Conditional GAN - NIH Chest X-Ray

Brendon Ng [*1] **Jivan Gubbi** [*1]

Abstract

Generative Adversarial Networks allow for the creation of synthetic data that is ideally indistinguishable from training data. This provides various improvements over real data, such as reducing the cost of data collection, expanding sparse data sets, and mitigating privacy concerns.

In the case of medical imaging, these problems are heightened, as data collection and annotation often requires buying expensive machinery and labeling data must be done by qualified professionals. In addition to this, patient privacy laws are stringent, often leading to sparse data sets for any particular condition.

We propose a Conditional GAN for Chest X-Rays to mitigate this problem for a variety of diseases that can be detected through X-Ray images. We trained our model using a data set collected by the NIH on 6 different disease classifications. We found promising results that show that our model is capable of creating realistic X-Ray images and given more time and compute resources, we are confident that we can expand this project to handle more detailed images and a wider variety of diseases.

Link to code:
https://github.com/brendon-ng/Chest-XRay-Conditional-GAN

1. Introduction

1.1. Background

X-ray scans are expensive, costing hundreds of dollars to operate and label per scan. Because of this obtaining data sets large enough to train classification models has significant financial barriers. While there exist publicly available data sets for a variety of diseases, the size of each data set is often limited to a few hundred images. Because of this, training machine learning models for classification is difficult. While adoption of machine learning models within clinical settings has been slow, it is promising that classifiers on larger data sets have been able to perform with high accuracy[1]. The main missing piece from getting more of these accurate models in a clinical setting is supplying sufficient data. This is where synthetic data set expansion comes into play. GANs have the ability to expand a data set which leads to better downstream performance of models trained on augmented data[2].

1.2. Problem Statement

Given the NIH X-Ray data set, which contains about 112,000 labeled X-Ray images, we trained two different GAN models on a subset of the data and labels. The first was a traditional DCGAN model while the second was conditional GAN. We propose the use of a Conditional GAN which allows for the specification of a label when generating new data. This leads to pre-labeled data images for downstream tasks at a fraction of the cost.

2. Literature Review

2.1. Data Set Augmentation

Data set augmentation can come in many forms. Among the most common techniques are transformations, kernel filters, random erasing and generative models to name a few. The purpose of these is to improve the performance of deep networks and reduce over-fitting in the case of limited data. Shorten and Khoshgoftaar found that utilizing GANs for data augmentation allowed for improvement of classifier performance by roughly 3%-10%.[3]

In the specific case of Chest X-Ray data augmentation, which is what we will be discussing in this paper, researchers compared performance across different data set sizes and amount of generated data used. It was found that at small data set sizes, there is an increase of performance of on average about 4% for GAN based augmentation in comparison to standard augmentation techniques.[4]

[1]UCLA Computer Science. Correspondence to: Brendon Ng <brendonn8@ucla.edu>, Jivan Gubbi <jcgubbi@ucla.edu>.

UCLA Computer Science, Course CS260C, Winter Quarter, 2022.
Copyright 2022 by the author(s).

[1](Baltruschat et al., 2019)
[2](Sundaram & Hulkund, 2021)
[3](Shorten & Khoshgoftaar, 2019)
[4](Sundaram & Hulkund, 2021)

2.2. Generative Adversarial Networks (GAN)

The first GAN was proposed in 2014 as a novel method of training two models in an adversarial manner. The generator estimates the data in the training set and the discriminator predicts whether given data originated from the generator distribution or from the real distribution. The authors assert that the process of pitting the two models against one another drives a competition that improves the generators ability to create counterfeit data.[5]

To learn the generator distribution, the generator learns a mapping function from a latent space noise distribution of arbitrary dimension or shape to a data space in the size and dimensionality of the desired output image. The discriminator maps from the input image to a single label of real or fake. The objective function of a GAN is a two-player min-max game. Given the generator G and the discriminator D, the game has the value function $V(G, D)$: [6]

$$\min_G \max_D V(D,G) = E_{x\ p_{data}(x)}[logD(x)] + E_{x\ p_x(z)}[log(1 - D(G(z)))] \quad (1)$$

The main improvement that this adversarial model provides over traditional generative models is its lack of reliance on Markov chains. Using the value function above, we can train our model using simple back-propagation to get gradients. The downside to this approach is that we have no control over which subsection of our data that the generated data comes from. The associated label of the data is unknown and out of our control.

2.3. Conditional Generative Adversarial Networks (cGAN)

Conditional General Adversarial Networks (cGAN) do not differ largely from GANs themselves. GANs can be extended to become a conditional model by conditioning the generator and discriminator on extra information to direct the generation process. This conditioning can be performed by simply adding another input layer. [7] With conditional GANs, we are now trying to learn the probability distribution of the image to be generated *given* the label for the generator. For the discriminator, it is trying to learn how to decipher real or fake images *given* the label. For example, a GAN trained to generate images of the handwritten digits learnt from the MNIST data set can be extended to be conditioned on the number value. The cGAN would then be able to generate a specified handwritten digit from zero to nine.

The best practice way to condition a generator and discriminator is to utilize an embedding layer of either a one-hot label vector or the label value itself along with a fully connected layer that will scale to the size of the image. This scaled image-sized embedding would then be concatenated on the model as an additional channel.[8] For the generator, two inputs would be taken by the model: one for the latent noise vector and one for the input label. The latent vector is scaled up into a multi-channel image of increasing size. Meanwhile, the embedding of the input label is fed into a fully connected layer and reshaped to be concatenated as additional channel(s) to the latent input somewhere along its upscaling. For the discriminator, the model takes in the input image and the input label. The input label embedding is fed into a fully connected layer and reshaped to become another channel of the input image. This input image with the additional channel is then fed through the discriminator convolution layers and beyond to determine its real/fake label. The objective function of a conditional GAN is also a two-player min-max game. Given the generator G and the discriminator D, the game has the value function $V(G, D)$:[9]

$$\min_G \max_D V(D,G) = E_{x\ p_{data}(x)}[logD(x|\mathbf{y})] + E_{x\ p_x(z)}[log(1 - D(G(z|\mathbf{y})))] \quad (2)$$

Conditional GANs do not need to only be conditioned on discrete labels, but can also be conditioned on other inputs, such as an image for image-to-image translation tasks, or text embedding for text to speech applications.[10]

3. Data

3.1. Data Structure

We used data provided by the National Institute of Health which contains a variety of X-Ray images that are labeled with 14 different disease categories. The 14 diseases present in the sample are Atelectasis, Consolidation, Infiltration, Pneumothorax, Edema, Emphysema, Fibrosis, Effusion, Pneumonia, Pleural Thickening, Cardiomegaly, Nodule, Mass and Hernia. The data set contains 112,120 high resolution images that are from 30,805 unique patients. Each image can have many, or no, disease classifications and the labels were generated using a natural language processing text mining algorithm. This means that while the diagnoses were created by medical professionals, there is a chance of mislabeling when converting between their diagnosis and actual data labeling. This accuracy is expected to be greater than 90%. The size of each full size image is 1024x1024.

3.2. Pre-processing

While we wanted to use the full data set as provided, we had to take some steps to reduce the data size. The goal of

[5](Goodfellow et al., 2014)
[6](Mirza & Osindero, 2014)
[7](Mirza & Osindero, 2014)
[8](Denton et al., 2015)
[9](Mirza & Osindero, 2014)
[10](Radford et al., 2016)

this was to finish training within the time period that we had. This fell into three categories, first reducing the image size, second the number of images used and finally the number of diseases considered for the conditional GAN.

The image size was originally 1024x1024 but we reduced it to 128x128 in order to make the total number of parameters that we had to train more reasonable.

While the total number of images in the data set was roughly 112,000 images, we only used 10,000 training the traditional GAN and 15,000 for training the conditional GAN.

Finally, instead of considering the full 14 diseases, we only considered the 6 most prominent diseases. We also took out the comorbidities of diseases and just selected a single label for each image. There are strong correlations between certain diseases and there are many cases where a single image has multiple diseases. Instead of modeling this using a more complex classification system, which would make the learning task more difficult, we selected a single label per image.

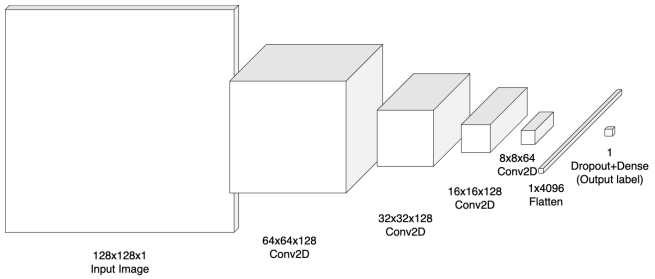

Figure 1. Normal GAN Discriminator Architecture

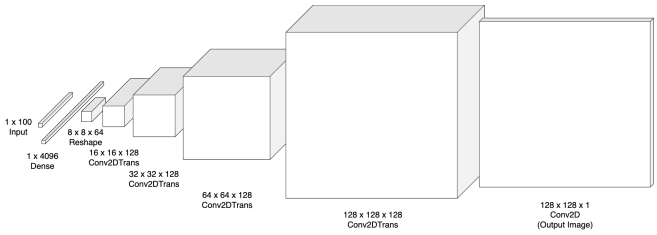

Figure 2. Normal GAN Generator Architecture

4. Model

Both GANs and cGANs are separated into two separate models: the generator and the discriminator. Both the discriminators and generators rely on convolution and convolution transpose layers respectively. Discriminators classify input 128x128 images as real or fake while generators produce an output 128x128 image by mapping from a 100-dimensional latent noise vector to the output image space. The GAN product is the generators, being able to generate images from randomly initialized latent space vectors and possibly an input label. The models are trained in an adversarial manner with the discriminator learning from real examples from the data set and fake examples from the generator. Meanwhile, the generator is training from random latent space inputs to generate outputs that are fed into the discriminator with the goal of tricking it, desiring a ”real” output from the discriminator. The ideal goal is the discriminator achieving 50% accuracy on generated images, meaning the generator produces indistinguishable images. Our normal GAN simply produces artificial images of chest x-rays from a random latent space vector, but our conditonal GAN takes in a random latent space vector in addition to an input label of which disease the desired output x-ray should have. The discriminator of the conditonal GAN determines if the image is real or fake given the disease label that it has.

Our models were built and trained in Python using Keras. The models for the non-conditoned GAN used Keras’ ’Sequential’ model functionality and the models for the conditional GAN used Keras’s ’Model’ functionality to be able to utilize multiple inputs. Code to define and train the model can be found in the project’s Github, linked in the abstract.

4.1. Model Architecture

4.1.1. GAN Architecture

Discriminator

The discriminator of the non-conditioned GAN is a binary convolutional neural network classifier, mapping from an input image to a single binary label of real or fake. The architecture of the GAN’s discriminator can be visualized in Figure 1. It takes in one 128x128 input image with one channel (black and white). It is then convolved using a stride of 2 and 128 5x5 kernels with Leaky ReLU to reduce the size of the image to a half sized 128-channel image. A total of four of these convolution layers are used to map the input image to an 8x8x128 image. The image is then flattened with a dropout of 0.4 before being fed into a fully connected layer with a single output channel with sigmoid activation.

Generator

The generator for the non-conditoned GAN learns a mapping from a 100-dimensional latent space of noise to an output 128x128 single-channel image. The architecture for the generator can be seen in Figure 2 The generator takes a single input of a 100x1 vector of noise and feeds it into a fully connected layer with enough output nodes to reshape into an 8x8x64 image. This image is then fed through convolution transpose layers with strides of 2 and 4x4 kernels with Leaky ReLU to upscale the image to a 16x16x128 image, then again three more times to upscale to a 128x128x128 image after four convolution transpose layers. This image

is then convolved with a single 16x16 kernel and a stride of 1 and padding and tanh activation to output a 128x128 single-channel image.

4.1.2. Conditional GAN Architecture

Discriminator

The discriminator of the conditioned GAN is very similar to that of the normal GAN in that it is a CNN binary classifier. However, the discriminator for our cGAN takes in an additional input of the class label of the image. The label input is an integer from 0 to 6 indicating which of the six diseases the patient in the input image has (or which disease the artificial image supposedly has) or the label 0 for no disease at all. The discriminator uses the 50-dimensional embedding of this label and feeds it into a fully connected layers with enough output nodes to fit the shape of the input image (128x128). The 16,384-dimensional vector is then reshaped to a 128x128 representation of the input label that is then concatenated to be a second channel of the input image. The two-channel input image/label is then fed through four consecutive convolution layers with strides of 2 and 125 5x5 kernels each with LeakyReLU to reduce the image size by half each time to finally be an 8x8x128 image. Like the normal GAN discriminator, this image is flattened with 0.4 dropout, then fed into a fully connected layer with sigmoid activation on a single binary output. Refer to Figure 4 for the conditional discriminator architecture.

Generator

The generator of the conditonal GAN is also similar to the normal GAN's generator, but like the conditional discriminator, it also takes in an additional input of the image label. The inputs to the conditonal generator are the 100-dimensional latent space vector and the single digit representing the class label. Similarly to the discriminator, the class label's 50-dimensional embedding is fed into a dense layer with enough output nodes to be reshaped into an 8x8x64 image. Similarly, the 100-dimensional input space is fed into a dense layer with its output nodes being reshaped into an 8x8x64 image. Both the representations of the input latent vector and the input label are concatenated together to form an 8x8x128 image. This image is then fed through four consecutive convolution transpose layers with strides of 2 and 128 4x4 kernels each with Leaky ReLU to upscale the image by a factor of two each time. The resulting 128x128x128 image is then fed through a convolution layer with stride of 1 a single 16x16 kernel with tanh activation to output the 128x128 single-channel image. See Figure 5.

4.2. Training

The two models within the GAN are trained in an adverarial manner. Each model's training depends on one another and they both improve together simultaneously. The discriminator is trained to decipher between real and fake given images from the real data set and from the generator. The generator produces images that are fed through the discriminator with the goal of "tricking" it, producing a "real" label designation.

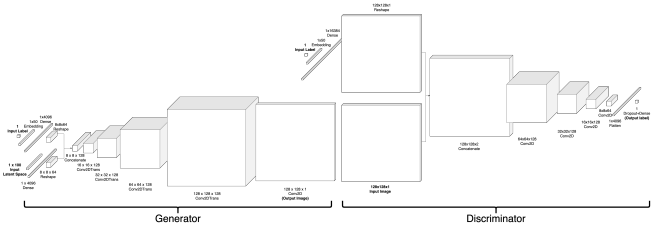

Figure 3. Combined Model for Generator Training

The GAN (both conditional and non-conditonal) as a whole is trained in a single training loop for 50 epochs. Within each epoch, the GAN is trained in batches of size 128. Each epoch has enough batches to train over the entire data set (i.e. dataset_size/batch_size batches per epoch). For each batch, the discriminator is trained on half a batch size worth of real samples from the actual data set, and half a batch size worth of fake images produced randomly by the generator from random input noise. Samples are taken randomly from the data set, so a random sample of labels are represented each time for real data, and for the conditional discriminator, the class labels are randomly drawn as well. The discriminator trains on the real batch with the expected label of "real" and trains on the fake batch with the expected label of "false". The generator is then trained in conjunction with the discriminator. A model in Keras is defined as a sequential model with the input going to the generator, and that generator's output being fed into the next "layer" of the sequential model which is the discriminator. The discriminator is set to be untrainable here as the purpose of this model is to train the generator. The architecture of this model can be seen in Figure 3. The combined model is trained on a full batch sized sample of random noise and random labels. These are passed in, the generator generates an image, and the discriminator (with set untrainable weights) classifies the image as real or fake. The goal of the generator here is to trick the discriminator, so expected output labels are set to "real". Since the discriminator weights are set to be untrainable, training the combined model in this manner allows the generator weights to be trained and updated with the goal of fooling the discriminator.

Out of the data set, x-rays with multiple diseases were ignored for the sake of labeling. All together, the model was trained on approximately 50,000 images of relatively uniform distribution across the seven disease classes.

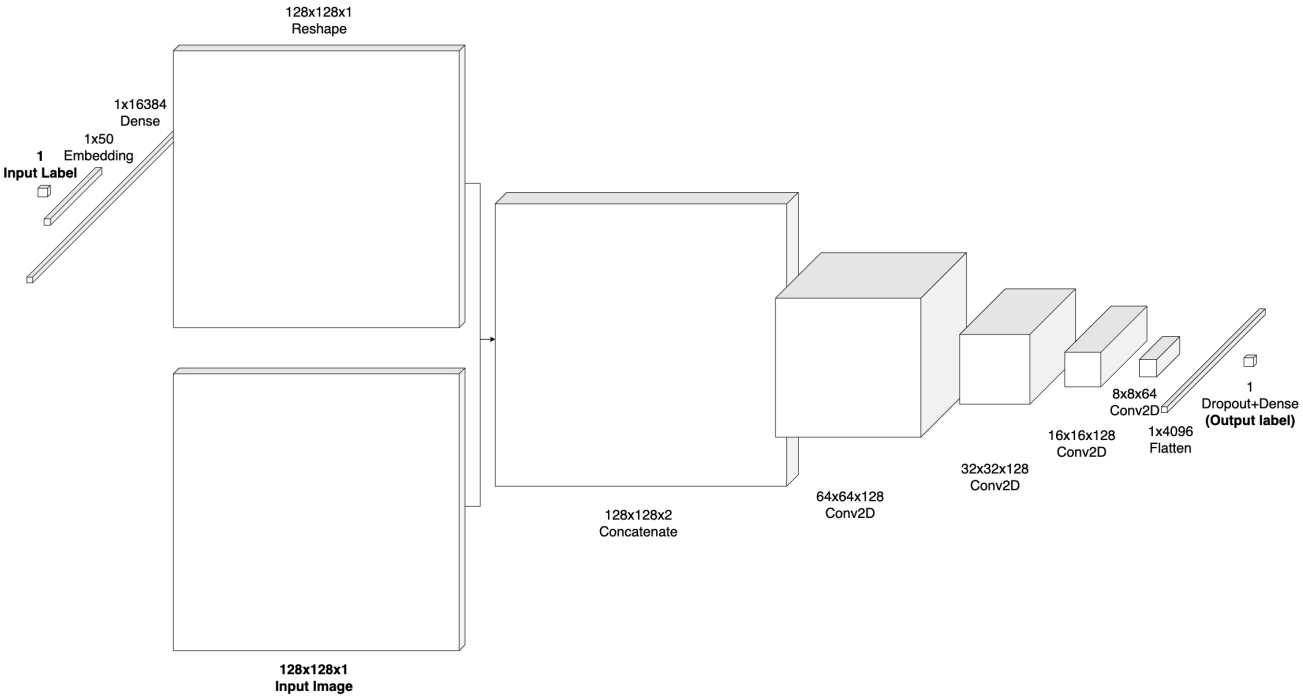

Figure 4. Conditional GAN Discriminator Architecture

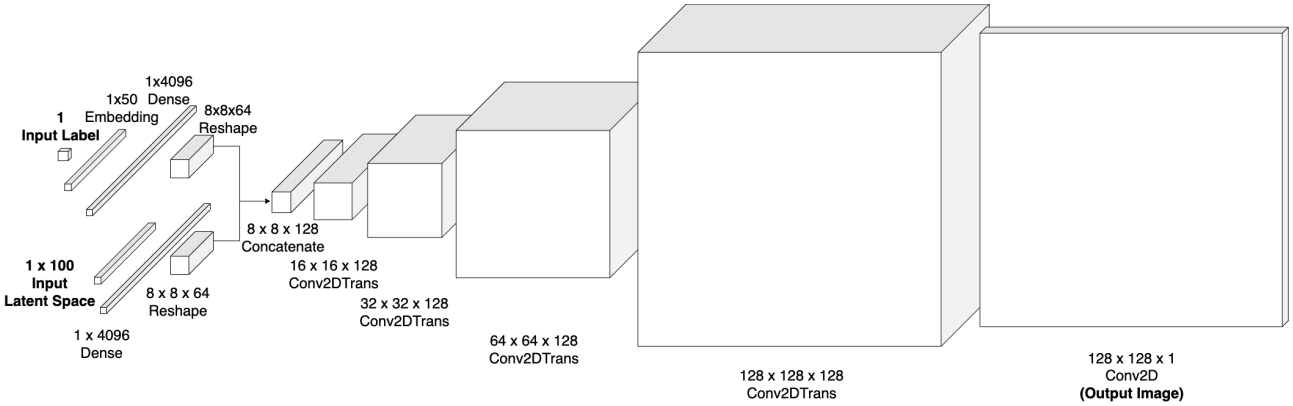

Figure 5. Conditional GAN Generator Architecture

Figure 6. Example output of X-Ray cGAN w/ disease label "Mass"

Epoch	Disc. Accuracy on Fake	Disc. Accuracy on Real	Disc. Loss	Gen. Loss
10	91%	82%	0.346	2.62
20	86%	79%	0.367	1.77
30	92%	81%	0.439	2.51
40	91%	86%	0.447	2.41
50	88%	82%	0.465	2.46

Table 1. The statistics of meme dataset after filtering

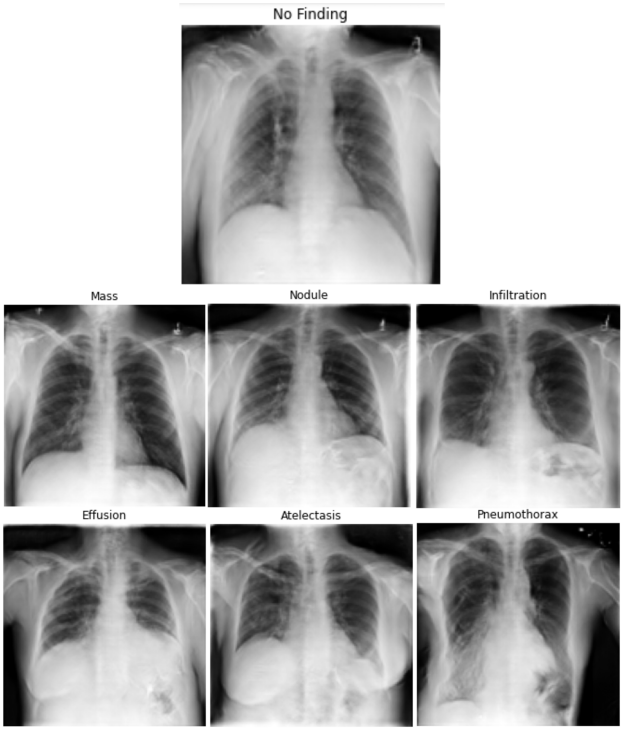

Figure 7. Example outputs of different disease labels generated from our cGAN

5. Results

After training for 50 epochs, we found encouraging visual results (See Figure 6). The example shown in Figure 6 was conditioned on the label "Mass" and produced a pair of lungs with a visible mass on the lower part of the right lung. The cGAN was also able to capture elements from the training data like the collar bones, ribs, arms, and shoulders. However, the discriminator accuracies for fake and real data were 88% and 82% respectively. This can be attributed to the discriminator being trained faster, being a simpler model. This is very common in GAN training and is often remedied by training the discriminator on fewer examples per batch.

Figure 7 shows examples of some generated images of each of the seven different disease classes. We're able to see that our cGAN is able to produce a variety of chest x-rays that indicate we did not overfit.

Table 1 shows the accuracies and losses for the generator and discriminator over the training period. The accuracies generally hovered around the same values for the duration of training, showing that the generator improved as the discriminator improved, but the generator did not improve at a fast enough rate to bring down discriminatory accuracy. Discriminator loss went up as did the generator loss.

6. Evaluation

GANs are notoriously difficult to quantify performance for. Since there is no single loss function to minimize, it is often unclear how to measure performance and evaluate the quality of our models. We largely tested the models through manual inspection of synthetic data. Shown in Figure 8 is a side-by-side comparison of a real image next to a generated one as an example of how we evaluated our generator. We kept in mind that the real images are of 1024x1024 pixel quality, while ours are only 128x128, explaining some of the quality loss.

While this was an easier approach than attempting to quantify the quality of our synthetic data, there are clear drawbacks. Neither of us have medical experience and while we know roughly what an X-ray looks like, there are nuances that we likely missed. This is especially true of the conditional GAN since we have little idea what the different diseases should look like. It was clear that there were differences when we changed the label for an image but we were unsure whether a doctor would agree that the X-ray now exhibits the specified disease.

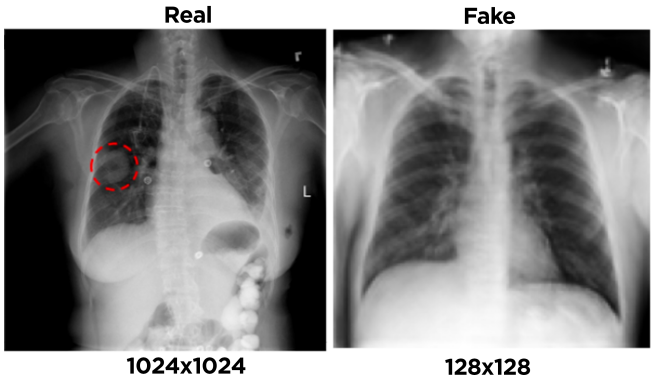

Figure 8. Example of a real chest x-ray from the data set and an example of a fake generated image by our cGAN.

7. Discussion

We were successfully able to artificially produce realistic that compare well upon the visual eye test. The model was capable of producing a variety of different images, all looking in similarity to actual chest x-rays. Computation limitations restricted us to training with only approximately 50,000 images and for only 50 epochs. With a longer available runtime and a more powerful GPU, we could train on more training images and for a longer time, yielding better results. However, the results we did get in 50 epochs were very encouraging for the potential of this model.

A slight problem that was observed was that given the *same* latent space input vector but different labels, the images differ but very slightly in a manner that is difficult to notice. Figure 9 shows the outputs of our generator upon being given the *same* latent input vector but different class labels. The side-by-side comparison helps us visualize the effectiveness of the input label and its embedding. Small differences between the images can be seen, but very subtle.

To validate our model architecture and our method of conditonalizing the GAN, we trained the same model on 50,000 images of the MNIST fashion data set for 50 epochs as well. An example of the MNIST-trained generator's output when given the same latent input vector and different labels is shown in Figure 10. The model trained for the same amount of epochs and on the same amount of data shows visibly completely different results for each class, leading us to believe that our model is not the problem. The lack of variation between generator outputs of our chest x-ray cGAN is likely due to the small variations between classes in the data set. All images in the data set generally look like lungs with all large, prominent features being shared. The difference between inputs to the MNIST fashion data set have much more variation. The structure of a pair of pants differs greatly from the structure of a sandal. Even

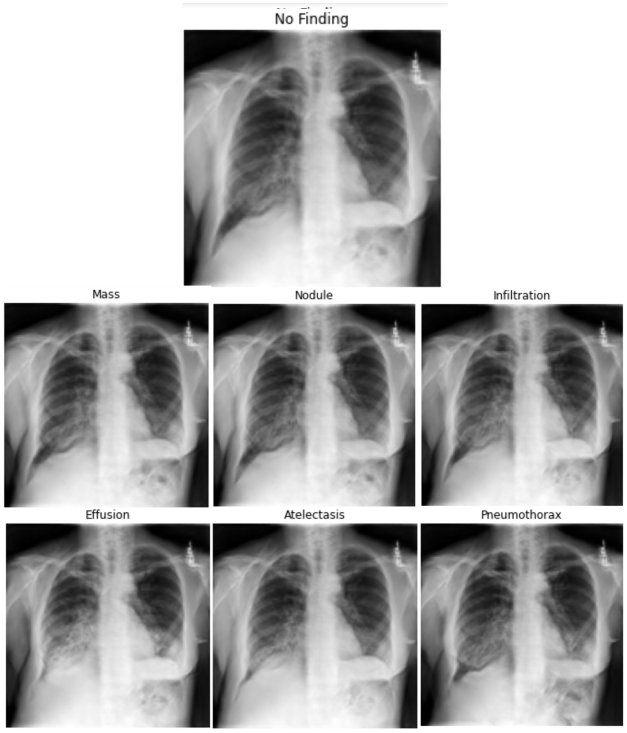

Figure 9. Comparison of the generator output given the same latent input vector but different class labels.

in the fashion MNIST-trained output, we see very similar images for the pullover, coat, and shirt labels. These three have similar structure given their two long sleeves and torso area, differing only by details. We can conclude that the conditional GAN struggles to pick up on the small nuances between classes and is more tuned to deal with large structural differences.

8. Limitations and Future Work

As discussed in data pre-processing, the main limitation of our work was the down scaling that we did of images and limited usage of the data set. Our results are suggestive that a conditional GAN could be trained on a larger size image with high quality output. Using a full size image of 1024x1024 would result in images that are more clinically relevant and likely can be better used for classifying real world data.

Another line of future work is to attempt to train the GAN on a sparse data set. The main application of this work is expanding small data sets. Testing our model on data with only a few hundred images or thousand images will prove that is has applications in a real setting.

Finally, we largely ignored the comorbidities of diseases which could have caused issues with the discriminator. It

Figure 10. Comparison of the MNIST-trained generator output given the same latent input vector but different class labels.

may have been penalized for guessing a sample was from a certain class because we ignored the label in the file. This is a major limitation of our GAN and encoding the labels differently may have helped with this. We could have created an encoding for an image having multiple diseases and train the GAN with that information.

9. Conclusion

In summary, we developed two generative models for X-ray images afflicted with a variety of diseases based on the NIH X-ray data set. The first was a traditional DCGAN while the second was a conditional GAN. Both models were able to produce high quality, realistic looking X-ray images despite limited compute resources and training time. We expect that with more time and financial resources, we would be able to expand this work beyond small images to handle a variety of diseases with high fidelity.

References

Baltruschat, I. M., Nickisch, H., Grass, M., Knopp, T., and Saalbach, A. Comparison of deep learning approaches for multi-label chest x-ray classification. *Scientific Reports*, 9(1), April 2019. doi: 10.1038/s41598-019-42294-8. URL https://doi.org/10.1038/s41598-019-42294-8.

Denton, E. L., Chintala, S., Szlam, A., and Fergus, R. Deep generative image models using a laplacian pyramid of adversarial networks. *CoRR*, abs/1506.05751, 2015. URL http://arxiv.org/abs/1506.05751.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial networks, 2014. URL https://arxiv.org/abs/1406.2661.

Mirza, M. and Osindero, S. Conditional generative adversarial nets. *CoRR*, abs/1411.1784, 2014. URL http://arxiv.org/abs/1411.1784.

Radford, A., Metz, L., and Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. In Bengio, Y. and LeCun, Y. (eds.), *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL http://arxiv.org/abs/1511.06434.

Shorten, C. and Khoshgoftaar, T. M. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1), July 2019. doi: 10.1186/s40537-019-0197-0. URL https://doi.org/10.1186/s40537-019-0197-0.

Sundaram, S. and Hulkund, N. Gan-based data augmentation for chest x-ray classification, 2021. URL https://arxiv.org/abs/2107.02970.