Detecting Covid from Children Lungs

Yue, Yu · Min Hung, Shih · Devika, Subramanian

April 22, 2022

Abstract

Covid, five times more deadly than the flu, has been a devastating event to all of us. However, like other types of pneumonia, pulmonary infection with COVID-19 results in inflammation and fluid in the lungs. X-ray images are an easily accessible and fast method of diagnosing COVID-19. The main objective of this project is to utilize machine learning models along with image processing techniques to provide a quick and confident diagnosis based on X-ray images. Highlights of this study includes: An image prepossessing pipeline that crops the lung region from the original images and a novel criterion for detecting outliers.

1. Introduction

Currently, there are many methods that intend to tackle the detection of COVID through X-Ray images or CT images, and many of them actually achieves high accuracy and are able to be aid in the diagonosis processes. Nonetheless, these models are mostly trained on adult lungs and haven't been well generalized for children lungs. On top of that, what features do the models actually pick up from the image data? These questions, despite being straightforward, are actually non-trivial and we will try to answer some of them in this paper.

Some preliminary assumptions and measures to facilitate the training process before we start include:

- To avoid the computation cost of training a neural net from scratch, we will focus on transfer learning, utilizing pretrained models available online.
- To mitigate the potential discrepancy between the input structure between adult lungs (pre-trained dataset) and children's lungs (given dataset), fine tuning the entire network is preferred over only training the classifier layer.
- To facilitate the training process, original images are first resized to 224 x 224 or 512 x 512 grayscale images before being uploaded to Google Colab for training.

2. EDA

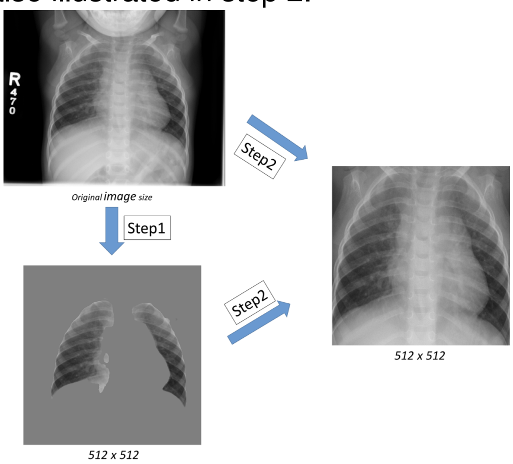
We first explored the original dataset and found some potential issues that may affect training:

- X-rays were taken from different angles, e.g. front, side or behind
- Various kinds of postures.
- Abnormal cropping or rotation of the images
- Various exposure levels

- There are annotations on many of the images.
- In some severe cases, the patients have life supporting equipments such as tubings and pacemakers.

2.1. A pipeline to crop lung regions

To mitigate the effects of the first and second bullet points and detect outliers, we developed a data pre-processing pipeline (shown below) that first utilizes a pre-trained U-net model[ref6] for lung segmentation (step 1) then based on the utmost border of segmentation, we crop the original images to obtain images that only contain lung regions, which is also illustrated in step 2.

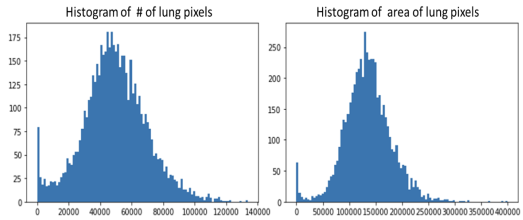


As is shown in the above image, information such as annotations and regions that are deemed irrelative to the chest, such as legs and arms, are excluded from the original images, which we believe can help mitigate off targeting issues during the training phase.

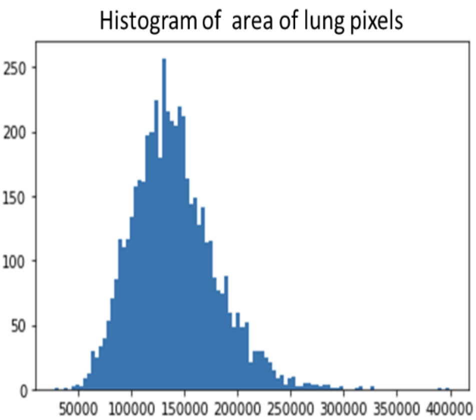
2.2. A pipeline to detect outliers

Since U-net performs lung segmentation based on the model's criterion to determine whether pixels in the image actually belong to the lungs, we can use this function to perform outlier identification. We came up with the following pipeline to identify the outliers in the images:

- Perform lung segmentation using U-net as discussed above.
- For each image, calculate the number of pixels and size of area that are considered "lungs" by the model.
- Classify images that contain too few lung pixels (less than 20,000) as outliers.



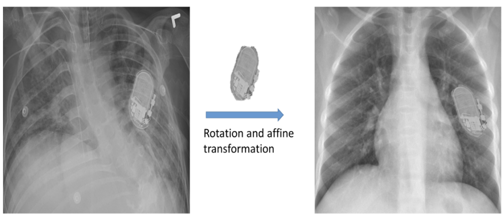
In the above figure, we plotted a histogram of number and area of lung pixels for the training dataset. As demonstrated, based on the U-net criterion, many training images don't even have enough pixels to represent a complete lung. We double checked these images and as expected, their input distribution greatly differs with the rest. We settled upon a cutting threshold of 20,000 pixels and below this threshold we classify these images as outliers. Below is the resulting histogram of area of lung pixels that outliers have been removed.



The resulting histogram of lung area is very close to a Gaussian distribution. The reason we calculated both the number and area of lung pixels is because it is possible that the area is large but the number of pixels is relatively small, vice versa. So a potentially better cutting threshold should be selected based on both criterions.

2.3. A method to ignore wrong targets within lung region

Now we have established one pipeline to exclude potential off targets (targets out of lung regions) such as letter labels and another pipeline to identify outliers. The remaining issue is, what if the wrong targets (such as pacemakers) are within the lung regions? There might be a correlation between having pacemakers in the images and the child has contracted COVID, this is because children with severe comorbidity usually have compromised immune systems who are prone to contracting COVID. However, a doctor may not directly diagnose the patient with COVID based on indirect features, such as whether a patient has a pacemaker or not. Instead, he will integrate clinical tests, EHR history as well as x-ray results for his or her decision-making. One of our objectives for this study is to detect COVID from lung x-rays so that indirect information such as pacemaker should be excluded from feature space of the model. In an attempt to train our model to become invariant of pacemakers, we will randomly superimpose 'fake pacemakers' to the lung regions of the images during data augmentation as shown in the below figure. This part of work is still under development and we could potentially use GANs to simulate more realistic pacemakers.



3. Data Augmentation & Overfitting

To train a neural network with millions of parameters from scratch we need lots of data, but we only have 5655 training data. A good option is to do transfer learning on an existing model that has been trained on chest x-ray images. However, over-fitting is still a significant or the most critical issue to deal with during the training phase. Here are the countermeasures we did for controlling over-fitting:

- Set the learning rate very low (<0.0001), or set a scheduler for LR decay.
- Use AdamW as the optimizer with a L2 norm regularization on weights.
- Use functions provided in torch.transforms for image augmentations.
- Freeze some of the parameters from being updated.

So far, we found the learning rate of 10^{-4} to be optimal because it gives a smoother training/validation curves, which shows slow over-fitting and has a relatively short training time. While it is possible a higher learning rate such as 10^{-3} combined with learning rate decay may be a good option. The higher learning rate may increase the stochasticity during the training phase and is also prone to being stuck at the local minimum. This implies multiple restarts, which requires more training time. L2 norm and image augmentation are powerful methods for controlling over-fitting which also implies that if we tune both values high enough then the model will under-fit. In short, a balance of both worlds is the key. More specifically, we set:

- weight decay of AdamW = 0.3
- Transformations:
  - RandomRotation(10)
  - RandomAffine(deg=10, translate=(0.1, 0.1)),
  - RandomHorizontalFlip(p=0.3)
  - ColorJitter(brightness=[0.5, 2])
  - ToTensor()
  - Normalize([0.5], [0.00048828125])

Rotation or affine translation with large angles, e.g. > 20 degrees are avoided, because large rotations require extra numbers of padding on the corners and edges of the image, which may become wrong targets of the model during the training phase. We also added a random brightness adjustment in order to train the model to become adaptive to various exposure levels. To cope with over-fitting, we also tried limiting the trainable layers of the network. After tinkering with several conditions, we obtain the best results from fine tuning the whole layers which is probably due to the significant difference in input distributions between children lungs dataset and those used for the pre-trained model(adult lungs).

4. Model

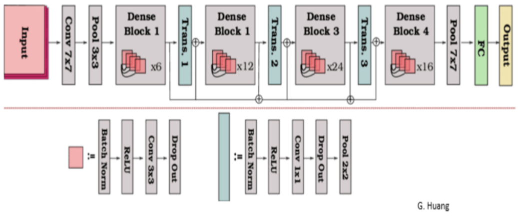
For the most part of our training, we settled upon using the pre-trained models from the torchxrayvision project hosted on Github[Ref4]. We first compared the models based on model's parameters: a Res-50 model has 23M parameters, whereas a dense-121 model has around 7M parameters.

Table 1: Parameters of models

Model	parameters
Res-18	11689512
Res-50	25557032
Dense-121	6948609

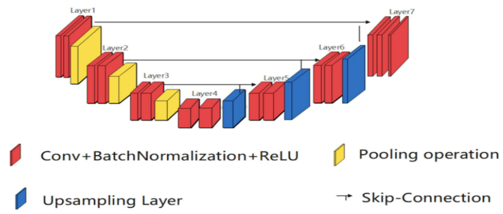
4.1. Model I: A simple transfer learning scheme

In Model I, we directly sent 224 x 224 training images as input to Densenet-121 or 512 x 512 to Resnet-50 for transfer learning. The Densenet-121 architecture is as illustrated in [Ref5], which is constituted of 4 blocks of convolution layers. Most of the time we used this model for training and exploring hyper-parameters because training is fast and we can easily freeze blocks of layers to control over-fitting. Resnet-50 has more parameters which takes longer to train but gives slightly better prediction performance.



4.2. Model II: Lung cropping + transfer learning

In the second phase of our study, we also adopted U-Net[ref6] as a method to segment out the lung area, and the architecture is shown as below. The cropping method for generating lung region specific training dataset has been described in section 2.2.Then the cropped images were used as new input to the models for transferlearning the process of which is similar to Model I.



5. Results

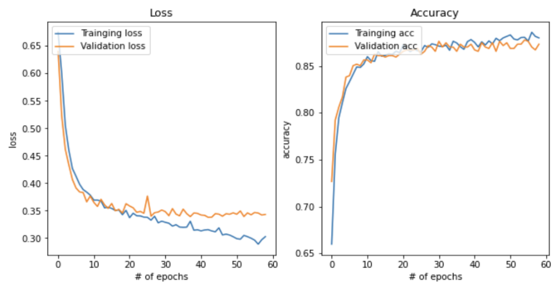
5.1. Best performing model

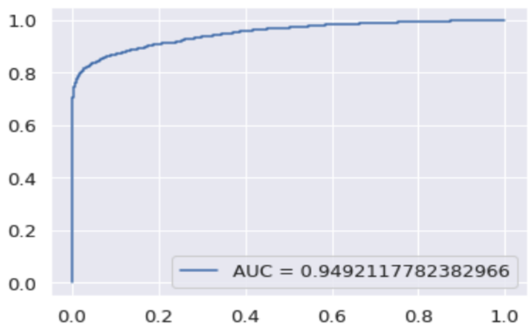
The best performing model in terms of prediction accuracy turns out to be Res-50 based on Model I. We think one of the reasons why Model I outperforms Model II is because Model I still contains indirect features in the training data that may slightly boost accuracy performance. But as will be discussed later, this boost in accuracy is in the expenses of off-targetting of the model. Nevertheless, the overall performance is good and parameters used for training are listed below:

- Input size = 512 x 512

- Fine tuning all the layers
- Learning rate = 0.0001
- optimizer = AdamW (weight decay = 0.3)
- Batch size = 16
- Number of epochs = 54
- Augmentation as mentioned above

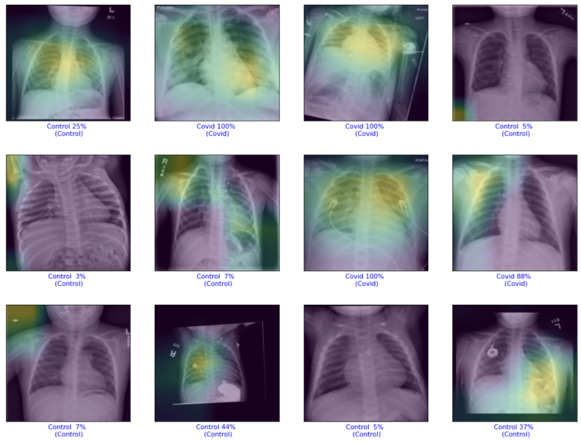
In addition, we include some of the metrics used for evaluating the model, including the loss, accuracy and the AUC score:



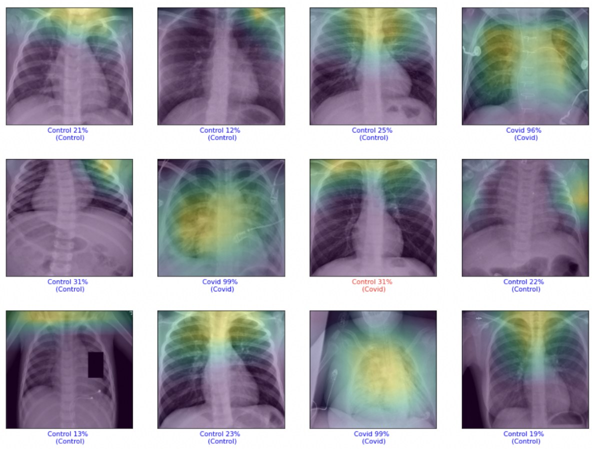


5.2. GradCam

In addition to accuracy, in order to evaluate the correctness of our trained model, we used the tool GradCam, which allows us to visualize the features extracted by the model used for predictions. Below are samples of the results of Model I:

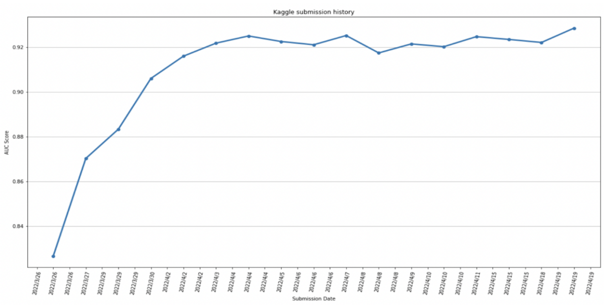


From the samples we can see, most of the time Model I tracks the targets within the lung regions. However, there are cases that the model tracks the wrong targets such as annotations, shoulders and corners of the image. This motivates us to find a better way to exclude these off targets. Eventually, we adopted a pre-trained U-net model[ref6] and built a pipeline for image preprocessing that automatically crops the lung regions from the original images and resized them to 224 x 224. These cropped images will become the input to a Dense-121 model, which we referred to as Model II. Again, samples of the GradCam images using Model II are shown below:



Benefiting from the cropping, targets of Model II show fewer off-targets compared to Model II, but often the model will still target on ribs and the spine region. Additionally, the AUC score of Model II has dropped from 0.94 to 0.92. This drop of performance is likely due to an increased discrepancy of input distribution between the dataset that was used to train the pre-trained model and the dataset of our cropped children lung images. Ideally, the better approach will be training a Dense-121 from scratch using a large adult x-ray dataset but with images cropped according to the method described above then do transfer learning using our cropped children lung dataset.

5.3. Kaggle submission history



We also included a Kaggle performance history (a sample of total of 56 submissions) in the above figure. In the beginning phase, the rising performance is due to our model selection, finding the optimal learning rate, tuning regularization and augmentation strength. In the middle plateau, the issue of off-targeting of Model I is recognized and instead of doing a grid search of hyperparameters to boost accuracy, we focused more on refining our approach of dealing with off-targeting, which motivated our development of U-Net cropping pipeline in Model II.

The last uptick in our performance is achieved by increasing our training epochs on the best performing weights with a very small learning rate and small augmentation.

6. Conclusions

- When dealing with relative large image size (1-2Mb for covid lung case), we should first resize or crop the images to a smaller size (224x224 - 20kb, 512x512 - 100kb) then upload to the cloud such as GCP for GPU training. This will significantly reduce the image loading time and accelerate the training speed to 30s/epoch compared to 2 mins/epoch.

- EDA and data pre-processing are critical as most of the efforts were spent on determining outliers and locating lung regions.

- Given enough training data, a neural network with more parameters may yield better training accuracy, though it will also cost more time due to computations for backpropagation.

- Controlling over-fitting is critical. A careful choice and balance of image augmentations and L2 regularization strength is needed.

- In this study Model I outperforms Model II in accuracy but Model II are better at dealing with off-targeting of the model.

- The outliers identified (based on less than 20,000 lung pixels criterion) are still valuable data but are much more difficult to train due to limited number and variability of images. However, we can develop better U-net models to achieve better lung segmentation even under conditions that the images are non-typical.

- Images with size of 512x512 yield better results compared to 224x224 likely because a higher resolution of image means it contains more pathological details for detecting COVID. Resolution of the image may not be very important if the job is to discriminate between a dog and a cat. However, subtle information underlying COVID pathology might be lost when downsampling the original image to a much lower resolution. One method that hasn't been tested is to cut the original images into smaller pieces for training while maintaining the original resolution.

References

1. jacobgil/pytorch-grad-cam: Many Class Activation Map methods implemented in Pytorch for CNNs and Vision Transformers. Examples for classification, object detection, segmentation, embedding networks and more. Including Grad-CAM, Grad-CAM++, Score-CAM, Ablation-CAM and XGrad-CAM (github.com)

2. Pneumonia Classification Using Pytorch | Kaggle

3. Application of Machine Learning in Diagnosis of COVID-19 Through X-Ray and CT Images: A Scoping Review - Hossein Mohammad-Rahimi1, Mohadeseh Nadimi2,3, Azadeh Ghalyanchi-Langeroudi2,3, Mohammad Taheri4* and Soudeh Ghafouri-Fard5*

4. TorchXRayVision: A library of chest X-ray datasets and models - arXiv:2111.00595 [eess.IV]

5. Densely Connected Convolutional Networks - Gao Huang, Zhuang Liu, Laurens van der Maaten, Kilian Q. Weinberger

6. U-net for lung segmentation: https://github.com/IlliaOvcharenko/lung-segmentation

7. A Quick Overview to the Transfer Learning and it's Significance in Real World Applications - Kanish shah

8. "Densely Connected Convolutional Networks," 2018 - G. Huang, Z. Liu and L. van der Maaten

9. A Novel Elastomeric UNet for Medical Image Segmentation - Sijing Cai1,2, Yi Wu1,3* and Guannan Chen1,3*