UNSW
SYDNEY

Group 3

UNSW COMP9491 Applied AI - Identification and Localization of COVID-19 Abnormalities on Chest Radiography Report

Lihuan Li
CSE
z5139949@unsw.edu.au

Zhihan Qin
CSE
z5290141@unsw.edu.au

Yang Ma
CSE
z5100105@unsw.edu.au

August 13, 2021

Introduction

COVID-19, a dangerous pandemic that was initially detected in December 2019, has spread worldwide rapidly. In addition to common flu symptoms such as cough, fever, or sore throat, a chest X-ray (CXR) can also facilitate diagnosing such infection by indicating opacity areas to radiologists. However, different from natural images in public datasets such as MS-COCO[9] or ImageNet[10], chest X-ray images have characteristics like high similarity, low contrast, and lack of interpretability.

In recent years, computer vision methods have been widely utilized in pathological detection and analysis of medical images, especially deep-learning-based methods such as convolutional neural networks have achieved state-of-the-art performance on medical imaging. Even so, limitations like the inefficiency of data usage and low generalization ability[13] are still reasons why deep learning methods cannot be widely applied in clinical medicine. Moreover, due to privacy concerns, drawbacks of chest X-ray datasets including scarcity of quantity and variety of data samples[12] increases the difficulty of analyzing COVID-19 diagnosis.

In this project, we aim to utilize a variety of artificial intelligence approaches to implement opacity detection and classification in terms of labeled chest X-ray images. There are four types of diagnosis, i.e. negative for pneumonia, typical, indeterminate, and atypical. Our experimented methods can be divided into three parts, including data enhancement, opacity detection, and diagnosis classification. For data enhancement, we have reproduced the implementations on DGM[1]. And we have attempted YOLOv4-Tiny and RetinaNet with various preprocessing methods and backbone models for both single-label and multi-label opacity detection. For classification, we have experimented on the Siamese Networks for feature extraction followed by machine-learning-based clustering/classifiers, as well as Covid-Net, our deep-learning-based approach on classification.

Related Work

To cope with the scarcity of data samples, generative adversarial networks (GANs) are frequently utilized in data augmentation and enhancement by generating various synthetic CXR images. CovidGAN[12] utilizes Auxiliary Classifier Generative Adversarial Network (ACGAN) for data augmentation to improve the performance of the binary classification of Covid diagnosis by popular CNN models, which improves the accuracy from 85% to 95%. Tang et al.[1] disentangles the opacity areas from abnormal CXR images to increase the accuracy of classification and interpretability of CXR images.

For the single-label detection of lung opacity, researchers have experimented with various network architectures. Saiz et al.[15] introduced a Single Shot Multibox Detector (SSD) model with a backbone of VGG-16 and trained on CXR images after applying CLAHE[14], which achieves 94.92% of sensibility and 92% of specificity. Tang et al.[1] experiments with RetinaNet to detect opacity and explains the difficulty to detect the mild opacity severity. Moreover, Faster R-CNN[20] employs region proposal networks to regress predictions to more accurate detection boxes.

For the classification of CXR diagnosis, different strategies are adopted to build the networks. Wang et al. designed a deep convolutional neural network named COIVD-Net[16] and constructed it under a machine-driven design exploration strategy which improves the sensitivity to positive Covid19 cases. The Siamese Networks[2] applies metrics-based few-shot learning by passing a pair of images into

two identical pre-trained CNN models to generate feature embeddings being classified in machine learning approaches. Park et al.[17] utilized a vision transformer with chest X-ray feature corpus and produced promising results on the diagnosis of COVID-19.

As for methods that increase interpretability, Grad-CAM[18] proposes gradient-based localization to highlight the areas which are detected by its CNN model and make object detection and classification more explainable. Furthermore, Ye et al.[19] applies Probabilistic-CAM pooling to "leverage" such localization ability, which can explicitly emphasize the region of interest on CXR images.

Methods

Data enhancement

Different from natural images that we can easily recognize and classify, diagnosis based on chest X-ray images requires professional medical knowledge. And it sometimes requires interdisciplinary approaches to make the results of medical image detection or classification by deep learning methods both accurate and interpretable. For example, Fig. 1 shows that the area covered in the yellow box is opacity but it is difficult to detect or explain for those who are without professional knowledge. And since a study on the interpretability of deep learning on chest X-ray images is still needed[3], we aim to use deep learning methods to achieve data enhancement as well as the improvement on the interpretability of our dataset. To fulfill our aim in data enhancement, we propose to work on a *deep disentangled generative model* (DGM)[1] and reproduce its results shown in the paper.

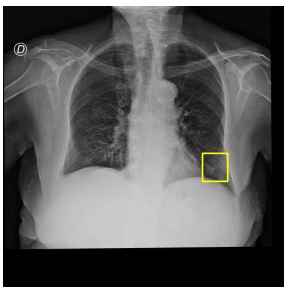


Fig. 1

Model Structure

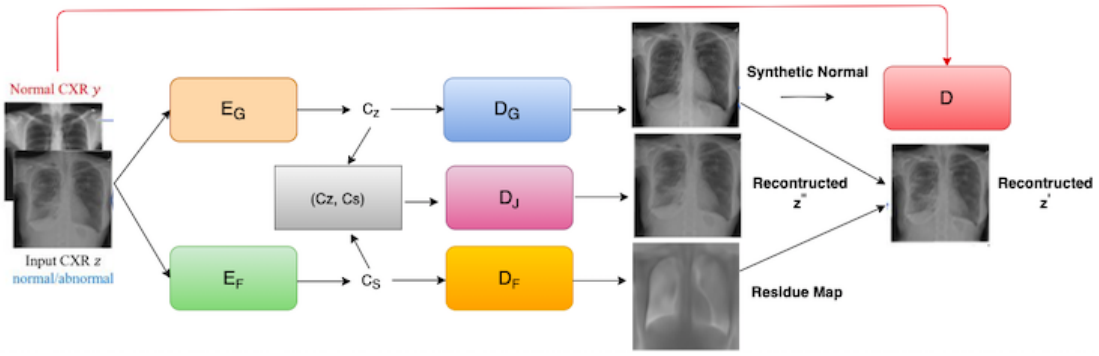


Fig. 2 The overall model structure of DGM

From Fig. 2, we can see that DGM has 6 components. E_g and D_g encode and decode the original images to generate synthetic normal images, E_f and D_f encode and decode original images to generate disentangled residue map, and D_j receives encoded features of both E_g and E_f to generate reconstructed original images. Discriminator D judges the quality of synthetic normal images from D_g. As the DGM paper does not provide a detailed implementation process and the unofficial implementation (https://github.com/YeongHyeon/DGM-TF) using the MNIST dataset that we refer to

is very different from the original DGM structure, we split our method into several stages of improvements.

Initial Generator Structure

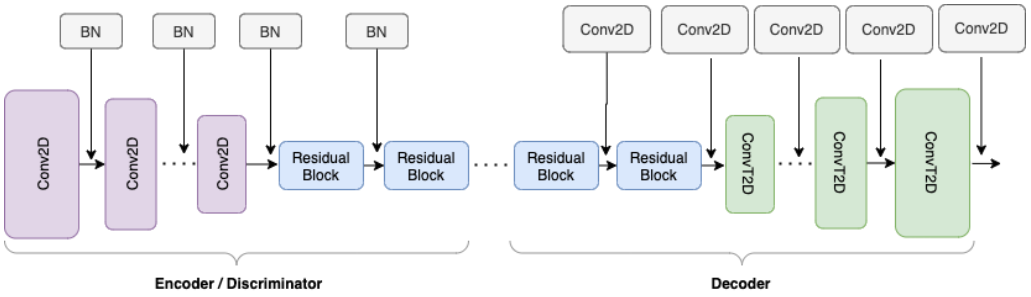


Fig. 3 Initial structure of our reproduced DGM

As fig. 3 shows, the encoder has a series of 2-dimensional convolutional layers and residual blocks and the decoder has several residual blocks followed by a set of 2-dimensional convolutional / transpose convolutional pairs. During encoding, we also apply batch normalization [4] after each convolutional layer to accelerate training. However, it causes *"mode collapse"* where the generator tends to map from one training sample to all synthetic images in a mini-batch.

Improvements on Generators

Based on the aforementioned GitHub repository and the parts of the original paper that were not explained in detail, we have done the following improvements. The final structure of the generators is displayed in fig. 4.

Instance and Layer Normalization To address mode collapse, we removed batch normalization as it calculates the mean and variance of a whole batch, which makes the generator confused about the features of images that have already been similar. Instead, we add instance normalization [5] in the encoder and layer normalization [6] in the decoder to lead the generator to learn specific features on each training sample.

Upsampling Though the generator can learn a correct mapping of each data sample, generated images are still blurred because the transpose convolutional layer with a stride value of 2 causes *"chessboard artifacts"*. We propose a 2-dimensional upsampling with a more complicated interpolation algorithm to replace transpose convolutional layers in decoders.

Adaptive Instance Normalization (ADAIN) We aim to produce better reconstruction results as reconstructed images influence the weights in both E_g and E_f that each generates synthetic normal images and residue maps. We consider encoded features C_z as content and C_s, as style, then utilize ADAIN[8] to fuse and integrate these two features before passing them into the decoder D_j.

Skip Connection To generate higher quality residue maps, we also extend a connection from the last convolutional layer of encoder E_f to the first upsampling layer in D_f as the generator is too deep so that it loses some important features while encoding. The connection method is a simple element-wise sum.

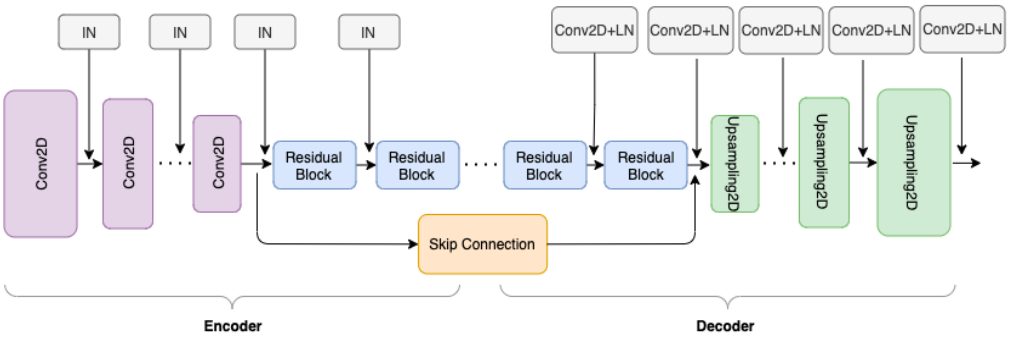


Fig.4 The final structure of our reproduced DGM

Multi-scale Discriminator

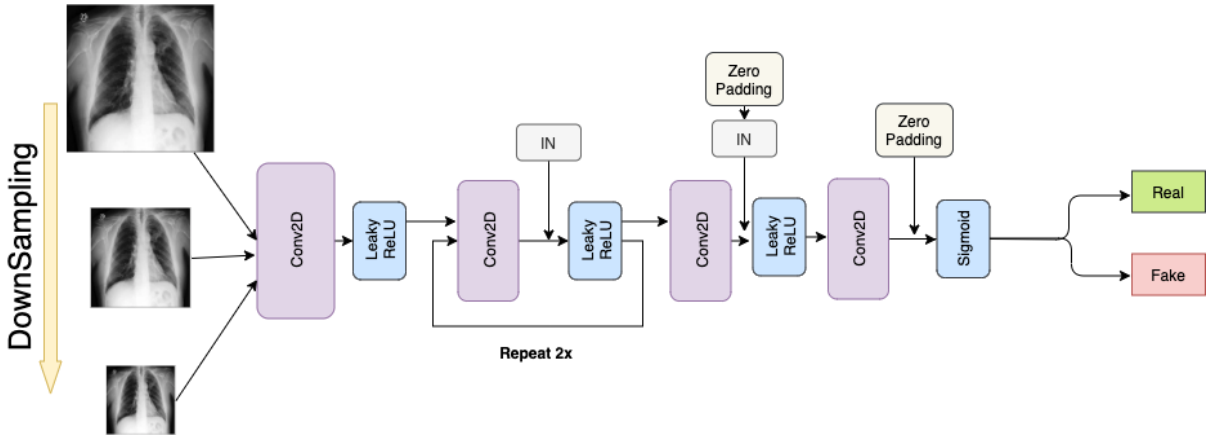


Fig.5 Structure of multi-scale discriminator

To address the scenario that discriminator loss quickly drops to zero which leads to gradient vanish of generators, we propose a multi-scale discriminator[7] shown in Fig. 5. Specifically, images are scaled into 224*224, 112*112, and 56*56. Then, the discriminator learns to minimize the loss in all scales. According to [7], different scales of images give the discriminator a "large receptive field" to make sure it can learn both the global view and details of an image.

Detection

In the beginning, we only designed a multi-label detection part to solve this problem. However, due to the poor performance of multi-label detectors, we split the task into single-label detection and classification. A detailed discussion is presented in the next section.

For the detector models, we have selected the most popular models, including SSD, Yolov4-Tiny, Faster-RCNN, and Retinanet. Since the SSD model and the Faster-RCNN model are taken charge of by our quitted teammate, we will not go into details here.

Both Yolov4-Tiny[27] and Retinanet[28] are one-stage detectors. Yolov4-Tiny is a tiny version of Yolov4, which is composed of 38 layers compared with 175 layers of Yolov4. Since the opacity area does not have obvious visual features, there is no need to use a deeper network. Though the yolov4-Tiny is not competitive in precision compared with other large-scale networks, it takes a quite

short time to train and is easy to experiment with. For Retinanet, the network structure has done a great balance of precision and speed, which outperforms the other one-stage detectors. The application of focal loss deals with the imbalance of positive and negative samples and thus, it is very suitable for our case.

For single-label detection, we have designed three types of comparison to figure out the working mode of each detector. The comparisons are conducted between different algorithms, different backbones and models applied CLAHE or not.

Classification

To better compare and discover the most suitable approach to handle the task, we deployed various models with different methodologies. For machine learning-based approaches, we propose a two-stage model where features are extracted and embedded in stage one and final classification is achieved in stage two. Specifically, for stage one, we utilize a metrics-based few-shot learning approach named Siamese Networks. And for stage two, we make comparisons with three machine learning classification/clustering methods: Support Vector Machine (SVM), K-Means Clustering, and Gaussian Mixture Model. The detailed model structure will be elaborated in feature extraction, feature dimensionality reduction, and classification. For deep learning-based approaches, as the aim of our task is to classify Negative for Pneumonia, Indeterminate Appearance, Typical Appearance, and Atypical Appearance, and considering the difficulties in medical images classification, covid-net utilises the PEPX to maintain representational capacity and demonstrates better sensitivity(recall) than VGG and ResNet with COVIDx Dataset.

Two-stage classification (A Machine Learning-Based Approach)

Feature extraction

Popular image classification models such as VGG, Resnet, or Faster-RCNN[20] are trained on much larger datasets and the features in those datasets are very different from chest X-ray images. Besides, there are only 6334 training samples in our dataset, which makes transfer learning less effective.

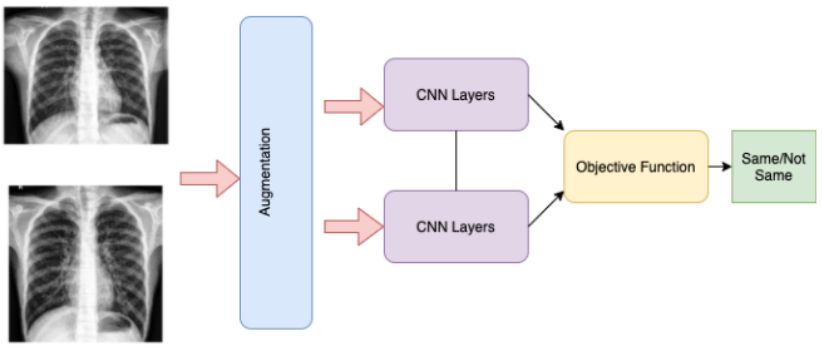


Fig.6 Structure of the Siamese Networks

Therefore, we apply Siamese Networks[2] for feature extraction. As Fig. 6 shows, the input of the model is a pair of images, which are passed into 2 identical CNN models with shared weights. If the images in the pair belong to the same class, CNN models will generate similar feature embeddings. If not, the feature embeddings will be very different.

Feature dimensionality reduction

Since each feature embedding has 1000 dimensions, feature dimensionality reduction is necessary as data samples in such high dimensions are very sparse and can cause a "curse of dimensionality". In our experiments, we have applied Principal Component Analysis (PCA) with t-distributed Stochastic Neighbor Embedding (t-SNE), and Linear Discriminant Analysis (LDA). PCA reduces the dimensions of data by maximizing the variance and minimizing the loss caused by dimensionality reduction. Then, t-SNE models the distribution of nearest neighbors of each data point processed by PCA to eventually generate feature embeddings that are ready to be fed into the classifiers or clusters. PCA and t-SNE are both unsupervised, whereas LDA is supervised. It reduces the dimension of labeled data samples by maximizing their inter-class variance and maximizing their intra-class variance. In our experiments, we have made thorough comparisons between these two groups of dimensionality reduction methods.

Machine learning-based Classification

We experimented on supervised machine learning classifier SVM and unsupervised K-Means clustering and Gaussian Mixture Model. We utilize these models to evaluate the quality of feature embeddings as machine-learning-based approaches are more critical of extracted features.

End-End Classification (A Deep Learning-Based Approach)

PEPX and long-range selectivity

Covid-Net[16] is a deep convolutional network designed concerning a human-machine collaborative design strategy. It's the first neural network to introduce lightweight projection-expansion projection-extension design. Such a pattern is discovered by the machine-driven design exploration strategy that enhances representational capacity and reduces computational complexity. As shown in Figure 7, it also demonstrates a "selective long-range connectivity structure" on the network. It also promotes representational capacity and simplifies the network at the same time.

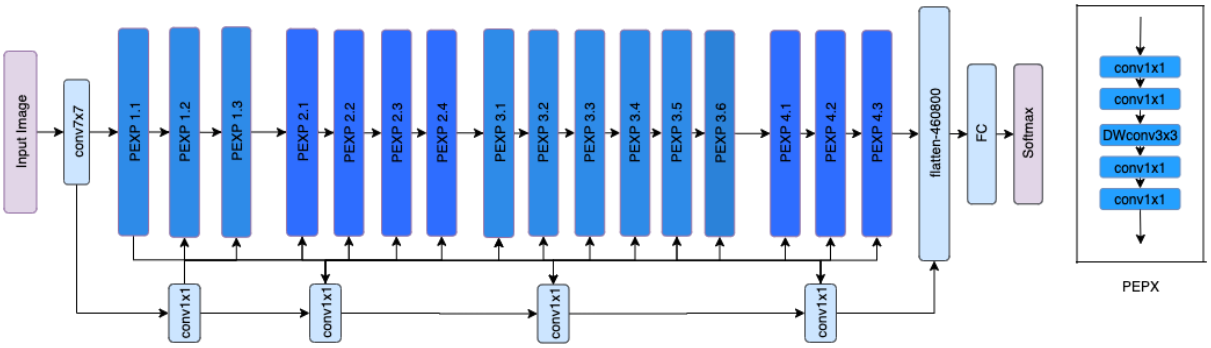


Fig.7 The structure of COVD-Net

COVID-Net auditing via explainability

Since the model is for diagnosis purposes, it is essential to study the network with transparency. COVID-Net is validated with an explainability-driven audit to make sure it is making decisions based on relative features, where GSInquire[23] is utilized in the machine-driven exploration strategy.

Experimental setup

Dataset

The dataset[24, 25] consists of 6334 CXR images, with bounding box locations and labels. The overall distribution is shown in Table 1. For RetinaNet, the dataset is split to 4434 and 1900 for training and testing sets respectively. For YOLO, the dataset is split into 2994 (excluding negative samples) for training and 1900 for testing. As for classification tasks, the dataset is split into 4500 and 1834, as shown in Fig.8. The labels are Negative for pneumonia, Indeterminate Appearance, Typical Appearance, and Atypical Appearance.

As we can see in table 1, the dataset is imbalanced, where the indeterminate class is one-third of the typical class, and the atypical is one-seventh of the typical case.

Hierarchy	Task	Quantity	Distribution
study-level	Classification of CXR Diagnosis	6054	1676(28%) - Negative for Pneumonia 2855(47%) - Typical Appearance 1046(17%) - Intermediate Appearance 474 (8%) - Atypical Appearance
image-level	Opacity (Single-class) Detection	6334	2040(32%) - None Opacity 4294(68%) - Opacity
bbox-level	Multi-class Detection	8157	6034(74%) - Typical Appearance 1494(18%) - Intermediate Appearance 629 (8%) - Atypical Appearance

Table 1. Table for data distribution

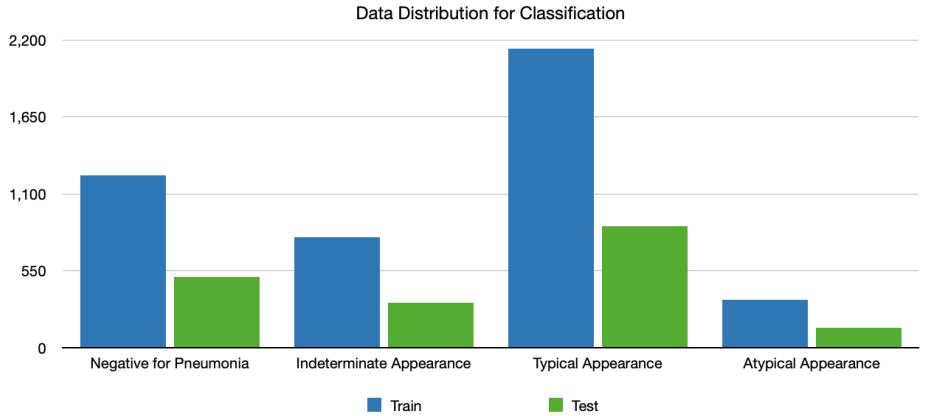


Fig.8 Bar chart for Data distribution in the classification task.

Evaluation Metrics

We use accuracy, recall, precision, F1-score, Silhouette Score and mean average precision as evaluation metrics in this project. The accuracy provides a direct instinct on the performance.

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictoins}$$

The sensitivity(recall) provides the measure of the proportion of people who tested positive among those who have the disease.

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

The positive predicted value(precision) is a measure of giving a positive test result, the probability that the individual will actually have that specific disease.

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

F1-Score is the weighted average of precision and recall. It is particularly valuable when we have uneven class distribution.

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

Silhouette score is applied to analyze the clustering-based approach, it measures the mean of intra-cluster versus inter-cluster distance.

$$SilhouetteScore = \frac{b - a}{max(a, b)}$$

Mean average precision is calculated by taking the mean of the average precision values of all the classes at IoU >= 0.5, where the IoU is the Intersection over Union.

Result and discussion

Data enhancement

In this section, the intermediate results of each improvement are displayed. And since the results can only be visually inspected, we only provide a qualitative evaluation.

Initial Generator Structure Fig. 9 (a) shows our results of synthetic normal images by the initial generator structure on the validation set. We can see that all generated images look very similar and unclear, which is what we call *"mode collapse"*.

Instance Normalization Fig. 9 (b) shows the results after we replaced batch normalization to instance normalization compared to (a). We can see that the model started to learn correct mappings from each data sample to its corresponding synthetic images. This is because, in style transfer, instance normalization learns the feature of each data sample instead of a whole batch.

Upsampling (c) shows the improved quality of synthetic images compared to (b) in Fig. 9. While we were implementing this method, we switched our dataset to cropped images to let the model only focus on the lung areas. We can see that in (b), the generator cannot learn details in a CXR image. However, it can learn much clearer images including human ribs in (c) after we replace transpose convolutional layers with upsampling.

Skip Connection Although the model could generate clear synthetic images after applying upsampling, it was still not able to learn salient residue maps. To address this issue, a shortcut is added that skips the residue blocks of both the encoders and decoders. As examples shown in Fig. 10, the areas pointed by red arrows at the right column are opacities in the corresponding images at the left. We infer that the skip connection provides the decoders with more complete information of the original images as there might be a loss of some information during encoding.

However, even though some good results have been successfully demonstrated by our reproduced model, the residue maps are still not as obvious as those in the original paper. And by inspecting (c) in Fig. 9, we found that the generator still tends to learn the original images instead of the normal ones. We suspect that though we have modified the discriminator multi-scale, it still cannot accurately judge whether a chest has opacities based on given normal samples. To improve the discriminator to be able to detect such subtle differences in image features, we leave our improvements such as PatchGAN[30] based discriminator structure in the future.

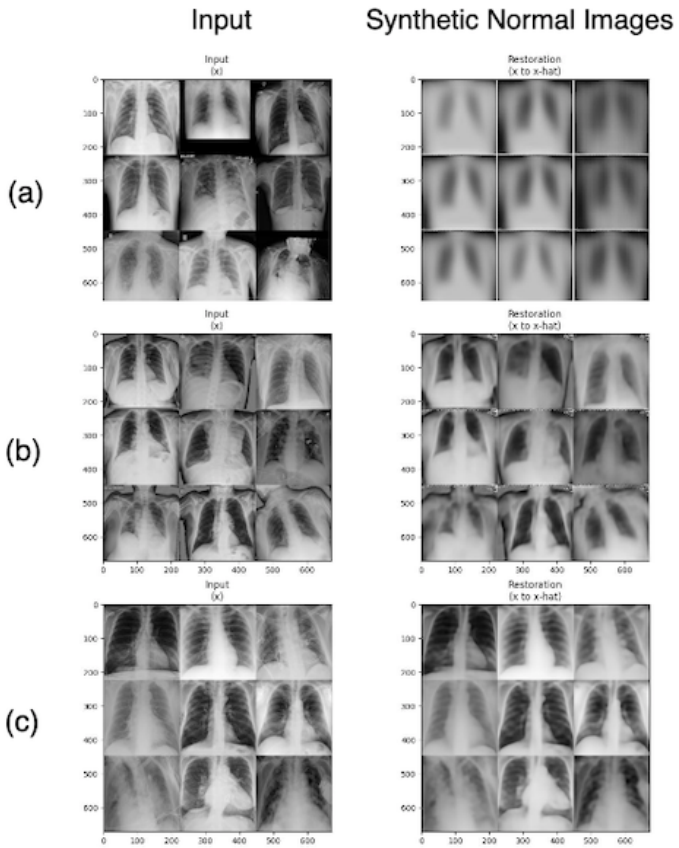


Fig.9 Improvements on synthetic normal images

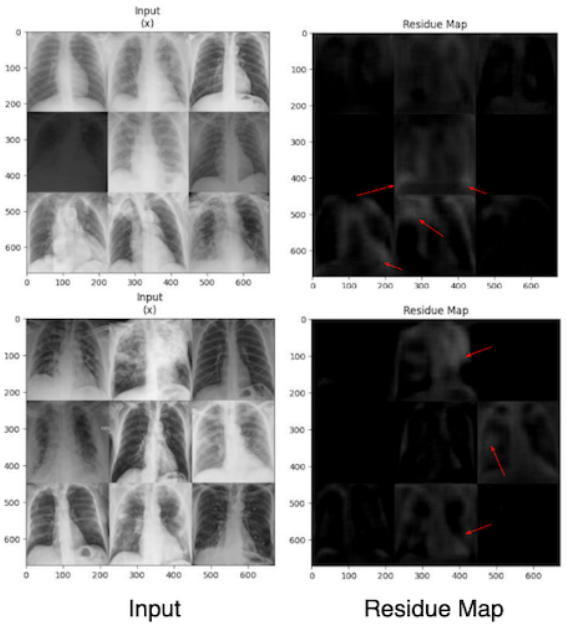


Fig.10 Reproduced results of the disentangled residue maps

Detection

Multi-label detection of multiple opacity appearances

It is well known that the most intuitive way to solve the problem is to apply a multi-label detector. Therefore, in the beginning, we trained a Yolo model to detect three types of opacity appearance. Fig 11 displays the trend of training loss and the mAP trend on the validation set. The best mAP was achieved at iteration 8700 and it stopped oscillating in later iterations. Table 2 lists out the average precision of each class. In this case, the multi-label detectors perform poorly as all of the mAPs are less than 20%. The results display the great bias within the detectors. Specifically, the average precision of typical appearance exceeds 40% while for intermediate and atypical ones, the average precisions are less than 10%. This is mainly caused by the imbalance of the given dataset and the similarity between different appearances. To deal with this problem, we split the task into two parts, single-label detection of opacity, and classification of opacity appearances, since different tasks have different focuses. In this way, our common goal is to figure out a combination of classification and detection methods that outperforms the multi-label detectors introduced here.

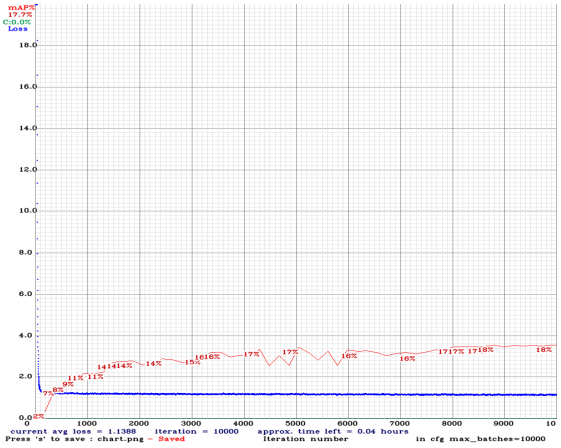


Fig. 11 loss trend of yolov4-tiny of 3 types of appearances

Model	AP@0.5 (TA)	AP@0.5 (IA)	AP@0.5 (AA)	mAP @0.5
yolov4-tiny	0.436	0.043	0.054	0.178
ssd	0.429	0.037	0.045	0.170
faster-rcnn	0.400	0.000	0.017	0.140

Table 2. mAP comparison of different models

Single-label detection of opacity

In this section, multiple sets of experiments are conducted. Performance comparisons are made between different model structures, different backbones, and whether applied CLAHE or not.

Yolov4-Tiny Vs Retinanet-Resnet50

A confidential score describes the confidence or probability of a detected object belonging to a class. Table 3 shows the number of boxes of which the confidential score is greater or equal to a given value. The average confidential score of the objects detected by Retinanet is much higher than the ones detected by Yolo. To some extent, this indicates that the overall detection ability of Retinanet is greater than Yolo except for producing more false-positive outputs which result in lower precision. In addition, the table helps to determine a reasonable threshold of the confidential score for each of the detectors based on the estimated number of output bounding boxes for the validation set. In this case, the thresholds for Yolo and Retinanet are 0.05 and 0.50 separately. Fig. 12 visualizes the typical situations which display the difference between these two models. In this case, Yolo and Retinanet are somehow complementary to each other in detecting the opacity of different scales.

model/conf_thresh	0.05	0.25	0.50	0.75
Yolov4-Tiny	1813	1170	69	0
Retinanet-Resnet50	29902	6450	3596	1371

Table 3. The number of boxes of which the confidential score is greater or equal to n n∈{0.05, 0,25, 0,50, 0.75

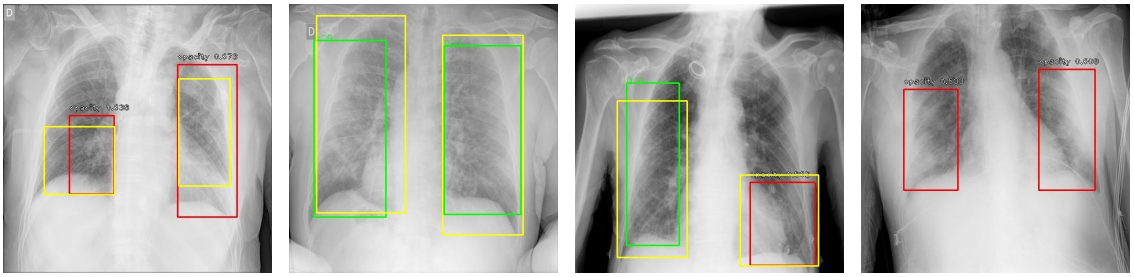
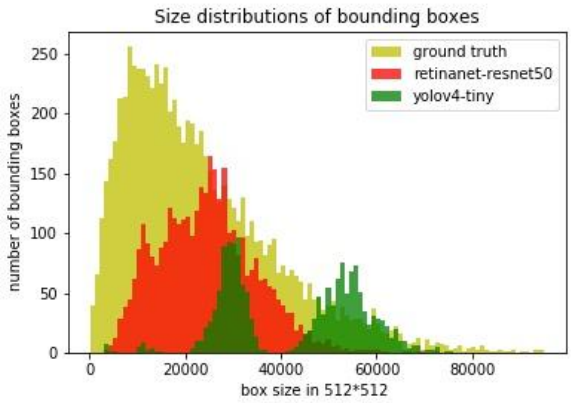


Fig. 12 Typical situations of the outputs of Yolov4-tiny and Retinanet-resnet50. (Yellow bounding boxes for the ground truth, green ones for yolov4-tiny and red ones for Retinanet-resnet50)

On the other hand, the sizes of bounding box annotations are widely distributed. However, by contrast, the size distribution of those produced by Yolo and Retinanet is quite compact as displayed in Fig. 13. To be specific, both of these two detectors fail in detecting smaller opacity areas. For example, in Fig. 14, opacity areas are detected in none of the four images. We tried yolov4-tiny3l, an extended model of yolov4 tiny, which adds an output layer to handle a wider range of objects of different sizes. Unfortunately, this did not work and even performed worse than the original structure.



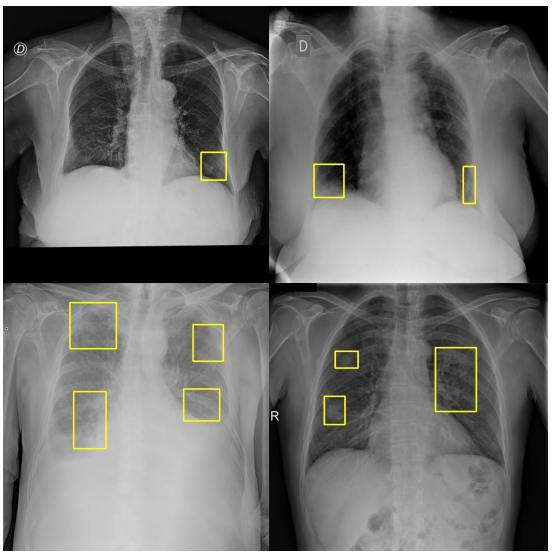


Fig. 13 (left) Size distributions of bounding boxes
Fig. 14 (right) Failure cases of small opacity areas

Retinanet-Resnet50 Vs Retinanet-Resnet101

For the comparison of different backbones, we experimented with resnet50 and resnet101 which are the most common backbones used by Retinanet. According to the evaluation criteria, there is little difference between these two models, and therefore, considering the training efficiency, Resnet50 is a better choice. More importantly, the training loss of Resnet 50 drops more quickly Fig. 15. Due to the limitation of GPU resources, 20 epochs are trained for each model of Retinanet. Therefore, the models might achieve better performance when trained until convergence.

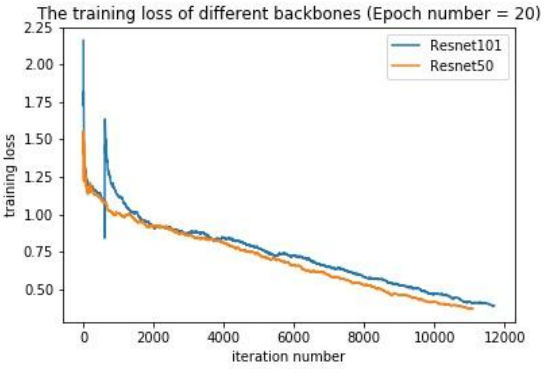


Fig.15 Training loss trend of different backbones

Yolov4-Tiny Vs Yolov4-Tiny (with CLAHE)

Overall, the performance of a single model is not improved after applying CLAHE and moreover, the performance of Retinanet even becomes a little bit worse. Nevertheless, models with such a preprocessing method are meaningful. Since some of the areas get enhanced with CLAHE, it becomes easier for the detector to locate them. Fig 16 displays four situations that the combination of the two detectors can do a better job.

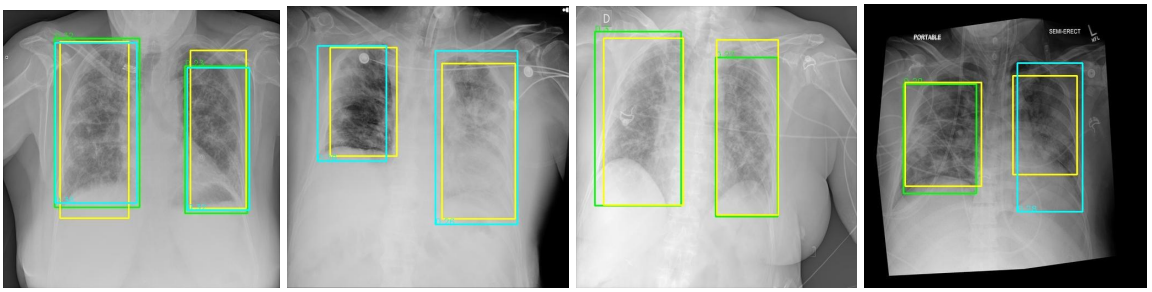


Fig.16 Typical situations of the outputs of yolov4-tiny and yolov4-tiny (with CLAHE). (Yellow bounding boxes for the ground truth, green ones for yolov4-tiny, and cyan blue ones for yolov4-tiny (with CLAHE))

Quantitative statistics of the six models that we have trained are listed in table 4. Here the precision and recall are calculated with the confidence rate threshold set to 0.05. In this table, the yolov4-tiny outperforms the other five models. More importantly, the significant diversity in Precision and Recall indicates that it should be a good idea to try model ensembling.

Model	Precision	Recall	mAP@0.5
Yolov4-Tiny	0.50	0.51	0.465
Yolov4-Tiny (with CLAHE)	0.46	0.52	0.465
Retinanet-Resnet50	0.11	0.73	0.387
Retinanet-Resnet50 (with CLAHE)	0.11	0.72	0.359
Retinanet-Resnet101	0.11	0.75	0.387

Retinanet-Resnet101 (with CLAHE)	0.10	0.76	0.376

Table 4. Performance of all the experimented detectors

Classification

To be able to both quantitatively and qualitatively evaluate our classification models, we split our training set of 6334 into 4500 for training and 1834 for testing.

Two-stage classification (Machine Learning Based)

To ensure the class labels of data samples are equally distributed, we further randomly sampled 1372 (343 samples in each class) for training and 560 (140 in each class) for testing.

Feature Dimensionality Reduction

In our implementation, we choose pre-trained Densenet121[11] as our backbone of Siamese Networks and generate a feature of 1000 dimensions for each data sample. In this section, we will compare the results of feature dimensionality reduction by PCA & t-SNE and LDA. Specifically, we downsize the feature dimensions to 180 by PCA and further downsize it to dimensions of three. In another branch, we downsize the feature dimensions directly to three by LDA. The reason why we choose a dimension value of three is that it is more convenient for us to demonstrate the visualization of processed data samples in a 3D figure.

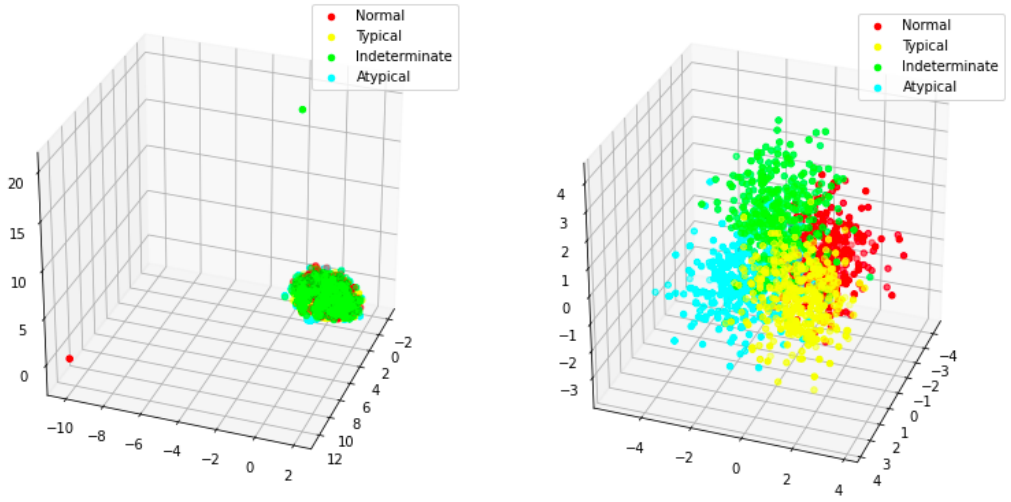


Fig.17 Visualization of Dimensionality Reduction by PCA & t-SNE (left) and LDA (right) on the training set

Fig.17 shows the comparisons of dimensionality reduction between PCA & t-SNE and LDA on the training set. We can see that LDA can correctly learn and transform the data features from high dimensions to low dimensions, although the inter-class distances are quite close. However, PCA & t-SNE are not able to downsize the feature dimensions.

Classification

We make comparisons among K-Means Clustering, Gaussian Mixture, and SVM on the testing set. Table 5 shows the quantitative results of these three clustering/classification methods and we can see that SVM produces the best results on all evaluation metrics except for Silhouette Score as it is utilized to evaluate clustering methods.

Method/Evaluation	Silhouette Score	Accuracy	Recall	F1	Precision
K-Means	0.40	24.46%	0.24	0.24	0.25
Gaussian Mixture	0.39	31.07%	0.31	0.31	0.31
SVM	N/A	**34.64%**	**0.35**	**0.35**	**0.35**

Table 5. Quantitative evaluation on clustering/classification methods by K-Means, Gaussian Mixture and SVM.

Analysis of Poor Performance

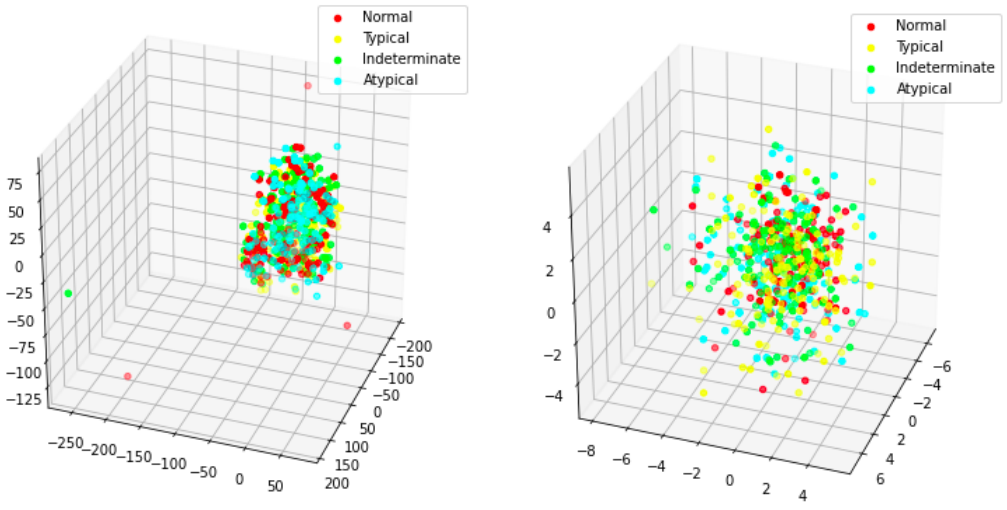


Fig.18 Visualization of Dimensionality Reduction by PCA & t-SNE (left) and LDA (right) on the testing set

To find the reasons for poor performance, we plot the feature dimension reduction results on the testing set in Fig.18 and it is apparent that the issue happens on the feature embeddings from Siamese Networks. Then, we inspected the dataset of the original few-shot learning repository (https://github.com/shruti-jadon/Covid-19-Detection-Few-Shot-Learning) and found the data utilized is clean and less varied than ours. We have also made some comparisons of these two datasets by visual inspection and found that in Fig. 19, human bodies in our dataset ((a) and (b)) are in quite different postures while those in the original repository ((c) and (d)) are much more similar.

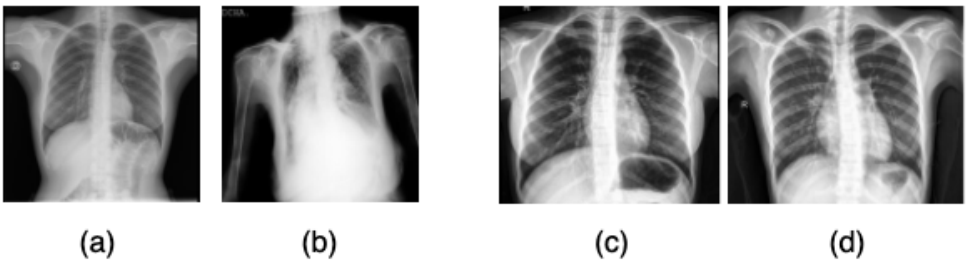


Fig.19 Different characteristics between data samples in our dataset [24, 25] and few-shot learning[2]

Therefore, we can conclude that our feature extractor, the Siamese Network, cannot learn the embeddings accurately on our dataset consisting of various chest X-ray images. Besides, variability and scarcity of data samples also make it difficult for the model to generalize.

End-End Classification (Deep-Learning Based)

ResNet

During the project, we also deployed the ResNet for better comparison. We trained the model with batch size 64 and the model converged at epoch 12. It achieved an overall accuracy of 59%, and an accuracy of 71% for negative for pneumonia, 76% accuracy for typical covid-19 appearance. However, it failed to learn the atypical and indeterminate class with an accuracy of around 10%. This is due to the limited dataset sizes on atypical and indeterminate classes and the difficulty of medical image classification.

	Overall	Typical	Atypical	Negative	Indeterminate
Accuracy	59%	76%	13%	71%	11%

Table 6. The accuracy overall and for each class.

COVID-Net

The Covid-net model was adjusted based on the official code provided in the paper[16]. The code was implemented and tested with python 3.6, Tensorflow 1.15, but the colab only supports python 3.8. To adapt to the environment we installed miniconda on colab to have the environment ready for experiments.

The pre-trained model was utilized for 3-class classification: negative for pneumonia, pneumonia, and covid. When we fine-tune the model, the model would break due to GPU graphics memory issues if the model structure grows too large(even with the config.gpu_options.allow_growth configuration). So the classification layer had only one dense layer with 512 nodes, any size larger than that would break the graphics memory. For the dataset, we are training a pre-trained model that was trained on a relatively large dataset, with small datasets. Moreover, our dataset was a subset of the dataset that was used for the pre-trained model, with different labels. The original dataset COVIDx was a combination of 5 different datasets[21, 22, 24, 25, 26], our training set is from one of them.

For transfer learning methodologies, we are supposed to simply train the classification layer. However, due to the memory issue, we tried to unfreeze the pre-trained model. It turns out that the model failed to learn atypical and indeterminate classes and that the training ruined the weights for

negative and typical appearance classification(shown in Fig.20). This is due to the limited training set and the newly introduced labels.

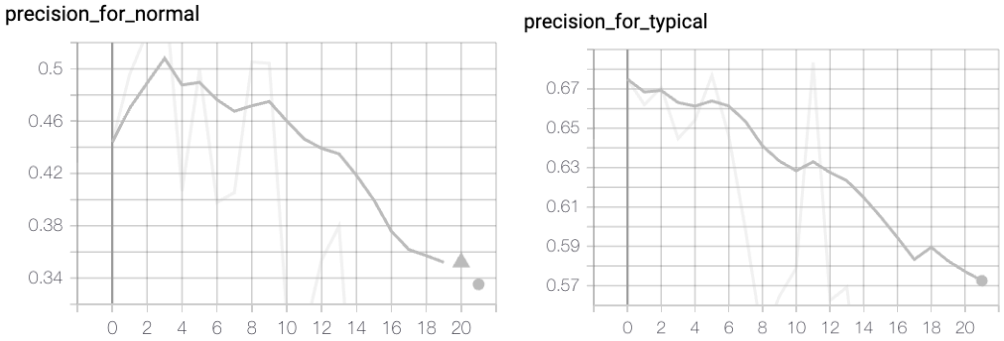


Fig.20 RHS - precision curve for negative for pneumonia class, LHS - prevision curve for typical appearance class.

Freezing the pre-trained model and training the classification layer would yield an overall accuracy of 61%(shown in Fig.21). However, it also failed to learn the atypical class as the typical and atypical were both classified as covid in the pre-trained model and the training set for atypical is relatively small. Moreover, it showed some learning for the indeterminate class, but the learning curve of every class vibrates within an interval(shown in Fig.21). This is due to the fact that the classification layer is too small and incapable of handling such tasks.

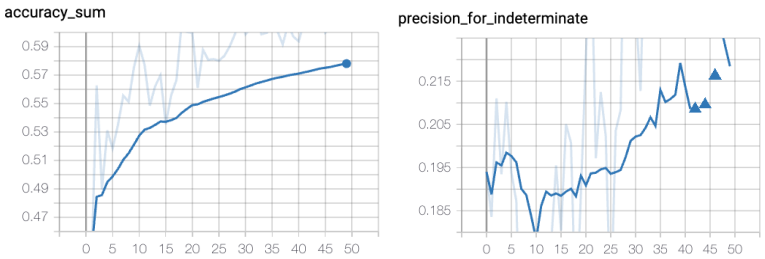


Fig.21 The curve on the left shows the learning curve for accuracy overall, the curve on the right demonstrates the precision curve for indeterminate.

Thus, to boost the performance we upsampled the training set with sample weights(0.148:0.232:0.085:0.533). With sample weights applied, the model was able to learn the atypical and indeterminate(shown in Fig.22), and the recall for typical and atypical classes achieved 55% and 72% respectively. However, as mentioned before the performance is limited by the classifier.

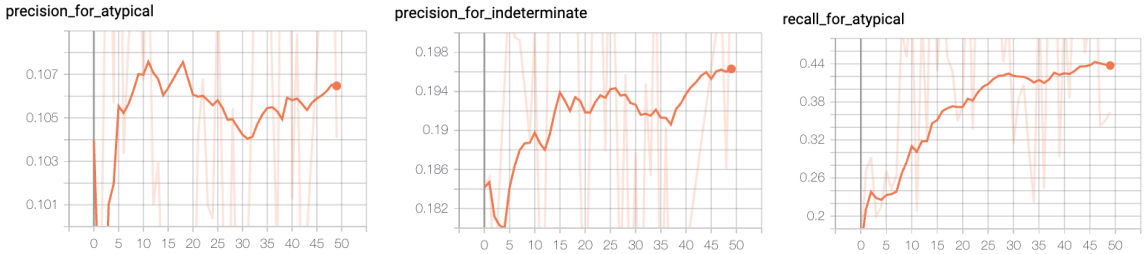


Fig.22 from left to right, are the prevision curve for atypical, indeterminate, and recall for atypical respectively.

Compared to ResNet, we have better precision on atypical and indeterminate. To further boost the performance, we also tried applying CLAHE and manually cropped the lung area. As we can see from

Tables 7 and 8, with cropped lung images as input, our model achieved 42% accuracy, with precision and recall of each class similar to previous results. However, with CLAHE, the accuracy is almost half of the original result. The precisions for each class are a little bit lower than others, but the recall for the atypical class is relatively high with 82%. The only drawback is that the recall for negative appearance is only 3%.

Method	Accuracy	Precision_ Typical	Precision_ Atypical	Precision_ Negative	Precision_ Indeterminate
Covid-Net1	46%	**83%**	10%	65%	21%
Covid-Net2	**42%**	75%	**11%**	**70%**	**22%**
Covid-Net3	**27%**	74%	8%	60%	18%

Table 7. Covid-Net1 - covid net with main structure frozen; Covid-Net2 - covid-net with cropped lung area as input; Covid-Net3 - covid net with CLAHE.

Method	Accuracy	Recall_ Typical	Recall_ Atypical	Recall_ Negative	Recall_ Indeterminate
Covid-Net1	**46%**	55%	72%	77%	43%
Covid-Net2	42%	**63%**	63%	**57%**	**67%**
Covid-Net3	27%	62%	**82%**	**3%**	36%

Table 8. Covid-Net1 - covid net with main structure frozen; Covid-Net2 - covid-net with cropped lung area as input; Covid-Net3 - covid net with CLAHE.

As the study is for clinical purposes, especially for inter-observer variability in patient care, we are not only concerned about the recall for covid classes, but also the recall for negative appearances. Because we need to avoid falsely diagnosing patients as covid-negative and expose them to the public. Thus the model with CLAHE preprocessing should be avoided, although it demonstrates 82% recall for atypical cases.

To further improve the approach, as mentioned before, the classifier is too small and incapable of handling such tasks, the simple and fast improvement method could be resolving the graphics memory issue and applying a better classifier. Another solution could be combining multiple datasets with required labels, however, there are no other datasets with the required labels at the moment.

Conclusion and future work

In this project, we have tried a large range of SOTA methods in the past few weeks. Specifically, the top-level strategy of our research is more like a tour of Breadth-First Search rather than Depth-First Search. Therefore, we have not achieved a quite pronounced performance so far and there is a long way to go to integrate our works. Nevertheless, we have got some interesting findings during the experiments and reflection.

For data enhancement, we are still struggling to improve the structure of DGM to reproduce the results displayed in the original paper. The paper is quite comprehensive in that it combines the distinctive design of components proposed by other SOTA papers in the field of GAN. If we get better residual or saliency maps later, we will concatenate them with the original images for enhancement of opacity areas. On the other hand, the limitation of computing resources is another obstacle. In the last few days, we tried to train the DGM model on the RSNA dataset used by the original paper with the same parameters provided by the authors. Unfortunately, the training process is not very smooth due to some memory issues and based on our estimation, if training on Colab Pro, the entire training will take more than 8 consecutive days. Nevertheless, we are very interested in the structure of DGM and will continue to try other ideas for improvement.

For single-label detection, the bottleneck happens when dealing with opacity boxes of smaller sizes. This is also a common problem for the other custom detection tasks. Small targets are difficult to detect, including low resolution, blurred images, and little information. As a result, the feature expression ability is weak. Specifically, when extracting features, there are very few features that can be extracted, which does not help the detection of small targets. The probable solutions mainly include integrating a targeted design of the network, for example, the feature pyramid networks, and applying data augmentation tricks. In addition, the difference in annotation habits might also be a misleading factor. As is depicted in Fig. 23, the conditions of the left lungs are quite similar but the annotations are totally different. Therefore, the inner issue of annotations is probably another bottleneck of this task.

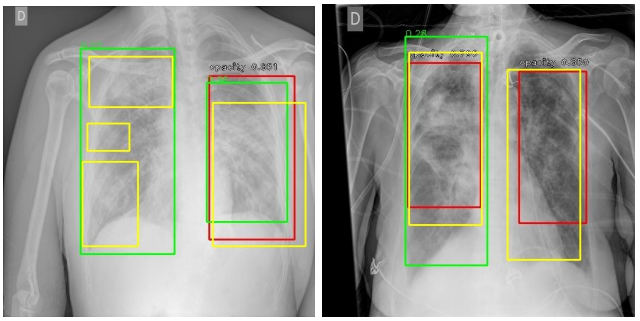


Fig.23 The detector tends to frame the whole left lung instead of the smaller parts

For classification, the bottleneck falls at the poor performance of the intermediate and atypical appearances. Though Covid-Net and few-shot perform well on the CXR dataset used in the original papers, it does not work well in this case. Our current experiments make it hard for a single common network structure to achieve competitive results. At the current stage, we plan to concatenate the features extracted by Covid-Net and few-shot, which acts as the input of the deeper layers. In this way, features get merged and the model is enhanced to learn more details of the images. Besides, it is necessary to dig into the dataset to figure out its unique features so that we can make more targeted improvements to the network structures.

Furthermore, for lack of medical and radiological background, it is still hard for us to get an intuition of opacity which remains a big obstacle for us. Therefore, we plan to read the documentation of the given dataset and try to integrate features of radiomics. In addition, we are very curious about the symbolic methods but we are not familiar with lung tissues and there is a long way to go to figure out the medical terminology. Despite the interdisciplinary difficulties, currently, there are some researchers that devote themselves to figuring out simple ways to explain the signs within the chest area [29]. Meanwhile, to apply symbolic representations, we must first find the tissues on the images,

which requires a huge quantity of annotations. Based on our experience, it might be hard to detect or segment the issues on CXR images as well due to the overlapping of tissues. Nevertheless, it is a good idea to try since it will produce semantic features that must be helpful to our models.

Finally, the most convenient way to improve the overall performance is to do ensembling. As is mentioned before, the models are complementary to each other. Therefore, we will try some ensembling strategies later.

References

[1] Tang, Y., Tang, Y., Zhu, Y., Xiao, J. and Summers, R.M., 2021. A disentangled generative model for disease decomposition in chest x-rays via normal image synthesis. Medical Image Analysis, 67, p.101839.

[2] Jadon, S., 2021, February. COVID-19 detection from scarce chest x-ray image data using few-shot deep learning approach. In Medical Imaging 2021: Imaging Informatics for Healthcare, Research, and Applications (Vol. 11601, p. 116010X). International Society for Optics and Photonics.

[3] Yasaka, K. and Abe, O., 2018. Deep learning and artificial intelligence in radiology: Current applications and future directions. *PLoS medicine*, *15*(11), p.e1002707.

[4] Ioffe, S. and Szegedy, C., 2015, June. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning* (pp. 448-456). PMLR.

[5] Ulyanov, D., Vedaldi, A. and Lempitsky, V., 2016. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*.

[6] Ba, J.L., Kiros, J.R. and Hinton, G.E., 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

[7] Wang, T.C., Liu, M.Y., Zhu, J.Y., Tao, A., Kautz, J. and Catanzaro, B., 2018. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 8798-8807).

[8] Huang, X. and Belongie, S., 2017. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 1501-1510).

[9] Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P. and Zitnick, C.L., 2014, September. Microsoft coco: Common objects in context. In *European conference on computer vision* (pp. 740-755). Springer, Cham.

[10] Deng, J., Dong, W., Socher, R., Li, L.J., Li, K. and Fei-Fei, L., 2009, June. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition* (pp. 248-255). Ieee.

[11] Iandola, F., Moskewicz, M., Karayev, S., Girshick, R., Darrell, T. and Keutzer, K., 2014. Densenet: Implementing efficient convnet descriptor pyramids. *arXiv preprint arXiv:1404.1869*.

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

[13] Savadjiev, P., Chong, J., Dohan, A., Vakalopoulou, M., Reinhold, C., Paragios, N. and Gallix, B., 2019. Demystification of AI-driven medical image interpretation: past, present and future. *European radiology*, *29*(3), pp.1616-1624.

[14] Pisano, E.D., Zong, S., Hemminger, B.M., DeLuca, M., Johnston, R.E., Muller, K., Braeuning, M.P. and Pizer, S.M., 1998. Contrast limited adaptive histogram equalization image processing to improve the detection of simulated spiculations in dense mammograms. *Journal of Digital imaging*, *11*(4), p.193.

[15] Saiz, F.A. and Barandiaran, I., 2020. COVID-19 detection in chest X-ray images using a deep learning approach. *International Journal of Interactive Multimedia and Artificial Intelligence, InPress (InPress)*, *1*.

[16] Wang, L., Lin, Z.Q. and Wong, A., 2020. Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images. *Scientific Reports*, *10*(1), pp.1-12.

[17] Park, S., Kim, G., Oh, Y., Seo, J.B., Lee, S.M., Kim, J.H., Moon, S., Lim, J.K. and Ye, J.C., 2021. Vision Transformer for COVID-19 CXR Diagnosis using Chest X-ray Feature Corpus. *arXiv preprint arXiv:2103.07055*.

[18] Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D. and Batra, D., 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision* (pp. 618-626).

[19] Ye, W., Yao, J., Xue, H. and Li, Y., 2020. Weakly supervised lesion localization with probabilistic-cam pooling. *arXiv preprint arXiv:2005.14480*.

[20] Ren, S., He, K., Girshick, R. and Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, *28*, pp.91-99.

[21] Cohen, J.P., Morrison, P. and Dao, L., 2020. Covid-19 image data collection. arxiv 2003.11597, 2020. URL https://github. com/ieee8023/covid-chestxray-dataset.

[22] Wang X, Peng Y, Lu L, Lu Z, Bagheri M, Summers RM. ChestX-ray8: Hospital-scale Chest X-ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases.

[23] Lin, Z.Q., Shafiee, M.J., Bochkarev, S., Jules, M.S., Wang, X.Y. and Wong, A., 2019. Do explanations reflect decisions? A machine-centric strategy to quantify the performance of explainability algorithms. arXiv preprint arXiv:1910.07387.

[24] Vayá, M.D.L.I., Saborit, J.M., Montell, J.A., Pertusa, A., Bustos, A., Cazorla, M., Galant, J., Barber, X., Orozco-Beltrán, D., García-García, F. and Caparrós, M., 2020. Bimcv covid-19+: a large annotated dataset of rx and ct images from covid-19 patients. arXiv preprint arXiv:2006.01174.

[25] Tsai, E.B., Simpson, S., Lungren, M.P., Hershman, M., Roshkovan, L., Colak, E., Erickson, B.J., Shih, G., Stein, A., Kalpathy-Cramer, J. and Shen, J., 2021. The RSNA International COVID-19 Open Radiology Database (RICORD). *Radiology*, *299*(1), pp.E204-E213.

[26] Chung, A.G., 2020. GitHub-agchung/Figure1-COVID-chestxray-dataset: Figure 1 COVID-19 Chest X-ray Dataset Initiative.

[27] Bochkovskiy A, Wang C Y, Liao H Y M. Yolov4: Optimal speed and accuracy of object detection[J]. arXiv preprint arXiv:2004.10934, 2020.

[28] Lin T Y, Goyal P, Girshick R, et al. Focal loss for dense object detection[C]//Proceedings of the IEEE international conference on computer vision. 2017: 2980-2988.

[29] Chiarenza A, Ultimo L E, Falsaperla D, et al. Chest imaging using signs, symbols, and naturalistic images: a practical guide for radiologists and non-radiologists[J]. Insights into imaging, 2019, 10(1): 1-20.

[30] Demir, U. and Unal, G., 2018. Patch-based image inpainting with generative adversarial networks. *arXiv preprint arXiv:1803.07422*.

Appendix

Contribution of Each Member

Report Contribution	
Introduction & Related Work	Lihuan Li & Yang Ma & Zhihan Qin
Methods - Data enhancement	Lihuan Li
Methods - Detection	Zhihan Qin
Methods - Classification - Two-Stage Few-shot	Lihuan Li
Methods - Classification - End-to-end Covid-Net	Yang Ma
Experimental Setup	Yang Ma
Results&Discussion - Data enhancement	Lihuan Li & Zhihan Qin
Results&Discussion - Detection	Zhihan Qin
Results&Discussion - Two-Stage Few-shot	Lihuan Li
Results&Discussion - End-to-End Covid-Net	Yang Ma
Conclusion and future work	Zhihan Qin

Table 9. contribution of each member on the report

Project Contribution	
Data enhancement **- DGM**	**Lihuan Li & Zhihan Qin** We (Lihuan and Zhihan) worked on this model together. Since there is no official code, we spent our major time reproducing the results because we believe this is a very novel method for improving the interpretability and opacity detection on medical imaging. We spent a lot of time studying a variety of SOTA structures of GAN as this is the very first time for both of us to build a GAN model on our own. We should have started with the basic GAN structures which would save us a lot of time to avoid some primary problems. Also, we experienced the memory leak, countless unexpected shutdowns of colab, the version confusion of TensorFlow, etc. However, these problems inspired us to conquer the task. This is the model that has stretched us the most and we are also willing to continue working on it in the future.
Detection	**Zhihan Qin** For the detection part, I trained several models and planned to hand over the models to our quitted member for ensembling. During the process, I made comparisons of groups of experimental results and gradually figured out the bottlenecks of this specific task. As time is limited, I did not do too much tuning and improved the separate models since I mainly focused on DGM which nearly ran out of my time. Later I'll keep doing some experiments as I think I have found my interest to play with these medical images and I still have a lot of ideas that I haven't had time to realize. Finally, I really appreciate having the chance to cooperate with my teammates. Both Lihuan and Yang inspired me with a lot of innovative ideas and they have shared many personal experiences with me. For me, this is the best group work experience up till now.
Classification **- Two-Stage Few-shot**	**Lihuan Li** For this method, I have experimented with metrics-based few-shot learning with machine learning clustering/classification methods. For few-shot learning, I chose the Siamese Networks which are based on Contrastive Learning as I considered it would be suitable for the size of our experimented dataset. However, this method turned out to be not suitable for this dataset. In the project demo, I concluded that the machine learning methods were not able to learn feature embeddings generated by the Siamese Network and it was not correct. Later, Zhihan assisted me in having some analysis on the results of multiple feature dimension reduction approaches to further prove that there were some issues with the previous experiment procedure and the final conclusion. And in this report, I have corrected the conclusion in our presentation.
Classification **- End to End Covid-Net**	**Yang Ma** For Covid-Net, I have conducted image preprocessing, model fine-tuning with a set of hyperparameters, and conducted analysis on different results. I tried to resolve graphics memory issues but wasn't successful. And I have also utilized the ResNet backbone as a baseline for better comparisons. Covid-Net is a deep and complicated model where I have learned a lot from. And the results of such a deep learning based approach have also implied some difficulties in classifying those diagnoses by computer vision methods. Appreciate the support from my teammates and the teaching group. With their help, I learned more than I had expected and saved a lot of time.

Table 10. the contribution of each member on the project