithm of the $D(x)$'s probability of correctly identifying the correct labels for both the real training data and the data generated by the $G(x)$. Simultaneously, the GAN model will also minimize $1-logD(G(x))$ which is the loss for the generator for being correctly identified by the $D(x)$ as fake. During training, the generator takes a random input and maps it to the data space of the actual data. The input passes through a series of convolutional-transpose layers in the neural network to generate a fake image resembling the training data. During training, the discriminator acts as a binary classifier as it tries to distinguish between real and fake data. The discriminator takes in the generated data as input and passes the data through a series of convolution layers in the neural network to generate a scalar probability of the data being real or fake.

3.1. Data

The COVID-19 positive images used to train the GAN models came from the COVID-19-radiography database from Kaggle, [10,11]. The repository consists of 3616 COVID-19 positive images, 10,192 normal lung images, 6012 non-COVID lung infection images, and 1345 viral pneumonia images. All images were resized to 299 × 299. In this paper, we will only be using the COVID-19 positive chest X-ray images to train the GAN model, as our objective is to generate COVID-19 chest X-ray images. Examples of COVID-19 positive chest X-ray images is shown in Figure 1 below.

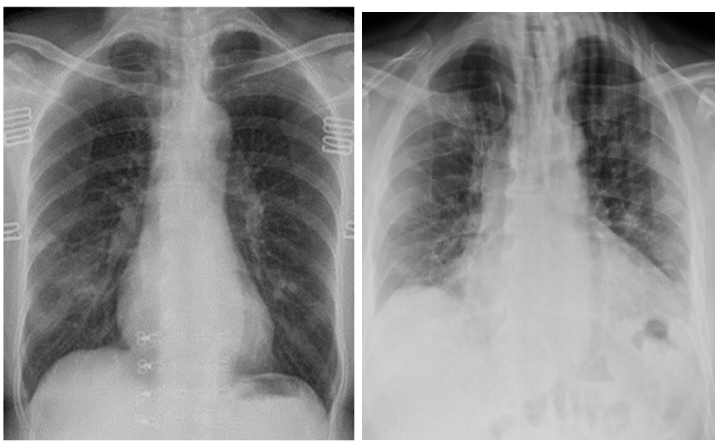

Figure 1. COVID-19 Positive X-ray Images.

3.2. Image Pre-Processing

The input passes through a series of convolutional transpose layers in the neural Contrast Limited Adaptive. Histogram Equalization (CLAHE) is used to pre-process the images before training the GAN models. We used the OpenCV python library [12] to apply CLAHE to the original training data, following which the COVID-19 chest X-ray images will be resized from 299 × 299 to 64 × 64 and normalized from having pixel values between [0, 255] to [−1, 1] before training the GAN models.

The application of the CLAHE technique to the chest X-ray images significantly increases the contrast between the lungs and the non-lung regions of the image. The increase in contrast was useful as it made the lung region more distinct from the non-lung region. As such, we trained our GAN models with images pre-processed using CLAHE.

3.3. Deep Convolutional Generative Adversarial Network

Deep Convolutional Generative Adversarial Networks (DCGAN) are an extension of the GAN model. DCGAN was first introduced by Radford, A., et al. [13]. In his paper, the DCGAN architecture explicitly uses convolutional-transpose layers in the generator and convolutional layers in the discriminator. In the generator, each convolutional-transpose layer is followed by a batch norm layer and a ReLU layer, while a LeakyReLU layer follows the convolution layer in the discriminator. The last layer in the discriminator uses a sigmoid layer to predict the probability that the image is real or fake.

Following the paper on DCGAN by Radford, A., et al. [13], the paper on WGAN-GP by Gulrajani, I., et al. [14], and the tutorial provided by PyTorch (Inkawhich), both the DCGAN and WGAN-GP had the same architecture for their generator and discriminator except for the loss function, where the DCGAN uses a binary cross-entropy loss function whereas the WGAN-GP uses the Wasserstein's distance with gradient penalty as its loss function.

For the architecture of the generator of the DCGAN, the generator takes in a 100 × 1 noise vector into the input layer. Following the input layer are five convolution-transpose layers to upsample the input. Next, each convolution-transpose layer was followed by a batch norm layer and a ReLU layer, except for the last layer. The last layer, the output layer, has a convolution-transpose layer followed by a Tanh layer that returns a 64 × 64 × 3 vector with values between [−1, 1].

The input layer in the discriminator took in a 3 × 64 × 64 vector generated by the generator. Following the input, a layer is composed of five convolution layers. Each convolution layer is followed by a LeakyReLu layer with a slope of 0.2 except for the last layer. The last layer, the output layer, is a convolution layer followed by a sigmoid layer that outputs the probability of the image generated being real or fake.

The DCGAN model uses the binary cross-entropy loss function to calculate the loss for both the generator and the discriminator.

3.4. Wasserstein Generative Adversarial Networks with Gradient Penalty (WGAN-GP)

WGAN-GP is also an extension of the original GAN mentioned in Goodfellow, I.J., et al. [1]. WGAN-GP was first mentioned in Gulrajani, I., et al. [14]. In Gulrajani, I., et al. [14], the WGAN-GP uses a different loss function for both the generator and discriminator loss as compared with the DCGAN described in Radford, A., et al. [13]. In the WGAN-GP architecture, the discriminator, called the critic, omits the sigmoid layer in the last layer. Instead of finding the probability that the input data given to the critic is real or fake, the WGAN-GP determines the score of the input data using Wasserstein's distance with gradient penalty.

The WGAN-GP closely follows the architecture of the DCGAN for its generator and discriminator. Except that the InstanceNorm layer was used instead of a batch norm in the discriminator to allow the critic's penalty to work as planned. Also, the discriminator omits the sigmoid layer in the output layer as it does not calculate the probability of the input data being real or fake. The WGAN-GP uses Wasserstein's distance with gradient penalty as opposed to the binary cross-entropy loss for calculating its loss. As opposed to

DCGAN, where the discriminator calculates the probability of each image if it is real or fake, the WGAN-GP utilizes the discriminator called critic to give each image a score. The score which is used as the realness of the data is calculated using the Wasserstein's distance with gradient penalty.

$$\text{Wasserstein's distance with gradient penalty} = E_{\hat{x}} \sim \mathbb{P}_g[D(\hat{x})] - E_x \sim \mathbb{P}_r[D(x)] + \lambda E_{\tilde{x}} \sim \underset{\mathbb{P}_x}{\sim}[(||\nabla_{\tilde{x}} D(\tilde{x})||_2 - 1)]^2 \quad (2)$$

In the Wasserstein's distance with gradient penalty, D represents the set of 1-Lipschitz functions. $E_{\hat{x}} \sim \mathbb{P}_g$ refers to the expected value of all the fake instances generated by the generator, $E_x \sim \mathbb{P}_r$ refers to the expected value of all the real instances from the training sample. $E_{\hat{x}} \sim \mathbb{P}_g[D(\hat{x})] - E_x \sim \mathbb{P}_r[D(x)]$ refers to the Wasserstein's distance between the generated data and the actual training data. $\lambda\, E_{\tilde{x}} \sim \underset{\mathbb{P}_x}{\sim}[(||\nabla_{\tilde{x}} D(\tilde{x})||_2 - 1)]^2$ refers to the gradient penalty used to enforce the 1-Lipschitz space for the critic of the WGAN-GP model. The λ represents the gradient penalty coefficient, $\mathbb{P}_{\tilde{x}}$ represents the distribution obtained by uniformly sampling along a straight line between the real and generated distributions $\mathbb{P}_r$ and $\mathbb{P}_g$.

3.5. Experiment Methodology with GANs for Image Generation

We conducted three different experiments to understand how GAN works. In the first experiment, we trained the DCGAN model with four datasets of different sizes. In the second experiment, we tune the epoch hyperparameters using one of the four datasets chosen from the first experiment. In the last experiment, we compared the performance of the images generated by DCGAN and the WGAN-GP using the results from the first two experiments.

As specified in the DCGAN paper by Radford, A., et al. [13], the DCGAN uses the Adam optimizer with a learning rate of 0.0002 and a Beta1 of 0.5. Compared with the WGAN-GP paper from Gulrajani, I., et al. [14], the WGAN-GP uses an Adam optimizer with a learning rate of 0.0001, Beta1 of 0.5 and Beta2 of 0.9.

While GANs are capable of generating fake data that closely resembles the actual training data, it is hard to evaluate the performance of GANs. One of the methods to evaluate GANs is through the use of the Fréchet Inception Distance (FID) score, first introduced in Heusel, M., et al. [15]. In his paper, he introduced the FID score as a metric to measure how the GAN model is performing by comparing the generated images to the real training images.

The FID score measures the quality of the images generated by the GAN by measuring the Fréchet distance between the generated images and the actual training images. The FID score is calculated based on the distribution of the real images used to train the GAN model against the synthetic images generated by the GAN model.

The FID score is calculated based on the distribution of the real images used to train the GAN model against the synthetic images generated by the GAN model. The FID metric measures the Fréchet distance between two multidimensional Gaussian distributions, Xg and Xr obtained from the 2048-dimensional activations of the InceptionV3 pooling layer for the real images and generated images, respectively.

As a rule of thumb, the lower the FID score, the better the GAN is performing, as the images generated by the GAN model are closer to the training images in terms of the Fréchet distance between them.

4. Comparison and Results of the DCGAN and the WGAN-GP

4.1. Results of the DCGAN

The first experiment was conducted to understand how the different sizes of the training datasets affect the quality of images generated by the DCGAN model. In the experiment, we generated 1000 synthetic COVID-19 positive chest X-ray images from the DCGAN models in [16,17] using four different datasets of varying sizes. The datasets

contain 500, 1000, 1500, and 2000 actual COVID-19 images, respectively. Each of the DCGANs were trained with 500 epochs and 128 batch sizes in order to obtain a fair comparison between the DCGANs trained on different datasets. We obtained the FID score for each of the four DCGAN models by comparing the 1000 images generated to the training data used to train the DCGAN model. The FID score for the 1000 images generated was calculated using the Pytorch-FID package from a public Github repository in Seitzer, M. [18] since we only generated 1000 images, a sample much smaller than we used the 768 as the dimension number instead. With a sample size smaller than 2048, we will not be able to take advantage of the InceptionV3 pool-layer [6,17,19] as such, we will use the previous layer, which is a Pre-aux classifier with 768-dimensional features.

In Table 1 below, we display the FID score for all datasets. All the FID scores are below 2, which means that the images generated by the DCGAN are similar to the original training images, indicating that the DCGAN performed well in generating synthetic X-ray Chest Images.

Table 1. FID Score for DCGAN Trained with Different Datasets.

Dataset Size (Number of Actual COVID-19 Positive Images)	FID Score of the Chest X-ray Images Generated
500	1.763
1000	1.494
1500	1.405
2000	1.249

We notice that the FID scores for images generated by DCGAN trained on 1000 and 1500 images are relatively close, with only a slight difference of 0.089. This shows that images generated by the DCGAN trained on 1000 images performs quite closely to the images from DCGAN trained with 1500 images. This demonstrates the capabilities of DCGAN to generate high-quality synthetic data despite having a smaller training dataset.

We can see in Figure 2 below, that at 50 epochs, the DCGAN model is able to generate images that resemble chest X-ray images, although the images are very pixelated and blurry.

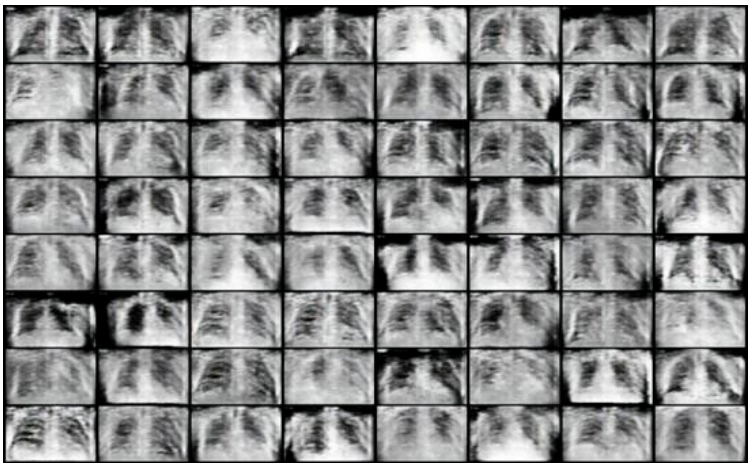

Figure 2. Generated COVID-19 positive chest X-ray images with dataset of 1000 images at 50 epochs.

In Figure 3 below, we observed that the images at 500 epochs produced by the DCGAN are now less blurry as compared with the images in Figure 2 and have a much higher resemblance to the COVID-19 chest X-ray images used to train the DCGAN model. Thus, showing that the DCGAN model improves and generates higher-quality images as training progresses.

Figure 3. Generated COVID-19 positive chest X-ray images with dataset of 1000 images at 500 epochs.

We observed in Figure 4 below that the GAN generator needed some time to 'warm up' and that the generator experienced a high loss during the starting iterations. As the generator continues to learn, the loss gradually reduced. However, the generator is still rather unstable, as observed in Figure 4, where the generator's loss is still fluctuating.

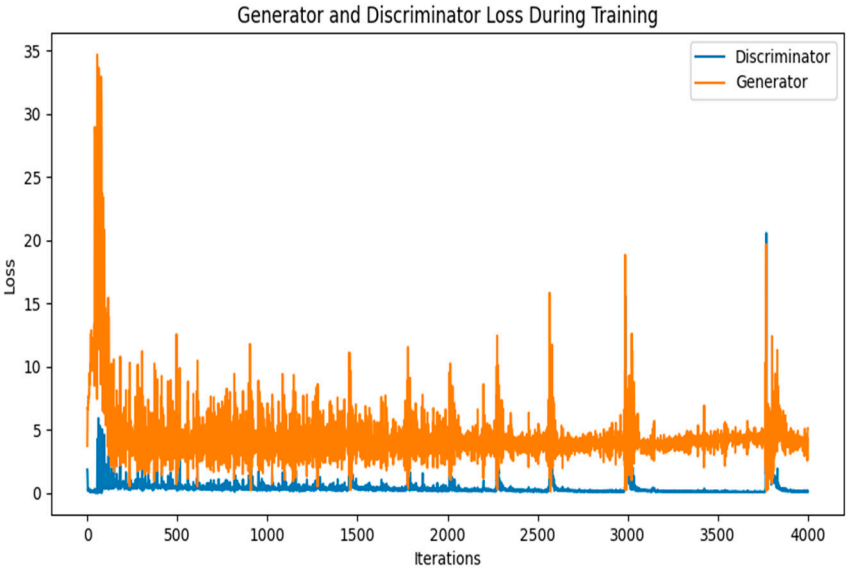

Figure 4. DCGAN's Generator and Discriminator loss trained with dataset of 1000 images.

The number of epochs used for training a model is a hyperparameter that directly affects the quality of the deep learning model, as the number of epochs affects how the model learns. The epoch hyperparameter is the number of times each training sample goes through a complete pass in the entire neural network. A complete pass refers to both the forward pass and the backward propagation.

Due to the limited computational resources and time needed to train the DCGAN on a larger dataset, we focused on using the 1000 images dataset and batch sizes of 128 to train the DCGAN and fine-tune the epochs hyperparameter to obtain the best-performing DCGAN.

To strike a balance between finding the optimal epoch number to train the GAN model and managing limited computational resources, we trained the DCGAN with 1000 images as the training data up to 800 epochs. At every 100th epoch, starting from the 500th epoch, we generated 1000 COVID-19 chest X-ray images to evaluate the performance of the DCGAN at the current epoch using the FID score.

From Table 2, we observed that training the DCGAN longer may not result in better performance due to the instability of the generator. The images generated by the DCGAN have shown to improve in quality, as the FID score decreases from 1.451 at 500 epochs to 1.415 at 600 epochs and 1.399 at 700 epochs. However, at 800 epochs, the images generated by the DCGAN perform worse as the FID of the images generated increases to 1.640. This indicates that training longer does not equate to better performance for the DCGAN due to the instability of the generator during training. From Table 2, we know that training the DCGAN with 700 epochs will give us the best performing DCGAN model.

Table 2. FID Score of DCGAN at Different Epochs.

Epoch	FID Score
500	1.451
600	1.415
700	1.399
800	1.640

4.2. Results for the WGAN-GP

We trained the WGAN-GP model [20] using 1000 COVID-19 chest X-ray images as the training data. The hyperparameters used to train the WGAN-GP model are batch size 128 and epoch 700. Using this set of requirements ensured that the comparison between the DCGAN and the WGAN-GP was fair.

From Figure 5 below, we see that at 700 epochs, the images generated by the WGAN-GP model show high resemblance to chest X-ray images, but they are still pixelated. The WGAN-GP Generator and Critic Loss were trained with a dataset of 1000 images, and the WGAN-GP's loss was stable, as it did not fluctuate randomly throughout the training.

Figure 5. Generated COVID-19 positive chest X-ray images with dataset of 1000 images using WGAN-GP at 700 epochs.

4.3. Comparison of the DCGAN and WGAN-GP

We used the FID score of the images generated by the two GAN models to compare their performance. To allow for model comparisons, the two GAN models, DCGAN and WGAN-GP, were trained with the same dataset consisting of 1000 COVID-19 chest X-ray images and the hyperparameters batch size 128 and epoch 700. Table 3 displays the FID scores of the two GAN models.

Table 3. FID for Images Generated by DCGAN and WPGAN-GP.

GAN Type	FID Score
DCGAN	1.399
WGAN-GP	1.583

From Table 3 above, we found that, during training at batch size 128 and epoch 700, the WGAN-GP performed worse than the DCGAN, as the images generated by the DCGAN had a better FID score than the images generated by the WGAN-GP.

Table 4 showcases how the InceptionV3 COVID-19 image classifier performs with the addition of augmented data from the GANs.

Table 4. Model Accuracy Metrics.

Dataset	Accuracy	Precision	Recall	F1 Score
Only COVID-19 positive	0.96	0.94	0.97	0.95
COVID-19 positive + images generated from DCGAN	0.98	0.97	1	0.98
COVID-19 positive + images generated from WGAN-GP	0.98	0.99	0.98	0.98

From Table 4 above, we observed that without the augmented COVID-19 chest X-ray images from the GANs, the InceptionV3 COVID-19 image classifier already has pretty good performance with an accuracy of 0.96. As such, the inclusion of augmented data from the GAN model only manages to improve the performance of the InceptionV3 COVID-19 image classifier slightly, from 0.96 to 0.98. Even though the increase in performance is not very significant, it signifies that the InceptionV3 COVID-19 image classifier is now more robust as it has been trained on a larger dataset.

We also observed from Table 4 that, overall, the precision, recall, and F1-Score metrics have shown slight improvement with the addition of data augmented by the GAN models. From this experiment, we can conclude that the addition of augmented data from the GAN models is indeed useful in improving the performance of the InceptionV3 COVID-19 image classifier.

5. Limitations of GANs

Even though GANs are very valuable for generating more data and X-ray images, there are limitations to using GANs. Firstly, there's still no intrinsic metric evaluation present for better model training and generating complex outputs. In addition, for density estimation, we still cannot predict the accuracy of the density of the evaluated model and state that this image is dense enough to move forward with. The evaluation of the GANs that are generating the data is being decided manually. GANs are elegant mechanisms for data generation, but due to unstable training and unsupervised learning, it is difficult to train and generate output.

6. Conclusions

We have demonstrated that GANs can successfully generate COVID-19 chest X-ray images. We generated 1000 synthetic COVID-19 chest X-rays images with FID scores all much lower than 2. We also found that it is not always necessary to train the GAN model

with a larger dataset to achieve better results, as we found that the DCGAN trained with 1000 images performed relatively similarly to the DCGAN trained on 1500 images. Further, due to how unstable the generator of the DCGAN is during training, training the DCGAN longer may not lead to better results, as the DCGAN performed much better at 700 epochs than at 800 epochs.

In addition, we also found that while WGAN-GP has a more stable training regimen than DCGAN, it takes WGAN-GP much longer to achieve the same performance as the DCGAN. In conclusion, GANs can avoid the problems that traditional data augmentation methods cannot avoid, which is to generate new image data from the training data that follows the same distribution as the training data without adding too much noise. Moreover, GANs can generate high-quality and realistic image data even when trained on a small dataset, which is undoubtedly very useful when training data and computational resources are limited. When trained on a small dataset, deep learning models are often overfit. With additional data generated from the GAN models, the problem of overfitting can be overcome.

7. Future Work

A possible future study would be to generate images that are larger than 64 × 64 (Height × Width) and compare how the DCGAN and WGAN-GP would fare. In this paper, we only managed to explore training the DCGAN up until the 800 epoch; in a future study, we could explore training the DCGAN using different batch sizes as well. In future iterations of this experiment, instead of focusing on using 1000 images to train the GAN model, we could run the same experiment again but with a focus on a larger training dataset.

Author Contributions: M.F.N.: Performed the formal analysis for this study using Python Programming. Prepared the original draft research paper. Jointly decided on the methodology for the analysis. Agreed to the published paper version. C.A.H.: Mentor—Supervised the project. Determined the approach for the study analysis. Reviewed the original draft paper and prepared the final paper version using the journal template. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable as open source public data was used.

Informed Consent Statement: Not applicable as open source public data was used.

Data Availability Statement: Data came from COVID-19 radiography database from Kaggle. https://www.kaggle.com/c/rsna-pneumonia-detection-challenge/data (accessed on 4 March 2022).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Networks. *arXiv* **2014**, arXiv:1406.2661. [CrossRef]
2. Albahli, S. Efficient GAN-Based Chest Radiographs (CXR) Augmentation to Diagnose coronavirus Disease Pneumonia. Available online: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7330663/ (accessed on 27 February 2022).
3. Venu, S.K.; Ravula, S. Evaluation of Deep Convolutional Generative Adversarial Networks for Data Augmentation of Chest X-ray Images. Available online: https://www.mdpi.com/1999-5903/13/1/8 (accessed on 1 March 2022).
4. Sharmila, V.J. Deep Learning Algorithm for COVID-19 Classification Using Chest X-ray Images. NCBI. Available online: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8594989/ (accessed on 4 March 2022).
5. Mann, P.; Jain, S.; Mittal, S.; Bhat, A. Generation of COVID-19 Chest CT Scan Images using Generative Adversarial Networks. *arXiv* **2021**, arXiv:2105.11241.
6. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. *arXiv* **2015**, arXiv:1512.00567.
7. Rehman, N.U.; Zia, M.S.; Meraj, T.; Rauf, H.T.; Damaševičius, R.; El-Sherbeeny, A.M.; El-Meligy, M.A. A self-activated CNN approach for multi-class chest-related COVID-19 detection. *Appl. Sci.* **2021**, *11*, 9023. [CrossRef]

8. Khan, M.A.; Alhaisoni, M.; Tariq, U.; Hussain, N.; Majid, A.; Damaševičius, R.; Maskeliūnas, R. COVID-19 case recognition from chest CT images by deep learning, entropy-controlled firefly optimization, and parallel feature fusion. *Sensors* **2021**, *21*, 7286. [CrossRef] [PubMed]
9. Allioui, H.; Mohammed, M.A.; Benameur, N.; Al-Khateeb, B.; Abdulkareem, K.H.; Garcia-Zapirain, B.; Damaševičius, R.; Maskeliūnas, R. A multi-agent deep reinforcement learning approach for enhancement of COVID-19 CT image segmentation. *J. Pers. Med.* **2022**, *12*, 309. [CrossRef] [PubMed]
10. Chowdhury, M.E.; Rahman, T.; Khandakar, A.; Mazhar, R.; Kadir, M.A.; Mahbub, Z.B.; Islam, K.R.; Khan, M.S.; Iqbal, A.; Emadi, N.A.; et al. Can AI help in screening viral and COVID-19 pneumonia? *IEEE Access* **2020**, *8*, 132665–132676. [CrossRef]
11. Rahman, T.; Khandakar, A.; Qiblawey, Y.; Tahir, A.; Kiranyaz, S.; Abul Kashem, S.B.; Islam, M.T.; Al Maadeed, S.; Zughaier, S.M.; Khan, M.S.; et al. Exploring the effect of image enhancement techniques on COVID-19 detection using chest X- ray images. *Comput. Biol. Med.* **2021**, *132*, 104319. [CrossRef] [PubMed]
12. Histograms—2: Histogram Equalization. OpenCV. Available online: https://docs.opencv.org/3.4/d5/daf/tutorial_py_histogram_equalization.html (accessed on 4 March 2022).
13. Radford, A.; Metz, L.; Chintala, S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *arXiv* **2015**, arXiv:1511.06434.
14. Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; Courville, A. Improved Training of Wasserstein GANs. *arXiv* **2017**, arXiv:1704.00028.
15. Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; Hochreiter, S. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. *arXiv* **2017**, arXiv:1706.08500.
16. DCGAN Tutorial. DCGAN Tutorial—PyTorch Tutorials 1.10.1+cu102 Documentation. Available online: https://pytorch.org/tutorials/beginner/dcgan_faces_tutorial.html (accessed on 4 March 2022).
17. Tejanirla. Image_classification/transfer_learning.ipynb at Master Tejanirla/Image_Classification. GitHub. Available online: https://github.com/tejanirla/image_classification/blob/master/transfer_learning.ipynb (accessed on 4 March 2022).
18. Seitzer, M. pytorch-fid: FID Score for PyTorch. Opgehaal van. Available online: https://github.com/mseitzer/pytorch-fid (accessed on 4 March 2022).
19. Keras Documentation: Inceptionv3. Keras. Available online: https://keras.io/api/applications/inceptionv3/ (accessed on 4 March 2022).
20. Aladdinpersson. (n.d.). Machine-Learning- Collection/ml/pytorch/gans/4.WGAN-GP Aladdinpersson/Machine-Learning-Collection. GitHub. Available online: https://github.com/aladdinpersson/Machine-Learning-Collection (accessed on 4 March 2022).