Deep Learning Small Project Report

Licheng Xu, Shijie Luo

March 2021

1 Introduction

COVID-19 has been a very serious issue globally for the past year, and it is still an important incident that concerns people these days. Due to the fact that the syndromes of COVID-19 infection mainly appear as a certain kind of lung damage, a question came to our mind: Is it possible to distinguish COVID infected patients with normal people and other non-COVID infected patients, by taking a look at their chest X-ray images?
Given a labelled data set consisting of chest X-ray images from different patients: normal, non-COVID infected and COVID infected. We decide to train a model to help us distinguish these three kind of patients. Due to these syndromes being high-level semantic information, we decide to adopt some deep neural networks, starting from ResNet-50 as our baseline, followed by ResNet-101 and ResNet-152[1] to see if we can achieve better performance.

2 Dataset

The dataset consists of three classes of images, including "Normal", "Infected-COVID", and "Infected-Non-COVID". The data distribution is shown in Figure 1.
In order to use this dataset, we create a custom loader for our model. We first transformed the dataset to pytorch tensor and then divided by 255 to normalized the dataset to $[0, 1]$ so that the model can converge faster. The distribution of the training set is biased, with non-COVID class having the most number of images, which is twice as many as other classes. Different from the training set, the test dataset and the validation dataset have almost identical number of data for each class.
However, in the actual testing, when we make the prediction on the validation dataset, all of the validation examples are classified as the "Infected-Non-COVID" (See figure 14), our model do suffer from the problem of biased training dataset.

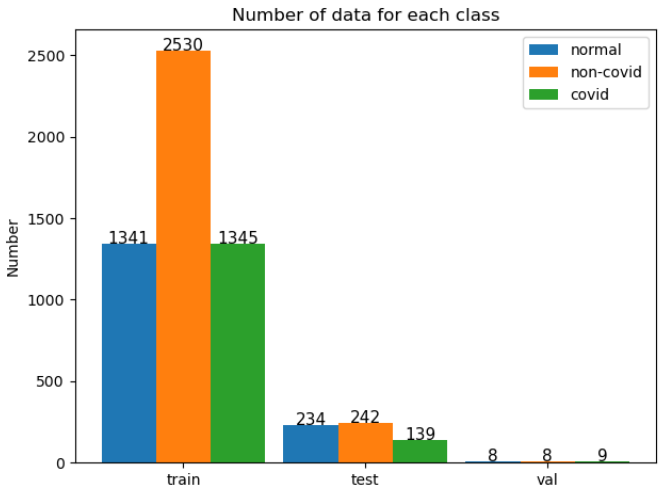

Figure 1: The dataset distribution

3 Methodology

In order to train the model, we followed ResNet-50, ResNet-101 and ResNet-152 [1] respectively, due to the fact that medical syndromes are usually high-level semantic features. However, there is one thing to settle: should we use one single three-class classifier, or should we use double binary classifiers?
In order to tackle this question, we analyze the advantage and disadvantage of these two approaches. We also conduct both experiments for comparison.
In order to carry out better training and testing, we decide to adopt some data augmentation techniques as well. The first problem we wanted to tackle was the biased distribution of data. However, the biased data distribution ratio (normal vs the rest in this case) did not exceed the empirical ratio 2:8. Thus we did not perform dataset re-sampling on the training set.
Since the X-ray images may have different brightness, and contrast, the model that can perform well on images with different brightness and contrast may also have better performance on the unseen new patients X-ray images. We decided to perform data augmentation in this way.
On the training set and the test set, we used the normalized training examples directly.

3.1 Data Augmentation

Due to the nature of the data set being a bio-medical dataset, it is not reasonable to adopt any random data augmentation techniques because the results produced by those techniques might not make bio-medical sense. For example, horizontal flip might not make any sense due to the existence of situs inversus

people. It also doesn't make sense to do image cropping because doctors do not want to miss any potentially important information from chest x-ray images. However, it is possible that different x-ray machines might produce result images of different images according to different post-processing procedures, and doctors might see the chest x-ray images under different brightness as well. Therefore, we adopted two data augmentation techniques:

1. Modifying Contrast
2. Modifying Brightness

3.2 Three-Classes Classifier

For three-classes classifier, it is an End-to-End classifier which directly outputs the logit for three different classes: Normal, COVID, and Non-COVID. It is convenient to train such a model because we do not have to design a hierarchical or parallel output structure for multiple binary classifiers. In this case, we use a softmax classifier.
However, such a model usually doesn't perform well when both coarse-grained classification and fine-grained classification are required at the same time. Empirically coarse-grained classifiers would be more sensitive to large differences, which means those classifiers might have larger weights; while at the meantime fine-grained classifiers are more sensitive to small differences, which means fine-grained classifiers might have smaller weights. It could confuse the model when both filters, with larger weights and smaller weights, are required to appear at the same time.

3.3 Double Binary Classifiers

For double binary classifiers, we train two different models to discriminate between the pair "Normal-Infected" and "COVID-Non-COVID" respectively. This requires extra computation because these two models do not share parameters, and we need to train them respectively. However, due to the fact that when training two models, we are tackling the task for coarse-grained classification and fine-grained classification separately, the model should not face the problem of considering fine-grained information as noise and neglecting them, nor the problem of considering coarse-grained information as outliers because fine-grained information and coarse-grained information do not appear at the same time, in this case. Intuitively, this classifier design should outperform the first one, given the assumption that this task is feasible.

3.4 ResNet Structure

The difference between the COVID and non-COVID images have very few difference. As a result, we expect that a deeper model might be capable to find out the deep features from these images. However, simply stacking numbers of layers may lead to overfitting or gradient vanishing problem, resulting in

degradation, which makes the training of the very deep neural network hard to converge.
As the result, we adopt the design of ResNet model [1]. The main part of the design of this model is the use of bottleneck layers. As shown in the image below, for each bottleneck layers, it consists two different path, one of the path consists a stack of 3 convolutional layers with kernel size (1×1) followed by another convolutional layer with kernel size (3×3), the last layer is a convolutional layer with kernel size (1×1) with. On another path, we simply use a convolutional layer with kernel size (3×3) if the input size is not equal to the output size. After that, we simply add the results from the two different paths together.

The usage of the identity mapping shown above helps adding a short cut between the input and the output. If the output from the deeper path have lower performance, it will simply be shortcut by the (3×3) path. Which prevents the problem caused by stacking too many layers. Besides, the use of the convolutional layer with kernel size (1×1), on the one hand, it reduces the number of parameters required, on the other hand, stacking of more layers introduces more non-linearity to the model.

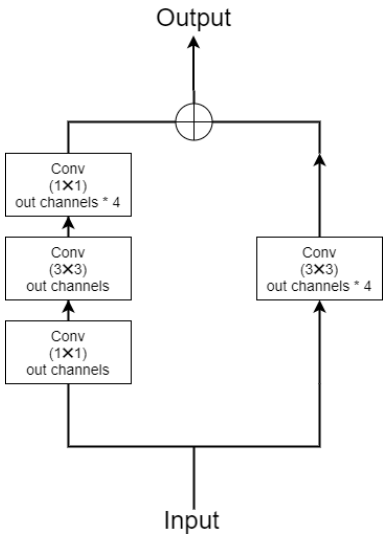

Figure 2: The bottleneck structure

Our ResNet [1] model is consisting of 4 different stages, we only do down sampling at the first layer bottleneck of the stage to reduce half of the feature map. In each stage, it contains several layers of bottleneck blocks. The number of the bottleneck blocks in each stage is shown in the table above.

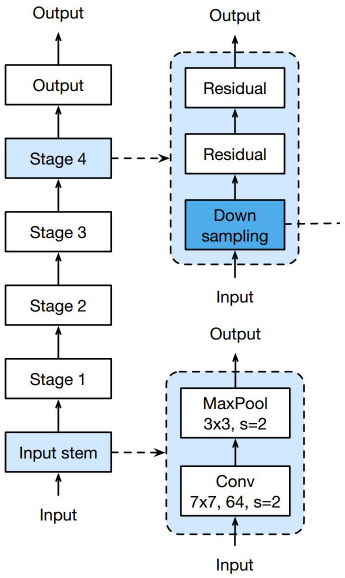

Figure 3: The resnet structure

Stages/model	Resnet 50	Resnet 101	Resnet 152
stage 1	3	3	3
stage 2	4	4	8
stage 3	6	23	36
stage 4	3	3	3

Table 1: Number of bottlenecks in each stage of ResNet model

4 Experiments

In our experiment settings, we tried 3 different models as our backbones: ResNet-50, ResNet-101 and ResNet-152 [1], each of them with a sigmoid classifier for binary classification, or with a softmax classifier for multi-class classification.
We use Adam as the optimizer for our model, with the default learning rate 0.001. Since this learning rate work well on most of the models, we expect that it will also perform well on our models. Since Adam optimizer update every dimension of the updates separately and it uses the true weighted average to calculate the momentum term, we expect it will reduce the step-size fast when we enter a very steep region and take larger step at a flat region. In such way that it can better optimizing our model.
We initialize the weight of the model using the default initialization weight algorithms directly and we apply weight decay to the model parameters with decay rate 0.001 as the regularization term. By restricting the parameters from getting too big, it can prevent the model from overfitting and make the model generalize well to some extent. The parameter 0.001 is commonly used by other people, and it is not larger than the learning rate, so we hope that it can help to make our model generalize well.
See table 2 for details.

4.1 Three-classes Classifier

For the three-classes classifier, we use a single Resnet model [1] with 3 output unit and softmax as the activation function to make predictions. All the training data is send to the model directly as three different classes. The cross entropy loss is used as the criterion for the model.

4.2 Double Binary Classifiers

In this approach, we trained two different Resnet models [1] each with one output unit with sigmoid as the activation function. One of the model is used to make prediction if the patient is a infected or normal person, while another one is used to make prediction on the infected images to further classify whether they are COVID infected or non-COVID infected. In this case the binary cross entropy loss is used as the loss function.

5 Findings

Firstly we conducted experiments with backbone ResNet-50. [1] We achieved a fair result by using double binary classifiers. Therefore, we assume that the model can be further improved by stacking more layers, extracting higher-level semantic information, especially those that indicate the key difference between COVID infected patients and Non-COVID infected patients. In this sense, we further experiment with backbone ResNet-101 and ResNet-152 [1].

Model	Loss	Accuracy	Precision	Recall	F1-Score
ResNet-50					
Three-Classes	7.780	76.7%	0.709	0.597	0.648
Normal-Infected	7.723	83.9%	0.807	0.974	0.883
COVID-Non-COVID	8.461	81.6%	0780	0.691	0.733
ResNet-101					
Three-Classes	7.893	76.4%	0.694	0.619	0.654
Normal-Infected	7.723	83.9%	0.807	0.974	0.882
COVID-Non-COVID	9.857	79.8%	0.742	0.683	0.712
ResNet-152					
Three-Classes	8.584	68.1%	0.497	0. 698	0.548
Normal-Infected	9.489	84.1%	0.804	0.982	0.884
COVID-Non-COVID	10.604	79.5%	0.729	0.612	0.713

Table 2: Grading Metrics for Different Backbones and Classifiers

However contrary to what we have assumed, the performance of the binary classifier distinguishing COVID-infected patients and Non-COVID-infected patients didn't change much. See Figure 4-9 for details.

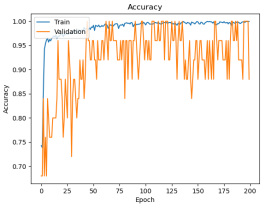

Figure 4: Accuracy for ResNet-50 of Infected-Normal Pair

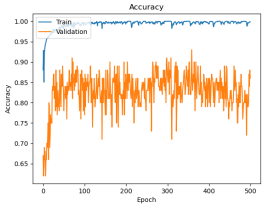

Figure 5: Accuracy for ResNet-101 of Infected-Normal Pair

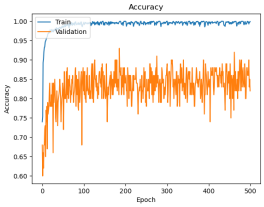

Figure 6: Accuracy for ResNet-152 of Infected-Normal Pair

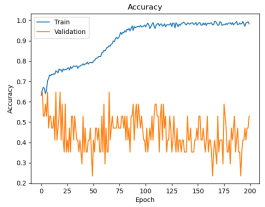

Figure 7: Accuracy for ResNet-50 of COVID-Non-COVID Pair

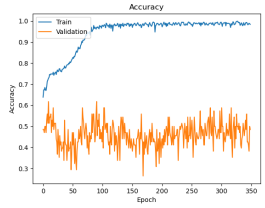

Figure 8: Accuracy for ResNet-101 of COVID-Non-COVID Pair

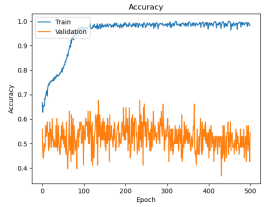

Figure 9: Accuracy for ResNet-152 of COVID-Non-COVID Pair

As we can see, for Infected-Normal pair classification, the accuracy converges at around 85% for all three backbones, ResNet-50, ResNet-101 and ResNet-152 [1]. This doesn't surprise us, because distinguishing infected and normal patients doesn't require a very high level of semantic information, thus simply stacking more layers might not be of much help. However, it is interesting that the model doesn't overfit when stacking 152 CNN layers. This might be due to the design of shortcut connections.
However, it is interesting that even with ResNet-152 [1], the performance for distinguishing COVID from Non-COVID patients is still quite poor. ResNet-50's accuracy fluctuated at 45%, while ResNet-101's accuracy fluctuated at around 45% as well, and ResNet-152's accuracy fluctuated at 50%. All these performances are no better than merely guessing, which has attracted our attention.
Due to the fact that recall is more appreciated while distinguishing patients than accuracy because we do not want to miss any of the potential patients in order to prevent COVID from further spreading, we also referred to the recall for our models, and we believe that this statistic will be more important compared to the accuracy, especially when our model actually suffered from the biased data distribution. As indicated from table 2, the recall for all three backbones for the COVID distinguishing task are around 65%, which we can merely say it is good. Empirically ResNet-152 [1] should be able to catch a fair amount of fine-grained features, but it didn't perform well on this task. Thus we highly doubt if COVID is distinguishable using just chest X-rays.
According to [2], a common syndrome that can be found from chest x-rays was peripheral ground glass opacities (GGO) affecting patients' lungs' lower lobes. However, this syndrome is not exclusive from COVID infection, nor is the only syndrome of COVID infection. Other kinds infection can also produce such syndrome, attribute to a probability. Up-to-now, we are yet to find any COVID-exclusive syndromes from lungs. In this case, It would be very hard, even for professional medics, to tell merely from a chest X-ray image if one is infected by COVID. This indicates that distinguishing COVID from Non-COVID infections might not be feasible with only chest X-ray images.
We also compare the models' performance with the multi-class classifiers (See figure 10-12 for accuracy details) and find that double binary classifiers do outperform single three-classes classifier, which corresponds to our expectation. In order to further explain this phenomenon, we printed the means, variances and standard deviations for the final fully connected layers of selected models. See table 3 for details. Here are several things we can see from the table:

1. All the models share a similar order of magnitude for their variances and standard deviations.

2. Binary classifiers share similar number for their variances and standard deviations, around 4.5 and 6.9 respectively.

3. Three-classes Classifiers share similar number for their variances and standard deviations, around 2.3.

Model	Mean	Variance	Standard Deviation
ResNet-50			
Three-Classes	-2.86×10^{-5}	2.50×10^{-3}	4.97×10^{-2}
Normal-Infected	-1.30×10^{-3}	4.80×10^{-3}	6.93×10^{-2}
COVID-Non-COVID	2.00×10^{-4}	5.10×10^{-3}	7.17×10^{-2}
ResNet-101			
Three-Classes	-1.28×10^{-6}	2.50×10^{-3}	5.00×10^{-2}
Normal-Infected	1.00×10^{-4}	4.30×10^{-3}	6.52×10^{-2}
COVID-Non-COVID	-1.00×10^{-3}	4.80×10^{-3}	6.96×10^{-2}
ResNet-152			
Three-Classes	4.27×10^{-5}	2.20×10^{-3}	4.73×10^{-2}
Normal-Infected	-8.00×10^{-4}	4.80×10^{-3}	6.95×10^{-2}
COVID-Non-COVID	5.00×10^{-4}	4.60×10^{-3}	6.82×10^{-2}

Table 3: Statistics of different models' fully connected layers

4. The means of three-classes classifiers always lie in between the means of the binary classifiers.

In this case, it is obvious that these classifiers face a domain shift problem. The domain of Infected-Normal classification and COVID-Non-COVID classification share fair similarities, but they have more differences. Therefore it isn't surprising that the three-classes classifier would struggle with these two domains and its performance would fluctuate because it is still very hard for machines to learn significantly different decision boundaries all by itself.

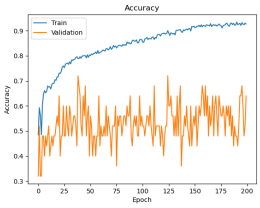

Figure 10: Accuracy for ResNet-50 of three classes

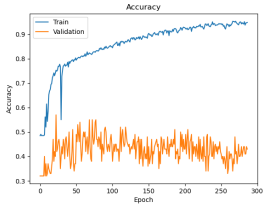

Figure 11: Accuracy for ResNet-101 of three classes

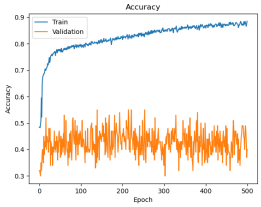

Figure 12: Accuracy for ResNet-152 of three classes

6 Conclusion

In general, we find that the double binary classifiers are better at classifying the COVID images than three-classes classifier. The reason is shown above. The feature distribution of the normal cases is different from the distribution of those infected cases. A more specifically trained classifier is able to capture

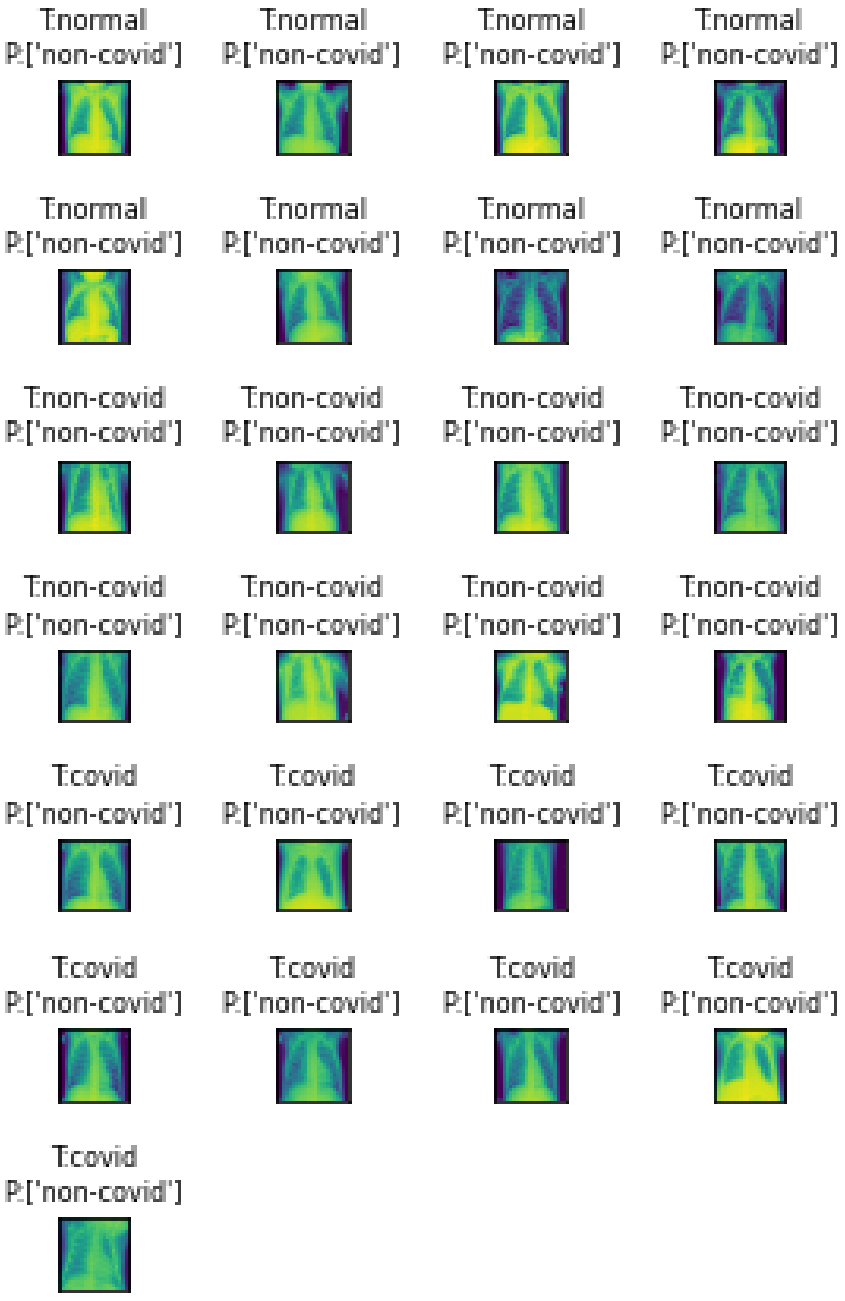

Figure 13: The prediction on validation set

the feature distribution of the COVID and Non-COVID cases better. However, we are surprised that the accuracy for the double binary classifier is still quite low compared to a "useful" classifier. This make us wonder if COVID is distinguishable from other lung infections, by viewing only at patients' chest X-ray images, and if it makes bio-medical sense to do so. According to our observation by training such classifiers, we suppose that it is not suitable to use only chest X-ray images to analyze whether a person is infected by COVID or some other source of infections, e.g., fungi, bacteria.
In short, it is very important to take domain knowledge into account instead of completely relying on the induction provided by deep learning because induction might be biased and inaccurate, which will lead the model to the wrong direction. Moreover, some specific problems might not be solvable, given some certain data. Diseases can share the same syndromes, different classes can also share the same features. It is very important for us to figure out if the problem is solvable given certain data first, before we ever build any models to train.

References

[1] Kaiming He et al. *Deep Residual Learning for Image Recognition*. 2015. arXiv: 1512.03385 [cs.CV].

[2] Liqa A. Rousan et al. "Chest x-ray findings and temporal lung changes in patients with COVID-19 pneumonia". English. In: *BMC Pulmonary Medicine* 20.1 (Sept. 2020). DOI: 10.1186/s12890-020-01286-5.