CSIC 5011 / MATH 5473 Project 2
Imaging's Potential to Assist in COVID-19 Crisis

Yingshu CHEN
Department of
Computer Science and Engineering
ychengw@connect.ust.hk

Wing Hei SUM
Department of
Industrial Engineering and Decision Analytics
whsumaa@connect.ust.hk

Ho Pan IP
Department of Mathematics
hpipab@connect.ust.hk

Zipeng WU
Department of Physics
zwubp@connect.ust.hk

Abstract

Coronavirus disease 2019 (COVID-19) is a contagious disease caused by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2). An efficient and accurate abnormalities detection in chest X-rays (CXR) of infected patients can assist health care staff in the battle against COVID-19. Recently, researchers collected some CXR images for detection via intelligent computer visual techniques, e.g., dimensionality reduction for clustering, deep-learning based classification. However, limited to total number of data, the visual classification has relatively low accuracy and may lead to mis-detection. The state-of-the-art generative model (StyleGAN2) for limited data is now capable of photorealistic image generation. We investigate the potential of synthetic images additional to existing dataset to improve COVID-19 CXR classification accuracy. Qualitative and quantitative evaluations validate the potential of imaging assistance in COVID-19 crisis. See detailed implementation in `https://github.com/chenyingshu/csic5011_project2`.

1 Introduction

COVID-19 pandemic crisis has been spreading throughout the world over a year and the front-line and support staff at hospitals and health systems have been keeping tirelessly battling the virus. In some countries and areas lacking detection kits, it is a challenge for health care professionals to differentiate normal people, COVID-19 infectious patients, and other patients sharing similar symptoms (e.g. other viral pneumonia) from symptoms and chest X-ray (CXR) images, etc. We explore the possibility of CXR images classification via some data analysis techniques including data dimensionality reduction such as PCA, machine learning techniques for classification, and generative model aiding limited visual data.

Recently, a limited set of COVID-19 CXR dataset (Rahman et al. (2021)) is widely probed in computer vision community. In this work, we investigate the dimensionality reduction approaches for data distribution visualization and evaluation, and propose a data augmentation pipeline to assist medical CXR image classification with limited data.

Our main discoveries and contributions involve:

- Exploration of dimensionality reduction methods as a visualization and evaluation means to visualize data distribution and get data distribution observation.

- In low dimensional space, it is revealed that non-COVID lung opacity infection and normal set of CXR images has similar or overlapped data distribution with COVID-19 one, which complicates COVID-19 detection from chest X-rays.
- High-quality chest X-ray images synthesis with limited data via a conditional GAN.
- We transfer the COVID-19 CXR detection problem to a 4-class classification problem considering other non-COVID lung infectious CXRs as well.
- We validate the effectiveness of synthetic images for medical data augmentation and visual classification enhancement. Qualitatively and quantitatively verify the potential of image generative model to assist in new respiratory infectious disease detection.

2 Related Works

Dimensionality Reduction for Data Distribution Analysis. Dimensionality reduction, or dimension reduction, is transforming data from a high-dimensional space into a low-dimensional space so that the low-dimensional representation retains some meaningful properties of the original data, ideally close to its intrinsic dimension. Working in high-dimensional spaces has many drawbacks: raw data are often sparse as a consequence of the curse of dimensionality, and analyzing the data is usually computationally intractable. Dimensionality reduction is common in fields that deal with large numbers of observations and/or large numbers of variables, such as signal processing, speech recognition, neuroinformatics, and bioinformatics (Boehmke and Greenwell (2019)).

Deep Learning based Classification on Medical Images. Convolutional neural network (CNN) is a class of neural networks in deep learning, which is commonly used in computer vision. It is constructed with architectures of convolution filters and kernels that slide through input features with several hidden layers, thus producing translation responses known as feature maps. Recently, in Kaggle community (Rahman et al. (2021)) a lot of programmers challenge the image classification on COVID-19 chest X-ray database and CNN is one of most commonly used techniques (e.g., Waheed et al. (2020)). Since it has been observed that most existing related works only consider two classes, namely COVID-19 infection and normal lung images, a 4-class CNN classification model that takes into account non-COVID-19 diseases including lung opacity and viral pneumonia would thus be built in this project.

Synthetic Data Augmentation in Deep Learning. Deep learning requires a large amount of training data to ensure the accuracy. But in some scenario such as medical imaging, there is a lack of sufficient images for research use. With the advent of generative adversarial networks (GANs) (Goodfellow et al. (2014)), image generation without explicitly probability density function modeling became possible. In recent years, state-of-the-art generative models (e.g., PG-GAN, Big-GAN, StyleGAN and StyleGAN2) are able to generate photorealisitc images that cannot be differentiated form real images and generative models thrive in the visual community and also in medical imaging field (Yi et al. (2019)). It is also possible to augment limited medical data with powerful generative models . For example, Waheed et al. (2020); Wu et al. (2018); Frid-Adar et al. (2018); Salehinejad et al. (2018) use GAN to enhance images for classification, Perez et al. (2018) augment data for skin lesion analysis, etc. In this work, we generate synthetic chest X-ray images to improve COVID-19 classification.

3 Methods

We propose a data augmentation pipeline for classification. From visualization via dimension reduction approaches, we obtain initial data distribution observation and relationship discovery of the COVID-19, lung opacity, viral pneumonia and normal sets of chest X-ray images. With preliminary investigation, we further inspect deep learning based classification and synthetic data augmentation for COVID-19 detection improvement. Please checkout our Github repository[1] for details.

[1]Source code in `https://github.com/chenyingshu/csic5011_project2`

Table 1: Data Statistics

Class	Normal	COVID-19	Lung Opacity[a]	Viral Pneumonia[b]
Train	9,942	3,366	5,762	1,095
Test	250	250	250	250
Total	10,192	3,616	6,012	1,345
Blended Train	9,942	7,000	5,762	7,000

[a,b] Non-COVID lung infection.

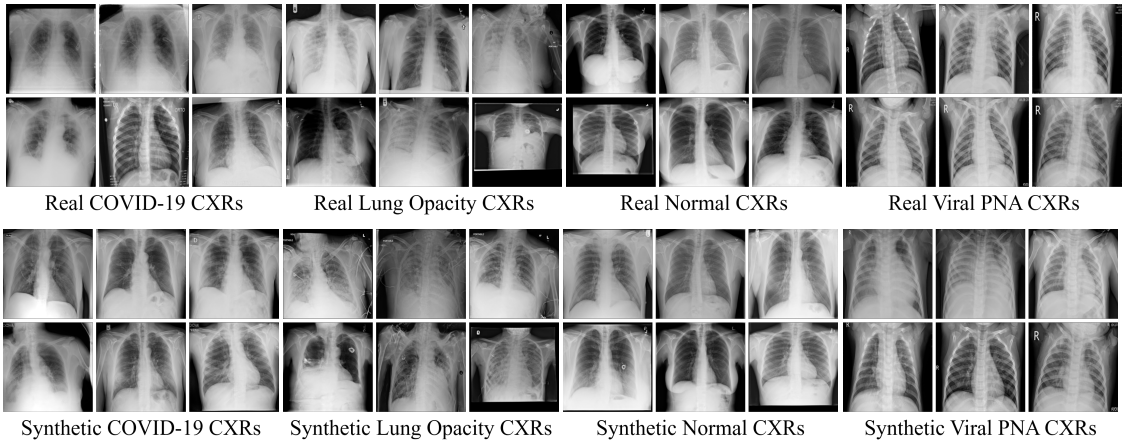


Figure 1: Data samples

3.1 Dataset and Preprocessing

We collected *Normal*, *COVID-19*, *Lung Opacity* and *Viral Pneumonia (PNA)* chest X-ray images in COVID-19 Radiography public Database (Rahman et al. (2021)), which contains 3,616 COVID-19 and 10,192 Normal and some other related CXR images. It is a small dataset with only around 20k images in total, and particularly Viral PNA only contains 1k images and COVID-19 over 3k images. Due to some primal images of 3 channels, all images of 4 classes (labeled as "Normal", "COVID", "Lung", "Viral") are preprocessed in the resolution of 256×256 in 1 channel (grayscale).

For classification we randomly take 250 images per class for testing and we generated 3,634 COVID-19 and 5,905 Viral Pneumonia synthetic images to augment training data for classification[2]. Please refer to Appendix Sec.A.2 for data combination experimentation details.

See more data statistics in Table 1 and image samples in Figure 1.

3.2 Dimensionality Reduction

In this work, we perform principle component analysis (PCA) and manifold learning methods to analyze the original data. We will briefly introduce the underlying idea of these methods.

Linear methods:

- **PCA**: PCA is a linear dimensionality reduction method using singular value decomposition of the data. It linearly transform data while keeping the variance as much as possibleWold et al. (1987).

Manifold learning Methods: Linear methods can be powerful, but often miss important non-linear structure in the data. Manifold Learning can be thought of as an attempt to generalize linear frameworks like PCA to be sensitive to non-linear structure in data.

[2]We trained classifiers with different combinations of real and synthetic data, and use this final version for best and effective performance.

- **Isometric Mapping (Isomap)**: Isomap is one of the earliest approaches to manifold learning. It seeks a lower-dimensional embedding which maintains geodesic distances between all points (Tenenbaum et al. (2000)).
- **Locally Linear Embedding (LLE)**: LLE seeks a lower-dimensional projection of the data which preserves distances within local neighborhoods (Roweis and Saul (2000)).
- **Spectral Embedding (SE)**: SE finds a low dimensional representation of the data using a spectral decomposition of the graph Laplacian. The graph generated can be considered as a discrete approximation of the low dimensional manifold in the high dimensional space (Belkin and Niyogi (2003)).
- **Multi-dimensional Scaling (MDS)**: MDS seeks a low-dimensional representation of the data while keeping distances close to the original high-dimensional space. In general, MDS is a technique used for analyzing similarity or dissimilarity data (Kruskal (1964)).
- **t-distributed Stochastic Neighbor Embedding (t-SNE)**: t-SNE is a stochastic way to model the high dimensional data. It would put 'similar' data points into nearby points and 'dissimilar' data points into distant pointsVan der Maaten and Hinton (2008).

We utilize scikit-learn (Pedregosa et al. (2011)) to implement the above mentioned techniques in both original and synthetic data.

3.3 Implementation of Classification Model

Regarding the implementation details of the convolutional neural network for classification, image data were randomly sampled as training, validation and testing sets for each class respectively. The size of testing data is 250 images per class, considering the class with the fewest images (1,345), which is roughly a 1-5 split. Model training utilized both training and validation images, of which 20% were assigned to the validation set.

In terms of the architecture of CNN, it contains 13 layers consisting of 3 layers of Conv2D and 2 layers of Dense equipped with ReLU and Softmax activation functions, along with MaxPooling2D, Dropout and Flatten hidden layers.

Table 2: Hyper-parameter settings

Optimizer	Loss	Metrics	Batch Size	Epochs
Adam	sparse_categorical_crossentropy	Accuracy	32	20

During the process of training, early stopping is adopted with the monitor being validation loss and patience equal to 5 in order to prevent the model from being over-fitting, thereby optimizing the performance over testing data.

3.4 Selection of GAN Model

To produce synthetic images, we apply conditional styleGAN2 architecture (Mirza and Osindero (2014); Viazovetskyi et al. (2020); Karras et al. (2020)) with adaptive discriminator augmentation mechanism (Karras et al. (2020)) for better performance.

GAN with limited data. The on-the-shelf GAN models such as Big-GAN, StyleGAN that support photorealistic image synthesis in high quality usually need enormous training data which is not suitable for our limited data. Fortunately, the up-to-date style-based generator with adaptive discriminator augmentation (ada) mechanism designed specifically for limited data, i.e., *StyleGAN2-ada*, in the work of Karras et al. (2020) can synthesize realistic images with only thousands of training images.

Conditional GAN. In our work, we trained a conditional generative model with 4 labels instead of 4 separate unconditional models. Karras et al. (2020) show that conditional model has better performance than unconditional one with experiments on CIFAR-10 dataset with 10 classes. In this report, we show a similar comparison experimental result to compare the performance of both types of models. Table 3 illustrates synthesis evaluation metrics comparison between generation from

separately trained unconditional models and the conditional model. Qualitative comparison can be check in Appendix Sec.A.1.

We conjecture it is because conditional GAN train with more images ($\sim$20k in total) while sole GANs were trained with only 1k to 10k images respectively. In addition, to some extent, using a common generator to produce CXR with different types assists better and quickly to converge the model since all CXR images share similar basic chest structure.

4 Experiments and Results

In this section, we conducted experiments on different data combination (e.g., real data only, real and synthetic blended data) to verify the effect of synthetic data augmentation for COVID-19 detection improvement. Please checkout our repository[3] for implementation details.

4.1 Evaluation Metrics

To evaluate the performance of COVID-detection, we apply dimensionality reduction approaches to visualize data distribution and different metrics to quantitatively assess image distribution similarity and classification accuracy for each class.

Low-dimension Visualization. we use dimensionality reduction methods (see Section 3.2) to visualize both synthetic and real images in low-dimension space to intuitively observe the closeness of image distribution.

Distribution Matching. Fréchet Inception Distance (FID) proposed by Heusel et al. (2017) evaluates the similarity of generated and real images, which correlates well with the human perceptual judgment of image quality. The Fréchet distance between two multivariate Gaussians $X_1 \sim N(mu_1, C_1)$ and $X_2 \sim N(mu_2, C_2)$ is

$$d^2 = ||mu_1 - mu_2||^2 + Tr(C_1 + C_2 - 2 \times \sqrt{C_1 \times C_2}). \tag{1}$$

where mu_n is the mean and C_n the covariance of distribution of samples.

We compute FID[4] between training data and 1k synthetic or 1k real images for each target CXR images. The lower FID is, the higher similarity of data distribution.

Classification Accuracy. We use precision, recall, F1-score, and accuracy to evaluate the outcome of our classifier. Precision shows the talent that our classifier could correctly find out the positive observations among all observations it identified as positive. Recall shows the capability that our classifier could detect the positive observations among all observations it correctly predicted. F1-Score gives a balanced result between precision and recall. Accuracy indicates the correct predictions among all the predictions. We regard True Positive (TP) as the classifier correctly identified the positive sample; False Positive (FP) as the classifier misidentified the negative sample as positive; True Negative (TN) as the classifier correctly identified the negative sample; False Negative (FN) as the classifier misidentified the positive sample as negative.

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1\text{-}Score = 2 \times \frac{Recall \times Precision}{Recall + Precision} \tag{4}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

4.2 Dimensionality Reduction Visualization

We first perform dimension reduction techniques to the original data. As shown in Fig.2, the distribution of four class data are not well separated. The 'Normal' data are closer to the 'Viral' data,

[3]Source code in `https://github.com/chenyingshu/csic5011_project2`

[4]Implementation code from `https://github.com/mseitzer/pytorch-fid`

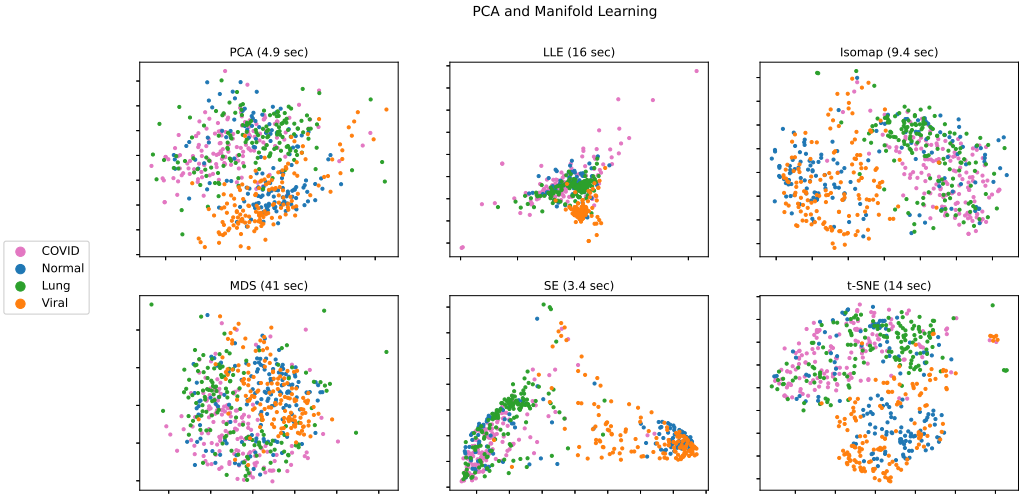


Figure 2: Dimension Reduction for original data. We implemented six different methods including PCA, LLE, Lsomap, MDS, SE and t-SNE. The data contains 4000 pictures from four classes: Normal, COVID, Lung Opacity and Viral Pneumonia.

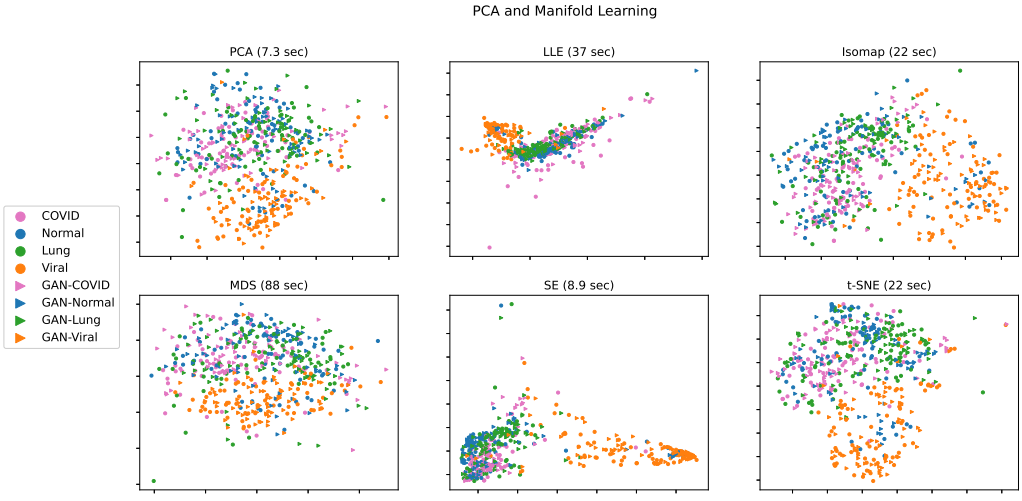


Figure 3: Dimension Reduction for original and synthetic data. The data contains 6000 pictures and the portion of number between original and synthetic pictures is 2:1. We label the original data with circle and the corresponding synthetic data with triangle.

while the 'COVID' data are closer to the 'Lung' data. In general, the 'Normal' data are scattered in all other different data areas.

Then we combine GAN-generated synthetic data with the original real data to perform a similar dimension reduction analysis. The Fig.3 shows the result of different methods. It implies the GAN-generated data distribute relatively close to the corresponding original data.

In summary, the dimension reduction result indicates : 1. Correctly classifying four class data could be difficult. 2. The generated data probably can improve the classification model since it enlarges the dataset with the corresponding distribution.

4.3 Results on Different Data Combination

Data distribution similarity. To validate conditional GAN works better than unconditaion GAN with the COVID-19 dataset, we also trained 4 unconditional GAN with each class of data respectively for comparison. Table 3 shows the FID between real training set and 1k synthetic images from different GAN models. We train a conditional GAN with 4 labels and 4 unconditional GAN using the same generator architector. Our conditional GAN outperforms each unconditional GAN in terms of each class.

Table 4 illustrates the distribution distance between real training set and 1k different combined images with FID metric. All classes of synthetic images (generated from our CGAN) have relatively low FID ($<$30), and the lower FID the more training data of that class. Particularly, our generative model learned best about 'Normal' CXR image distribution with even lower FID than real ones. In other words, the original real normal images are sparse with high FID, and 10k training data can be expected as sufficient scale to train a generator.

Table 3: Unconditional vs. Conditional Generative Model

Model*	Unconditional				Conditional			
	Normal	COVID-19	Lung Opacity	Viral PNA	Normal	COVID-19	Lung Opacity	Viral PNA
FID↓	18.28	33.48	28.74	28.25	**16.39**	**25.24**	**24.22**	**26.85**

*Four unconditional models and one conditional model with 4 labels

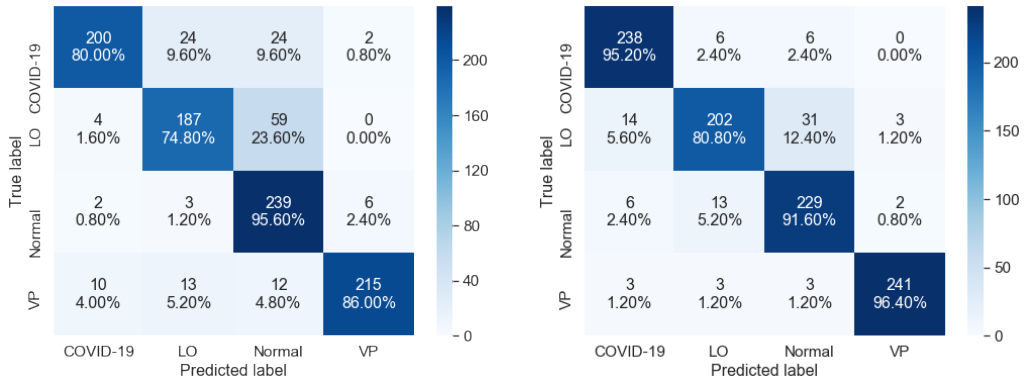
Table 4: FID with Different Data

Data	FID↓	Dataset	FID↓
Real Normal	90.36	Real Lung Opacity	10.65
Real + Synthetic Normal	32.33	Real + Synthetic Lung Opacity	15.16
Synthetic Normal	16.39	Synthetic Lung Opacity	24.22
Real COVID-19	14.46	Real Viral Pneumonia	2.91
Real + Synthetic COVID-19	16.68	Real + Synthetic Viral Pneumonia	12.28
Synthetic COVID-19	25.24	Synthetic Viral Pneumonia	26.85

Classification Accuracy. Figure 4(a) shows the performance of the deep learning classification with the actual data set only. It indicates the classification on COVID-19 and Lung Opacity give weaker performance than the Normal and Viral Pneumonia classes. In the first row of the matrix, only 80% of the prediction correctly classify the CXRs as the COVID-19 cases. The CXRs that are misclassified as Lung Opacity and Noraml cases each occupies 10%. From the second row, less than 75% of the prediction can classify the CRXs as Lung Opacity cases. Normal cases accounts for over 20% of false prediction.

Figure 4(b) analyzes the classification result of the combination of GAN-generated COVID-19 and viral Pneumonia data with the original data. The correct classification in the COVID-19 and Lung Opacity increases from 80% to 95.2% and from 74.8% to 80.8% respectively. The number of misclassifications in the column of Normal reduce a lot. It means the probability that we misplace the patients with COVID-19, Lung Opacity, or Viral Pneumonia to the Normal group will be lower. With the assist of synthetic data, our predictions of the CXRs classes are strengthened. The overall accuracy improves from 84.1% to 91% after the addition of synthetic data.

We further investigate on the classification between COVID-19 and non COVID-19 classes. We can see that the precision rises from 93.6% to 98.4% in FSigure 5(a) and 5(b) respectively, which

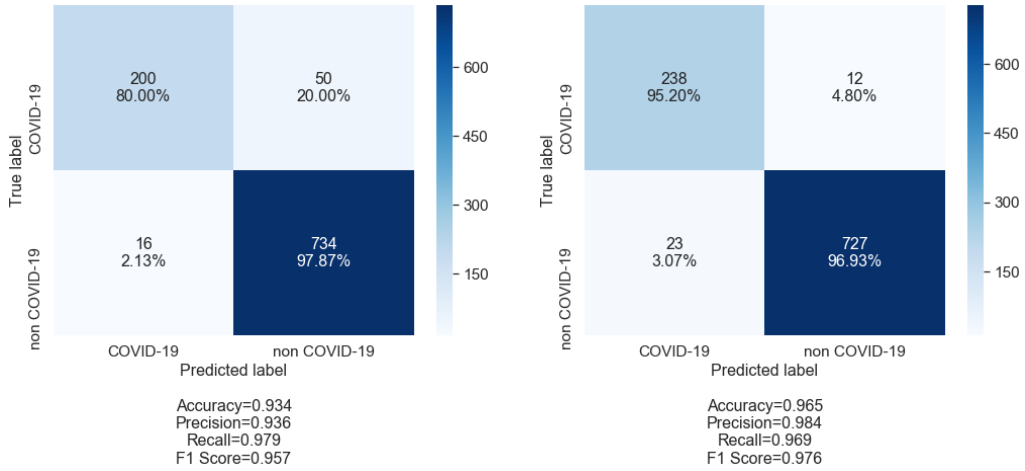


(a) Actual data only

(b) GAN-generated data (COVID-19 and Viral) with actual data

Figure 4: Confusion matrix for all 4 classes

means the ability that our classifier places the true COVID-19 CRXs into the correct class out of the samples predicted as COVID-19 increases. We also yield a higher accuracy (96.5%) and F1-score (97.6%) with the GAN-generated COVID-19 and Viral Pneumonia data with actual data. This brings out that the synthetic data augmentation in deep learning creates meaningful characteristics which boost the classification performance and accuracy.



(a) Actual data only

(b) GAN-generated data (COVID-19 and Viral) with actual data

Figure 5: Confusion matrix for COVID-19 and non COVID-19

5 Conclusion

In this work, we explore the potential of assistance from visual data in COVID-19 crisis. We investigate the possibility of synthetic data augmentation for classification. With the utilization of evaluation metrics (i.e., accuracy and FID) and dimenstionality reduction visualization (e.g., PCA, LSOMAP, etc.), we verify that photorealisitc synthetic data can learn real data distribution and effectively augment exiting actual data for better image classification. With our data augmentation flow, the pretrained classifier can reasonably detect COVID-19 in CXR images and accelerate the detection with the lack of detection kits in some severe pandemic areas. Despite that it does not perform perfectly from medical perspective, but because of its high speed of detection using the pretrained probabilistic classification model, we can use it as speedy initial detection and categorize potential patients in different infectious potential levels for further medical detection.

References

Rahman, T.; Chowdhury, M.; Khandakar, A. COVID-19 Radiography Database. 2021; `https://www.kaggle.com/tawsifurrahman/covid19-radiography-database`, (Accessed on 10 May 2021).

Boehmke, B.; Greenwell, B. M. *Hands-on machine learning with R*; CRC Press, 2019.

Waheed, A.; Goyal, M.; Gupta, D.; Khanna, A.; Al-Turjman, F.; Pinheiro, P. R. Covidgan: data augmentation using auxiliary classifier gan for improved covid-19 detection. *Ieee Access* **2020**, *8*, 91916–91923.

Goodfellow, I. J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Networks. 2014.

Yi, X.; Walia, E.; Babyn, P. Generative adversarial network in medical imaging: A review. *Medical image analysis* **2019**, *58*, 101552.

Wu, E.; Wu, K.; Cox, D.; Lotter, W. *Image analysis for moving organ, breast, and thoracic images*; Springer, 2018; pp 98–106.

Frid-Adar, M.; Klang, E.; Amitai, M.; Goldberger, J.; Greenspan, H. Synthetic data augmentation using GAN for improved liver lesion classification. 2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018). 2018; pp 289–293.

Salehinejad, H.; Valaee, S.; Dowdell, T.; Colak, E.; Barfett, J. Generalization of deep neural networks for chest pathology classification in x-rays using generative adversarial networks. 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2018; pp 990–994.

Perez, F.; Vasconcelos, C.; Avila, S.; Valle, E. *OR 2.0 Context-Aware Operating Theaters, Computer Assisted Robotic Endoscopy, Clinical Image-Based Procedures, and Skin Image Analysis*; Springer, 2018; pp 303–311.

Wold, S.; Esbensen, K.; Geladi, P. Principal component analysis. *Chemometrics and intelligent laboratory systems* **1987**, *2*, 37–52.

Tenenbaum, J. B.; De Silva, V.; Langford, J. C. A global geometric framework for nonlinear dimensionality reduction. *science* **2000**, *290*, 2319–2323.

Roweis, S. T.; Saul, L. K. Nonlinear dimensionality reduction by locally linear embedding. *science* **2000**, *290*, 2323–2326.

Belkin, M.; Niyogi, P. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation* **2003**, *15*, 1373–1396.

Kruskal, J. B. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika* **1964**, *29*, 1–27.

Van der Maaten, L.; Hinton, G. Visualizing data using t-SNE. *Journal of machine learning research* **2008**, *9*.

Pedregosa, F. et al. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* **2011**, *12*, 2825–2830.

Mirza, M.; Osindero, S. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784* **2014**,

Viazovetskyi, Y.; Ivashkin, V.; Kashin, E. Stylegan2 distillation for feed-forward image manipulation. European Conference on Computer Vision. 2020; pp 170–186.

Karras, T.; Aittala, M.; Hellsten, J.; Laine, S.; Lehtinen, J.; Aila, T. Training Generative Adversarial Networks with Limited Data. Proc. NeurIPS. 2020.

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. *Advances in neural information processing systems* **2017**, *30*, 6626–6637.

Appendix

A Additional Results

A.1 Unconditional and Conditional Generation Comparison

In this section, we show visual results to compare the performance of unconditional and conditional models. Fig.6 displays supplementary synthetic CXR images of all 4 classes generated by 4 separately trained models and a 4-class conditioned generative model. From the generation samples, we can see synthetic images produced by unconditional GAN are more likely to generate artifacts, distortion, etc., especially 'COVID-19' and 'Viral PNA' CXRs have most artifacts. It is believed that it is due to the insufficient training data (3k of COVID-19, and 1k of Viral PNA).

A.2 Classification Performance on Different Data Combination

As 'Normal' data are sufficient, out of training efficiency we only consider adding 'COVID', 'Lung' and 'Viral' synthetic data and try to make all classes in equivalent scale. Table 5 displays training performance on different real and synthetic data combination. It shows that with only real data, the validation accuracy is 86.41%. After adding different combinations of synthetic images generated from GANs to our classification training data, the overall performance generally improved. For example, if combining 'COVID' synthetic images with real images, the prediction accuracy of 'Lung' increased significantly from 75.2% to 82.4%, and also has the highest accuracy in classifying 'Lung', while if adding 'COVID' and 'Viral' synthesis, it has the highest overall validation accuracy of 90.35%, with the highest prediction accuracy over 'COVID' and 'Normal' classes, which are 95.2% and 91.6% respectively. Lastly, by adding all kinds of synthetic images except for the 'Normal' class, it shows the highest prediction accuracy of 97.2% over the 'Viral' class.

All training models are under the same configuration mentioned in Sec.3.3.

Table 5: Classification Experiments on Data Combination

Data	Validation Acc[a]	Prediction Acc per Class[b]			
		COVID	Lung	Normal	Viral
Real only	86.41%	92.4%	75.2%	82.4%	92.0%
Real + 3,634 COVID synthesis	88.04%	90.4%	**82.4%**	89.2%	95.2%
Real + 1,238 Lung synthesis	86.05%	88.8%	81.2%	88.8%	90.4%
Real + 5,905 Viral synthesis	89.09%	94.4%	80.04%	87.6%	94.8%
Real + 3,634 COVID & 5,905 Viral synthesis	**90.35%**	**95.2%**	80.0%	**91.6%**	96.4%
Real + 3,634 COVID&1,238 Lung&5,905 Viral synthesis	89.35%	94.4%	81.2%	89.2%	**97.2%**

[a,b] Different from Sec.4.1, here we use $\#correct/\#prediction$ in validation set[a] and testing set[b] as accuracy.

B Implementation Details

B.1 Training Detail of Conditional GAN

We trained the conditional GAN with labels 'COVID':0, 'Normal':1, 'Viral':2, 'Lung':3. The training configuration we followed the configuration for 256x256 resolution data in the paper of Karras et al. (2020).

Training and inference used the same workstation that contains 2 GPUs of NVIDIA GeForce RTX 2080 Ti with 11GB memory, and we repeatedly trained 25 million images for each model with the same training configurations.[5]

[5]Each model is trained for over 6 days on average. More statistics of training time can be referred to official document: `https://github.com/NVlabs/stylegan2-ada-pytorch#expected-training-time`.

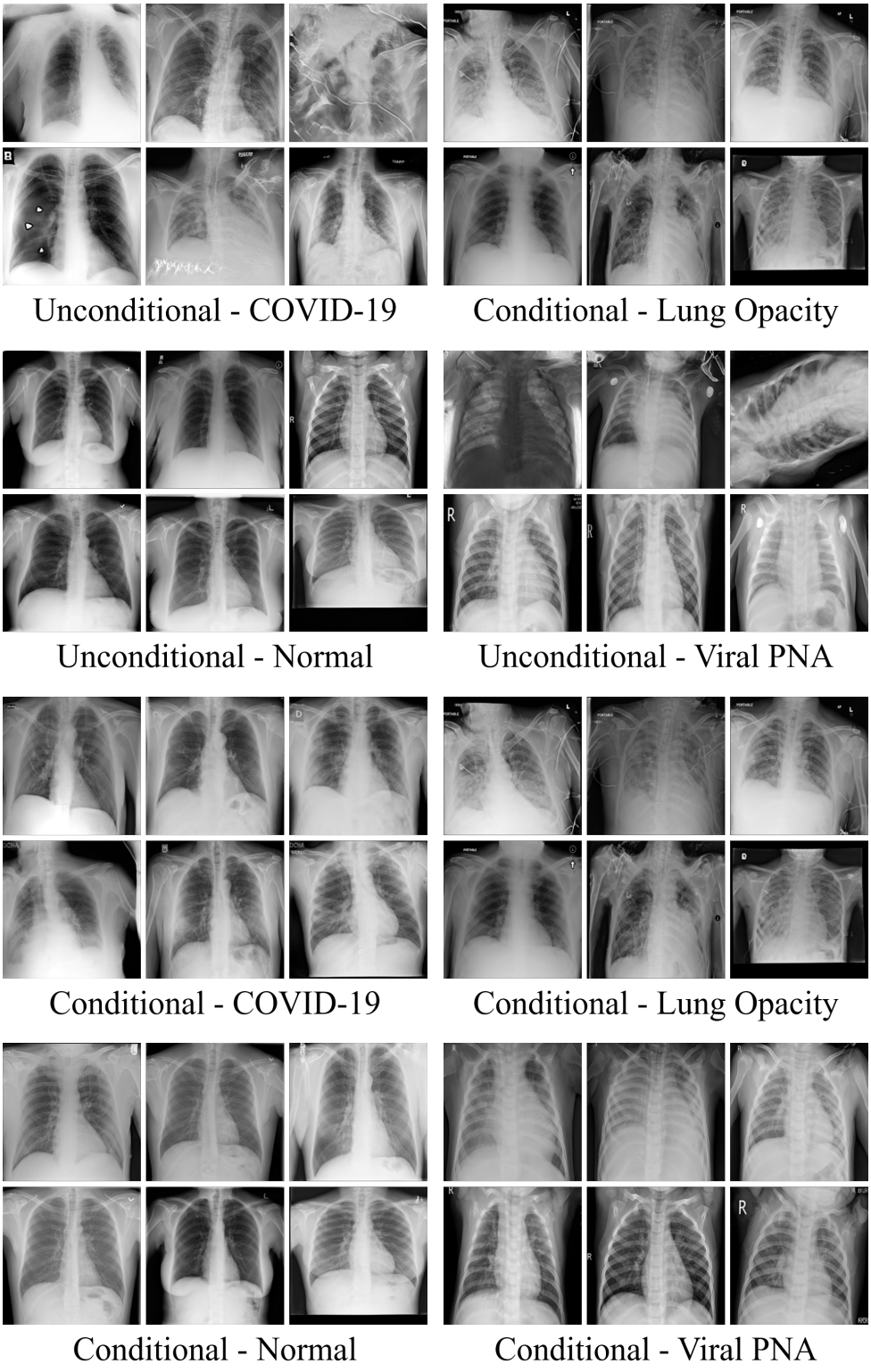


Figure 6: Unconditional vs conditional generation.

C Members and Contribution

Yingshu CHEN data preprocessing, code implementation (GAN) and integration, report writing (Abstract, Sec 1, 2, 3.1, 3.4, 4.1, 4.3, 5, Appendix), presentation.

Wing Hei SUM data collection, code implementation (evaluation), report writing (Sec 4.1, 4.3), presentation.

Ho Pan IP data preprocessing, code implementation (classification), report writing (Sec 2, 3.3, Appendix), presentation.

Zipeng WU code implementation (data visualization), report writing (Sec 2, 3.2, 4.2), presentation.