COVID-19-Affected Lung CT Image Generative Network

Jivan Gubbi
jcgubbi@g.ucla.edu

Shikha Mody
smody@g.ucla.edu

Brendon Ng
brendonn8@g.ucla.edu

UCLA Computer Science

Abstract— **The effects of COVID-19 span practically the entire globe. Many studies on its diagnostics have been conducted and many researchers have attempted to understand it. We have developed a Generative Adversarial Network (GAN) that generates images of lungs with COVID-19 to serve as additional data. This GAN utilizes Computed Tomography (CT) images of lungs to train a generator, and optimizes output from 100-dimensional vectors of "noise" to generate images of COVID-19-afflicted lungs. In addition, the generated images can model the progression of COVID-19 with linear interpolation while being compared to actual training images with latent space visualization. Ultimately, the GAN produced relatively accurate images of COVID-19-afflicted lungs, which could then be compared using linear interpolation to model the steps between one generated image and another.**

I. Introduction

COVID-19 is a highly contagious respiratory illness that has caused pervasive damage to communities around the world. While many scientists and researchers are focusing on the spread and detection of this virus, it is also important that we gain a better understanding of the progression of the disease in patients. Using generative adversarial networks (GANs), we can model the progression of the illness in the lungs of patients in the hopes of drawing both short and long term conclusions.

Previous clinical studies have helped build a timeline of recovery in patients without severe respiratory distress. Based on computed tomography (CT) imaging of the lungs, four stages of recovery have been identified based on lung abnormalities found in images (Pan, et al.). The average recovery time is 17 days with the most intense symptoms and lung involvement 10 days after diagnosis.

Because COVID-19 is a rapidly evolving disease, being able to see its future stages would assist medical professionals in resource allocation and course of treatment for critical patients. Using a GAN can create a semi-concrete result for disease progression before the disease actually progresses to that point. This generated image can then be compared to actual CT images. Also, modeling the disease progression can increase the overall knowledge base for COVID-19, which could help in finding a cure/vaccine/treatment for it.

A. Review of State of the Art

Computer vision is a broad field with which its applications can be used in many areas of study and industries. Medical experts have turned to computer vision techniques in the past to help with the detection, diagnosis, and analysis of diseases and conditions through various medical imaging and computer vision techniques. The intersection of the medical field with computer vision has seen many different applications from the use of convolutional neural networks and deep learning to detect skin cancer accurately and cost effectively (Saba, et al.), to the use of Artificial Neural Networks to aid current diagnosis techniques in the prediction of Thromboembolic Strokes (Shanthi, et al.).

As COVID-19 spread quickly throughout the world, researchers flocked to find ways to detect, track, and analyze the effects of coronavirus on the body. Many researchers have rushed to find ways to detect COVID-19 using various types of medical imaging techniques and computer vision algorithms. A group from Tel-Aviv used images from non-contrast thoracic CT scans in conjunction with 2D and 3D deep learning models to track the evolution of the disease over time to distinguish between COVID positive and negative cases, achieving an incredible 0.996 AUC (Area Under the ROC Curve) (Gozes, et al.). Others, such as a group from the Zonguldak Bulent Ecevit University in Turkey, have looked at rapid diagnosis of COVID-19 through X-ray imaging using convolutional neural network models. This group used a pre-trained neural network (ResNet50) and were able to obtain an accuracy of 98% on a data set of 50 COVID patients and 50 control patients (Narin, et al.). While this is very promising research, the limited data set remains an issue. A larger scale study was conducted with over 4300 CT scans to develop a 3D deep learning framework also built upon RestNet50 called COVNet. This study included patients with community acquired pneumonia to ensure that COVNet would be able to distinguish COVID from other diseases that impact the lungs. COVNet is able to identify CT scans with COVID and community acquired pneumonia with a 0.96 AUC for COVID-19 and a 0.95 AUC for community acquired pneumonia (Li, et al.)

1) GANs and Their Applications in Medical Imaging: Prediction of the evolution and progression of COVID-19 in the body has been an application high on the wish lists of many doctors throughout this pandemic. This type of application has been solved with computer vision for diseases in the past by way of GANs. A generative adversarial network (GAN) is a machine learning structure that uses two neural networks in order to generate results that are similar to real-world data. GANs utilize two competing neural networks, one generates new examples that are similar to what we expect by upsampling a one-dimensional vector of random noise, and the other classifies between real samples and fake samples. These two models compete with each other (hence the term ”adversarial”), creating a feedback loop where both the generator and discriminator improve their performance as they train against each other. This results in a generator that can create new data that is eventually indistinguishable from the real data.

GANs are useful for machine learning and computer vision applications to model the evolution of certain features. In medical imaging, GANs have the potential to monitor disease progression. They have already been used to model the progression of Alzheimer’s disease with MRI images (Bowles, et al.). In this study, MRI images with certain structures and features (i.e. locations of the brain, ventricles, hippocampus) were normalized and supplied to a GAN in order to see a possible evolution of the disease for that patient. For a COVID-19 application, CT images with isolated features in the lungs would be the data source, and the disease progression would be modelled in the lungs themselves. This application extends the Alzheimer’s disease study to be applied to a different style of medical imaging, and different features to extract from the dataset. Nonetheless, the concept of applying a GAN for monitoring disease progression is the same.

In this project, we develop a GAN designed to produce CT images of COVID-infected lungs from a random noise vector with a dimensionality of 100. The data our GAN is trained on includes early stage, middle stage, and late stage CT scans of COVID-infected lungs. We then explore the 100-dimension random latent space to find structure and interpolations that can model the progression of the diseased lung over time.

II. Methods

A. Data Processing

We used the COVID-19 CT scan database hosted on Github compiled by UCSD-AI4H. It consists of 349 scans of lungs with COVID-19 from Wuhan, China. This dataset also contains scans of lungs without COVID-19 for baseline diagnostics, but since this model is meant to generate images of lungs with COVID-19, the focus stayed on these scans.[1] The labels for this data set were relatively inconsistent due to inconsistent time records and the rapid nature of COVID-19 dataset availability. Nonetheless, these CT scans provided a foundation for the GAN to learn and generate images of lungs.

Due to the fact that all of these scans were different sizes, preprocessing the data was necessary. After extracting the images, each one was converted to grayscale and resized to be 128x128 pixels. We also doubled the size of the data set by flipping each image laterally. This allowed for a uniform training data set. The data set for the latent space widget consisted of the same COVID-19 lung scans along with the non-COVID 19 scans for more robust exploration.

B. Model Architecture

The GAN is comprised of two competing neural networks. One generates fake images and the other attempts to distinguish between the generated fake images and real images. Specifically, it is a deep convolutional GAN (DCGAN) because of the various convolutions and deconvolutions included in the model. The generator takes in multidimensional random noise as an input. The generator and discriminator have almost symmetric designs. In the generator we use deconvolution to upscale the input to 128x128. We do so by doubling the image size with each layer. The discriminator takes in an image and uses convolution to decrease the image size by half with each layer until we have a single output.

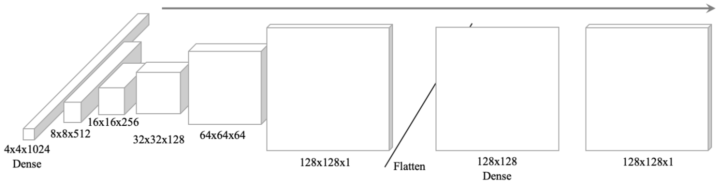

Fig. 1: Generator Architecture

As you can see in Figure 1, we have a slightly different structure with the last three layers of the network than we do in the discriminator. In the generator we added a fully dense layer as the final step in the network to fix issues with image pixelation that arose with upscaling our image. With the fully dense final layer we had less granular images and thus, improved performance.

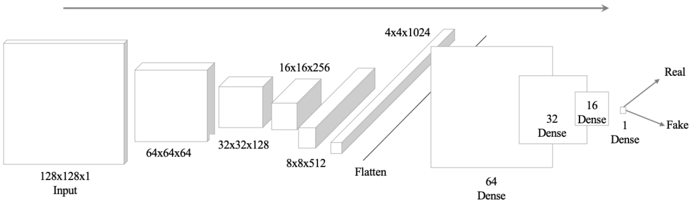

Fig. 2: Discriminator Architecture

C. Model Training

We trained the model for 5000 epochs using mini batch gradient descent with batch size set to 50. We chose this rather than stochastic gradient descent because it allows us to better estimate the true gradient of the testing set during the training. Binary cross entropy was used as the loss function. Loss values were saved at each epoch.

Training the model for 5000 epochs resulted in relatively high quality images that resembled lungs. What we are looking for when training is the optimal point of the min-max equation in Figure 3. This figure shows that GAN training is a zero-sum game. When the discriminator succeeds in classifying a fake image, the generator fails. When the

[1] Non-COVID scans were used to train the latent space widget in order to have an all-encompassing picture of the range of CT scans.

generator fools the discriminator, then it is the discriminator that fails. Because of this, training a GAN requires a balance of generator and discriminator capability. If the discriminator becomes too powerful, then the gradient of the generator will vanish and it will become unable to learn and produce images. Also, because of the adversarial nature of the network, performance often oscillates and fails to converge.

$$\min_{G}\max_{D} V(D,G)$$

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

Fig. 3: Min Max Training

We attempted to train the model in two different ways. We began by training the generator and discriminator the same amount, but as we began testing on larger epochs it became clear that stability of the model was an issue during training. We also noticed that our discriminator seemed to consistently outperform the generator. However, training the generator twice as much as the discriminator per batch results in higher stability and keeps the discriminator loss from vanishing (Bang, Shim). We attempted to train our model in this way as well, training the generator on twice the amount of data as the discriminator per batch. We found little difference in the quality of the output between the two methods and opted for the simpler one-to-one training. We generated a series of images from different noise vectors after each training to evaluate and test the model.

III. Results

A single training of a GAN would let it make a small set of different lungs based on the input that was given. As you can see in Figure 4 and Figure 5, the lungs produced by the GAN are high quality and show similar structure to the original data set. The structure of the lung is clear and you can also see affected areas of the lung, similar to what is seen in pneumonia and other respiratory diseases. We utilized a visualization widget (Duhaime) with an autoencoder structure to smoothly transition between images in the data set. This allowed us to explore the latent space of the images in the data set to compare generated images (COVID and non-COVID for more robust and varied visualizations)[2].

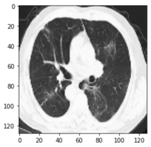

Fig. 4: Real Data

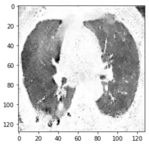

Fig. 5: Generated Data

We found different sizes and different stages of the COVID-19 disease based on noise input with 100 dimensions. A major issue faced was that our model often suffered from mode collapse. A single model would only generate a few distinct lungs despite passing in different noise vectors each time. The main method of evaluation that we used was qualitative comparison between generated images and images part of the data set. In the future, we could incorporate quantitative methods such as average log-likelihood to measure the variation of the generated images (Borji). Nonetheless, qualitative evaluation methods provided a guide for how to adjust our GAN (Borji).

In Figure 6, it is evident that when the discriminator loss is high, the generator loss is low and vice versa. Also, the generator loss is much higher in general than the discriminator loss. It is much more difficult for the discriminator to generate realistic images than it is for the discriminator to figure out which images are fake. The natural difference in difficulty of these two problems leads to much different loss values.

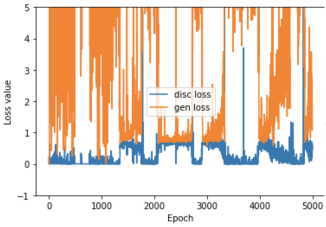

Fig. 6: Loss Over Epochs

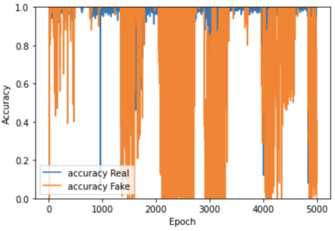

Fig. 7: Discr. Accuracy

In Figure 7, the accuracy of the discriminator is plotted. This shows the difficulty of training a GAN. The GAN goes between cycles of a high performing discriminator that can distinguish the real images from fake images and cycles of not being able to distinguish anything meaningful. When the latter occurs, the model diverges and the generator begins to produce black images. This was a large issue that we found when training the model as it would fall into long periods of only black images being produced lasting hundreds of epochs.

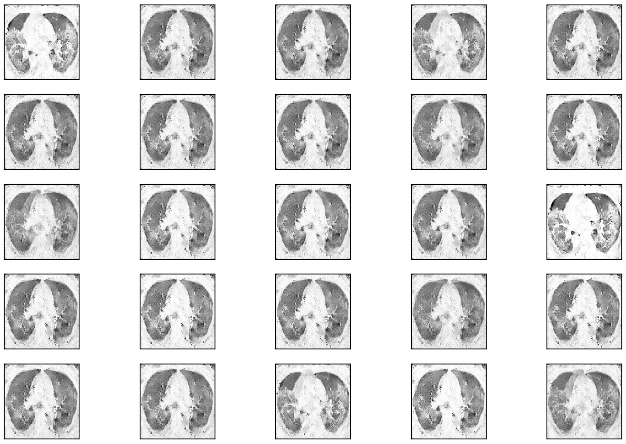

Fig. 8: Various Generated Images

[2] See Appendix for full description

IV. Interpolation

The 100-dimension latent space that follows a multivariate normal distribution is mapped by the generator to create the distribution of sampled images that follow the 16384 (128x128) dimension ”probability distribution” of COVID-infected lungs. As the 100-dimension latent space has structure, the resulting distribution of generated images has structure as well. Interpolating between two noise vectors allows us to see the progression between two images in an incremental manner. Interpolating between a CT scan of an early stage diseased lung to a late state diseased lung will allow us to approximate the progression of the disease over time. We utilized linear interpolation for ease of computation. Based on how many steps to interpolate, different ratios would be used to strategically map each step of the progression. The key components of this modeling procedure were that it used noise vectors to interpolate, and generated images based on those intermediate noise vectors. In essence, all generated images to model the progression of lungs came from our GAN.

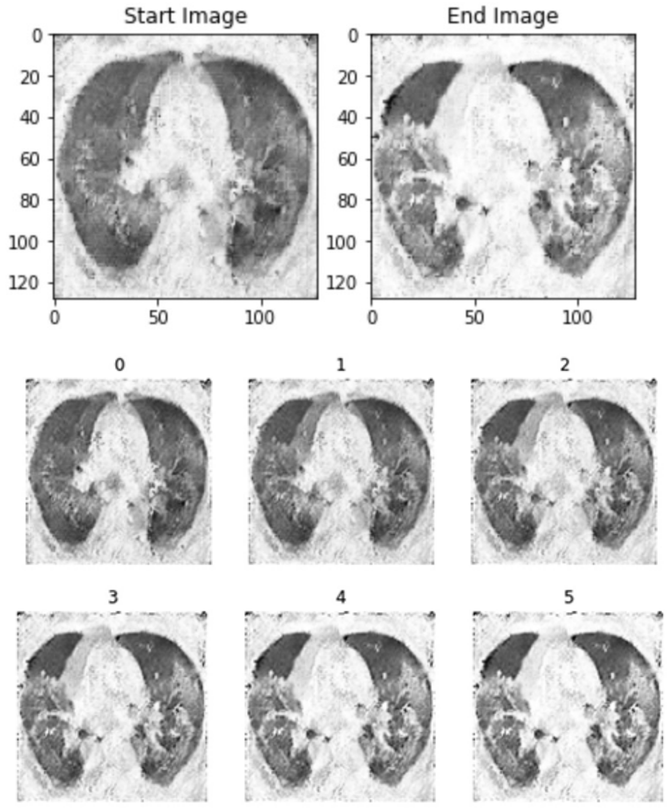

Fig. 9: Interpolation modeling progression from early to late stage infected lungs

V. Discussion

A. Extensions and Future Directions

The ultimate goal for this GAN model is to be used to model the progression of COVID-19 through the lungs. Instead of generating images from 100-dimensional noise, the ultimate goal is to convert from image to image, similar to a Pix2Pix generator (williamFalcon). The main aspects this would change would be the structure from a GAN to a conditional GAN, where we take the labels of the images into account. This way, we can utilize the latent space that the GAN generates to model a linear progression of how COVID-19 progresses inside the lungs, as viewed in CT scans. Potential challenges of this approach include: inconsistent data labelling and restructuring of the current GAN. Data labelling and quantity is a clear limitation of COVID-19 since it is a relatively recent disease and there is not a massive database of imaging data. Nonetheless, by training with many epochs, we can overcome the problem of quantity. Inconsistent labelling is still a substantial issue.

Another area of future study would be to mitigate the effects of mode collapse on our GAN. Using a Wasserstein loss function has proven effective for mitigating mode collapse (Arjovsky et al.). In addition to this, a VEEGAN has been proposed as an alternative to the traditional GAN. VEEGANs use an additional neural network to map from data to noise vectors to reduce the chance of mode collapse (Srivastava et. al). Being able to produce a wide variety of outputs would be favorable, since the true distribution of lung CT scans is quite varied based on individual lung structure. We attempted using a WGAN on this data but had limited success. Implementing a VEEGAN or further training a WGAN could improve the variety of the resulting lung data and reduce mode collapse and make our model more generalizable. This would also result in a more meaningful latent space than the case of repeated generated photos with many different random noise vectors.

Other general improvements that could be made in the future include training for more epochs, expanding the data set, and potentially adjusting the structure of the GAN itself. Training for more epochs would, in theory, improve the quality of generated images. However, when we tried to train for more epochs, we found that the GAN experienced either divergence or mode collapse. There is a delicate balance between training of the GAN and the results that it produces. Expanding the data set could widen the scope of generated images, and also made the network more robust. When initially researching this topic, we came across examples of GANs with the MNIST and facade database, image data sets with tens of thousands of images. In contrast, medical images with COVID-19 were relatively restrictive; the data set we used only had 349 CT scans. As a result, the quality of output images was lacking. By augmenting the data set with left-to-right flipped images as part of the data set, we were able to overcome the challenge of a less dense data set. Nonetheless, with machine learning in general, more data usually means better results. Lastly, changing the structure of the GAN itself could output more accurate images. As discussed previously, WGAN or a model like VEEGAN could improve the performance of the current GAN. This would require more extensive reworking and training, and with more time and computing resources, this could be an avenue for greater exploration.

B. *Conclusions*

The conclusions that we drew mainly revolved around the training process of our GAN. The result of our work was a GAN that could produce realistic looking lungs that are similar to those of patients with COVID-19. However, our model is limited to producing a small subset of lungs. We found that as we increased the number of epochs that we trained for, we had clearer images but the training remained unstable. Most times it did not converge or we saw vanishing gradients with the generator. Future experimentation into using different GAN structures and loss functions should be done to determine a more reliable training method. Nonetheless, our results are significant because of their qualitative similarity to actual lung scans with COVID-19. This can potentially lead to a greater amount of data in this realm which can assist in other efforts to diagnose and analyze the progression of COVID-19. In addition, our process of interpolation is similar in function to a conditional GAN; our process could set the stage for a potential conditional COVID-19 GAN and other advancements to arise.

Exploration into the latent space and interpolation allows us to draw conclusions about degree progression and severity of the illness. Future study into this will allow better interpolation between images, leading to more realistic progression tracking, and usable synthetic data to expand the amount of available data.

REFERENCES

[1] Bowles, Christopher, et al. “SPIE Medical Imaging.” SPIE Library, 2018, www.spiedigitallibrary.org/conference-proceedings-of-spie/10574/0000/Modelling-the-progression-of-Alzheimers-disease-in-MRI-using-generative/10.1117/12.2293256.short?SSO=1.

[2] Gozes, Ophir, et al. “Rapid AI Development Cycle for the Coronavirus (COVID-19) Pandemic: Initial Results for Automated Detection & Patient Monitoring Using Deep Learning CT Image Analysis.” ArXiv, 10 Mar. 2020, arxiv.org/pdf/2003.05037.pdf.

[3] Li, Lin, et al. “Artificial Intelligence Distinguishes COVID-19 from Community Acquired Pneumonia on Chest CT.” Radiology, 2020, p. 200905., doi:10.1148/radiol.2020200905.

[4] Narin, Ali, et al. ”Automatic Detection of Coronavirus Disease (COVID-19) Using X-ray Images and Deep Convolutional Neural Networks.” ArXiv, 24 Mar. 2020, https://arxiv.org/abs/2003.10849

[5] Pan, Feng, et al. “Time Course of Lung Changes On Chest CT During Recovery From 2019 Novel Coronavirus (COVID-19) Pneumonia.” Radiology, 2020, p. 200370., doi:10.1148/radiol.2020200370.

[6] Saba, Tanzila, et al. “Region Extraction and Classification of Skin Cancer: A Heterogeneous Framework of Deep CNN Features Fusion and Reduction.” Journal of Medical Systems, vol. 43, no. 9, 2019, doi:10.1007/s10916-019-1413-3.

[7] Shanthi, D, et al. “Designing an Artificial Neural Network Model for the Prediction of Thrombo-Embolic Stroke .” International Journals of Biometric and Bioinformatics, vol. 3, no. 1, Mar. 2009.

[8] Bang, Duhyeon. Shim, Hyunjung “Improved Training of Generative Adversarial Networks Using Representative Features.” Jan, 2018.

[9] Tiu, Ekin. Understanding Latent Space in Machine Learning. 4 Feb. 2020, towardsdatascience.com/understanding-latent-space-in-machine-learning-de5a7c687d8d.

[10] williamFalcon. “WilliamFalcon/pix2pix-Keras.” GitHub, 27 Mar. 2017, github.com/williamFalcon/pix2pix-keras.

[11] Srivastava, Akash et al. ”VEEGAN: Reducing Mode Collapse in GANs using Implicit Variational Learning” Advances in Neural Information Processing Systems 30, 3308-3318, 2017.

[12] Arjovsky et al. ”Wasserstein GAN” eprint 1701.07875, 2017.

[13] Borji, Ali. “Pros and Cons of GAN Evaluation Measures.” Computer Vision and Image Understanding, vol. 179, 2019, pp. 41–65., doi:10.1016/j.cviu.2018.10.009.

[14] Duhaime, Douglas. “Visualizing Autoencoders with Tensorflow.js.” Douglas Duhaime, 26 May 2019, douglasduhaime.com/posts/visualizing-latent-spaces.html.

APPENDIX

A. *Latent Space Visualization Widget*

The latent space visualization widget serves as a benchmark to compare generated images with a data set of actual CT images. It consists of an autoencoder structure, where an encoder takes the original image and maps it to a lower dimensionality, and a decoder maps it to an increased dimensionality once again (Tiu). This allows us to see the connection between data points in a smoother manner. Latent space is essentially a compressed representation of the data at the lower dimensionality (Tiu). Similar images will be closer together in the latent space since they would have similar features (Tiu). This is useful for our GAN because instead of isolating solely discrete images to compare the generated image to, the latent space representation creates a continuum of images based on the data set itself; it shows how each image is connected to each other, and hence, we can see how our generated images can fit into these images. The goal of the GAN is to generate realistic images, so using the latent space can assist in determining where exactly they fit in.

The encoder consists of a series of five densely-connected layers with the final layer at the size with lower dimensionality. The decoder takes this as input and uses densely connected layers to map it back to the input image size (128x128); in essence, the structure of the encoder and decoder are mirror images of each other. Combining these components in a single model (with optimization and loss functions for greater efficiency and measurement) creates an autoencoder.

This autoencoder was trained on the same dataset as the GAN, with added non-COVID-19 lung scans in order to visualize the entire spectrum of lung CT scans (healthy included). This allowed the latent space to be more dense than if only a subset of data were used. After 100 epochs, the loss (Mean-Squared-Error) stabilized, and thus the latent space was generated. The loss stabilized at approximately 240, which is relatively high. Optimizing this visualizer for decreased loss could be possible with a more dense data set; another possible strategy (which we employed) was to increase the number of layers in the model itself. However, the loss value stabilized in the same region as with less layers.

The widget itself utilizes an exported version of this autoencoder model to dynamically display the latent space map with trackball navigation. It smoothly transitions between features and images so that the user can see how images are related to each other. This results in an interactive benchmark to compare the generated images.

B. *Demo and Model*

Link to the code and demos: Github link
Link to the model and data: Model link