50.039 Deep Learning Small Project Report

Huang Yizhe 1003697 Nguyen Minh Dang 1003615 Xu Jiayue 1003628

Introduction

This project aimed to develop a deep learning model for classification between COVID-19 pneumonia, non-COVID-19 pneumonia, and healthy chest X-ray images.

Dataset and Data Preprocessing

The dataset consists of in total 1493, 2780 and 1583 chest X-ray images for COVID-19 pneumonia, non-COVID-19 pneumonia, and healthy patients. The distribution of different dataset is shown in Fig.1.

As shown in Fig 1.2 and Fig 1.3, the dataset is relatively balanced for the 3 classes - normal, infected_covid and infected_non_covid. The train-validation-split ratio is around 9:1, with a rather small test set size of 25.

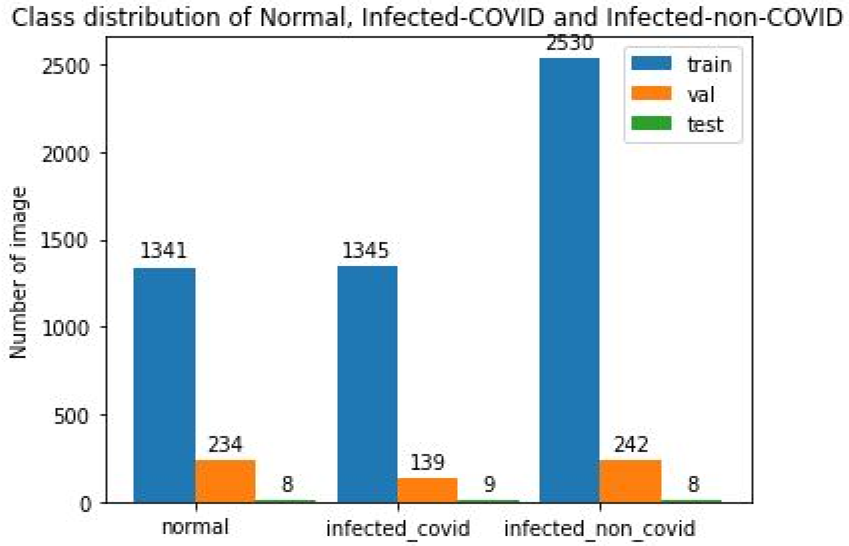


Fig.1.1 Class distribution of Normal, Infected-COVID and Infected-non-COVID images (Bar chart)

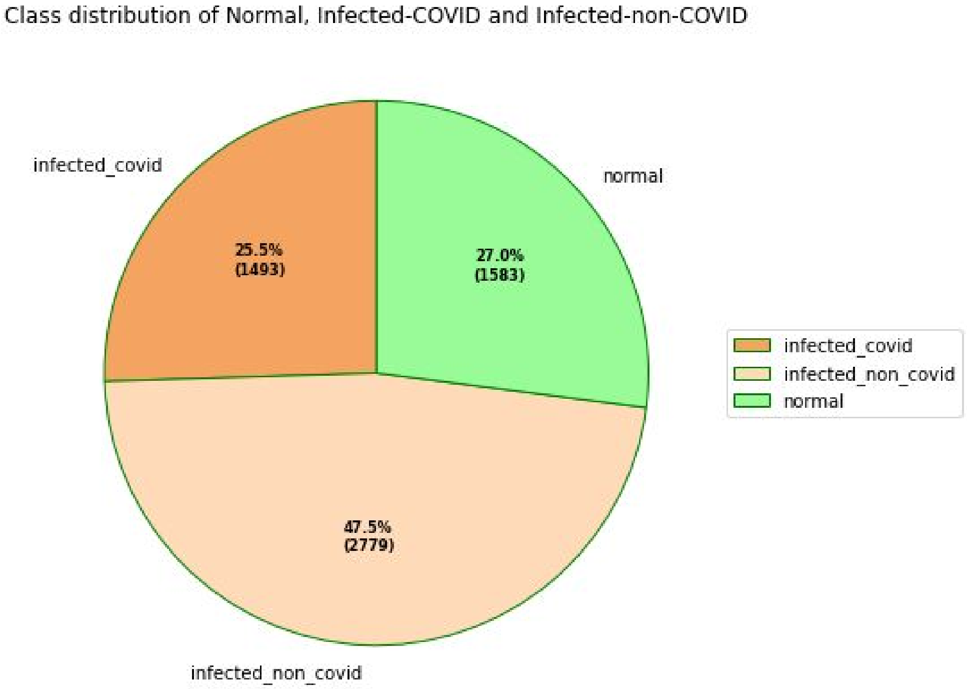


Fig.1.2 Class distribution of Normal, Infected-COVID and Infected-non-COVID images (Pie chart)

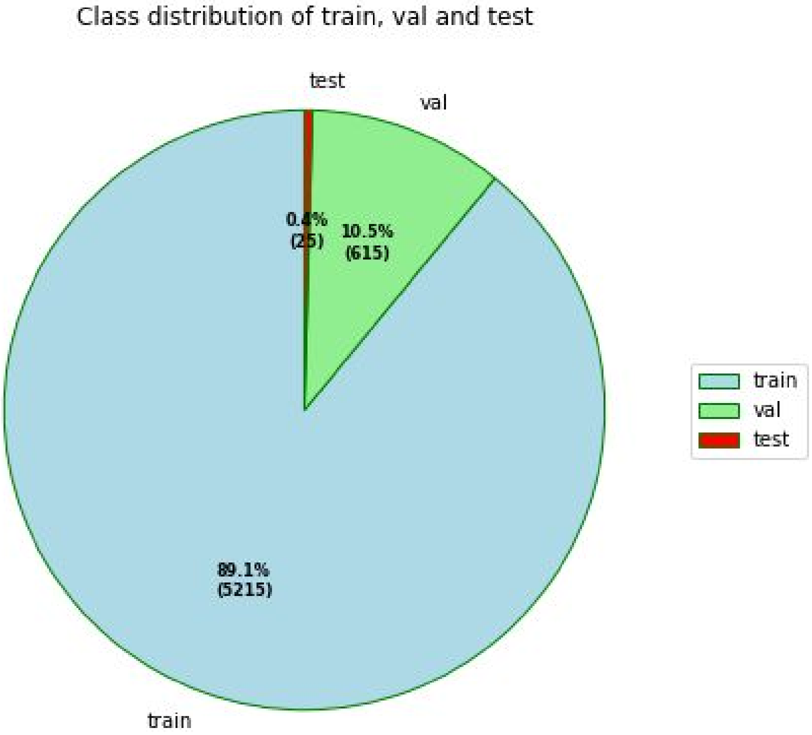


Fig.1.3 Class distribution of train, val and test datasets (Pie chart)

Data preprocessing includes standardisation of grayscale images, and batch normalization to the range of [-1, 1]. Typically, normalization is implemented to reduce redundant information in a dataset, and ensure the common features can be extracted from the same gradient calculation process. It also helps to increase the learning speed, as datasets are standardised to the same range.

Data augmentation

Data augmentation is a great tool that can provide new images that preserve original features, but it can also generate noise that can be harmful to the training phase. As an example, applying rotations and flips for detecting an object in an image, is a measure usually taken, however, it would not make sense for us to apply the same geometrical augmentations for classifying a number like 2, 5 or even 6, 9. If the purpose was to recognize a certain object in an image, then rotation could be useful, but for detecting COVID-19 in an X-ray image, rotation could harm the training phase as most x-rays are taken in the same direction. Furthermore, the accuracy of the deep learning will be heavily impacted by the rotation degree.

Applying the augmentation step must be sensible and align with the existing pattern in this case. In other words, the network trained with augmentation needs to be more robust and accurate than expected variations of the same X-Ray images.

1. Rotation: Since most chest X-rays are taken in a standard direction, it is unlikely for any chest X-ray to be taken in a rotated format, thus this can add noise to the model which is undesirable.
2. Reflection: Reflection would lead to non-physiologic images (left lung ends up in the right thorax, right lung in the left, ect.), which will confuse the network.
3. Translation: Translation might be a useful augmentation step. This is because the X-ray images do not always produce lungs in the center of the image, it depends on the patient's position. Having X-ray images where the lungs are centered could lead to a more robust classifier.

Model Architecture

Out of the two possible implementations of architectures, we decided to use the 3-classes classifier architecture.

The 2 binary classifier architecture is under the assumption that covid cases are intrinsically more similar to other pneumonia cases compared to normal cases, hence it would make sense to use a binary classification first to separate the normal and the "infected". However, if that is not the case, which means that whether covid infected patient or non-covid infected patient data shows more assemblance with the normal batch, the first binary classification step is very likely to have a rather low accuracy, which will cause the second classifier to have an "inaccurate" input, leading to more problems.

Since multi-class classifiers does not oppose any assumptions about the underlying relations between each dataset, we choose to use 3 classes classifier which provides confidence score for each class, from which we select the one with highest probability score to label the data.

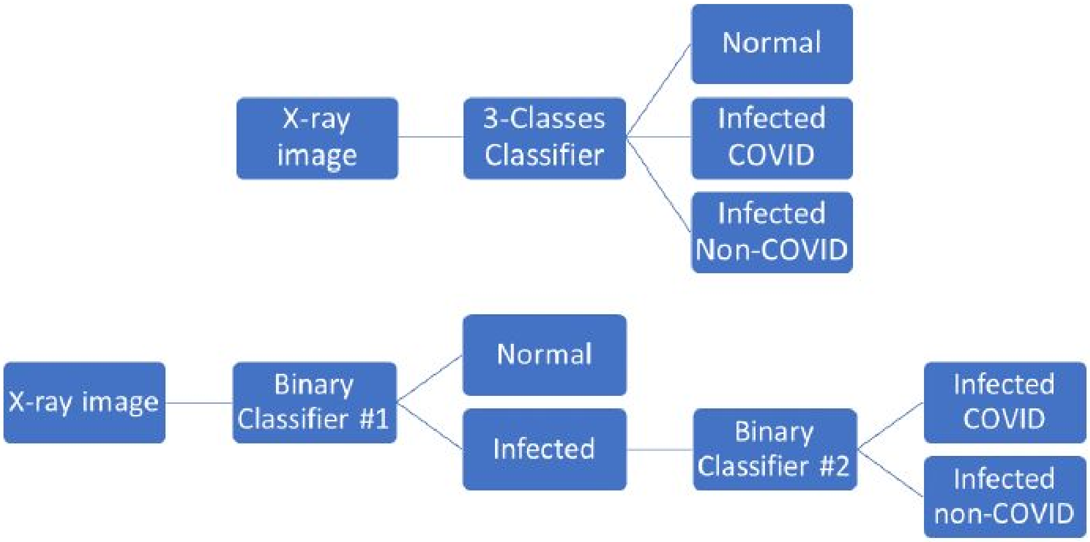
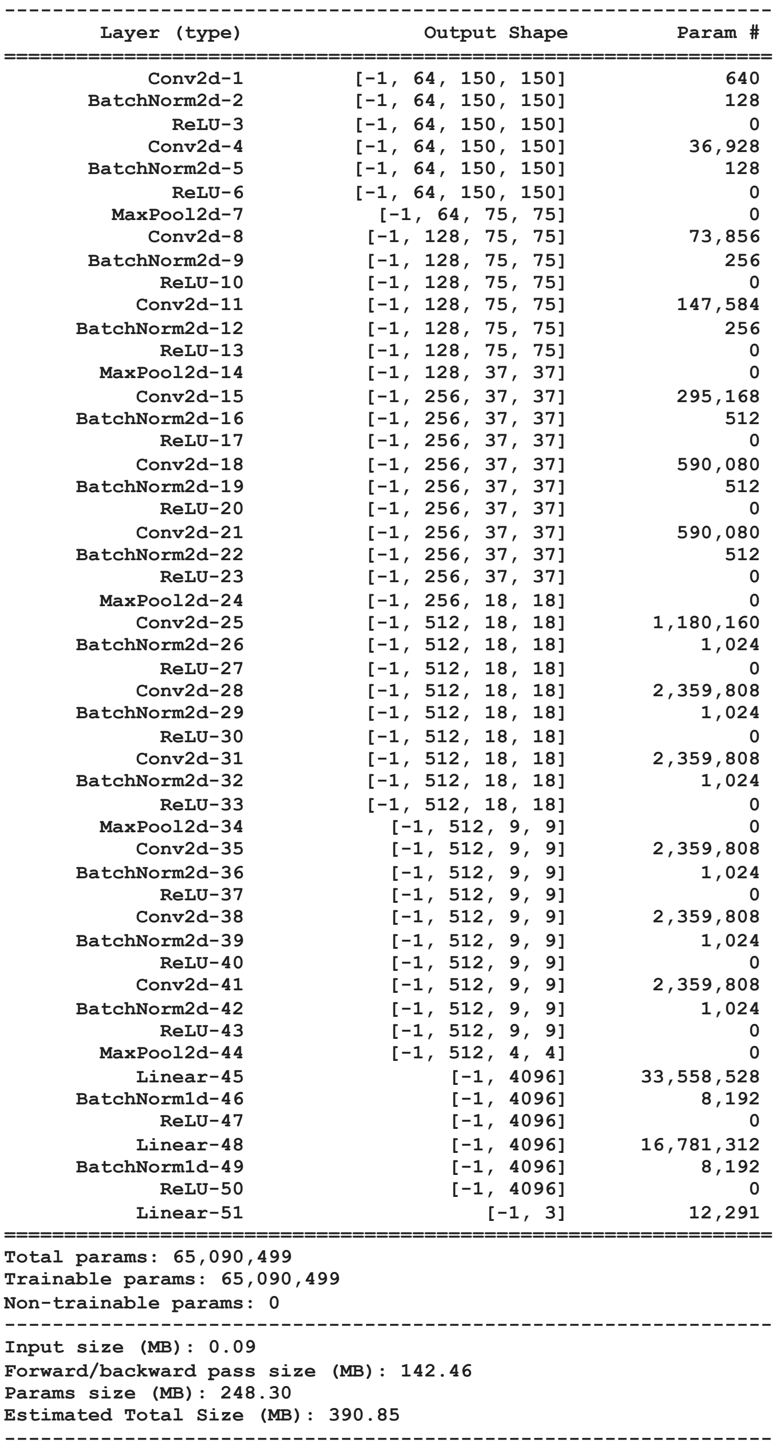


Fig.2 Two possible implementations of architectures

Fig.3 shows a summary of architecture of the implemented model. The model is trained from scratch without using pretrained weight from other architectures. We got the inspiration of using repeated convolutional blocks from researching on several VGG models (Zhang, Zou, He, & Sun, 2016). We used a variant of the VGG-16 architecture, with a batch normalization layer added after each 2D convolution layer. Doing so speeds up the training time and helps provide regularization effects (Yim, 2020).



```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1         [-1, 64, 150, 150]             640
       BatchNorm2d-2         [-1, 64, 150, 150]             128
              ReLU-3         [-1, 64, 150, 150]               0
            Conv2d-4         [-1, 64, 150, 150]          36,928
       BatchNorm2d-5         [-1, 64, 150, 150]             128
              ReLU-6         [-1, 64, 150, 150]               0
         MaxPool2d-7           [-1, 64, 75, 75]               0
            Conv2d-8          [-1, 128, 75, 75]          73,856
       BatchNorm2d-9          [-1, 128, 75, 75]             256
             ReLU-10          [-1, 128, 75, 75]               0
           Conv2d-11          [-1, 128, 75, 75]         147,584
      BatchNorm2d-12          [-1, 128, 75, 75]             256
             ReLU-13          [-1, 128, 75, 75]               0
        MaxPool2d-14          [-1, 128, 37, 37]               0
           Conv2d-15          [-1, 256, 37, 37]         295,168
      BatchNorm2d-16          [-1, 256, 37, 37]             512
             ReLU-17          [-1, 256, 37, 37]               0
           Conv2d-18          [-1, 256, 37, 37]         590,080
      BatchNorm2d-19          [-1, 256, 37, 37]             512
             ReLU-20          [-1, 256, 37, 37]               0
           Conv2d-21          [-1, 256, 37, 37]         590,080
      BatchNorm2d-22          [-1, 256, 37, 37]             512
             ReLU-23          [-1, 256, 37, 37]               0
        MaxPool2d-24          [-1, 256, 18, 18]               0
           Conv2d-25          [-1, 512, 18, 18]       1,180,160
      BatchNorm2d-26          [-1, 512, 18, 18]           1,024
             ReLU-27          [-1, 512, 18, 18]               0
           Conv2d-28          [-1, 512, 18, 18]       2,359,808
      BatchNorm2d-29          [-1, 512, 18, 18]           1,024
             ReLU-30          [-1, 512, 18, 18]               0
           Conv2d-31          [-1, 512, 18, 18]       2,359,808
      BatchNorm2d-32          [-1, 512, 18, 18]           1,024
             ReLU-33          [-1, 512, 18, 18]               0
        MaxPool2d-34            [-1, 512, 9, 9]               0
           Conv2d-35            [-1, 512, 9, 9]       2,359,808
      BatchNorm2d-36            [-1, 512, 9, 9]           1,024
             ReLU-37            [-1, 512, 9, 9]               0
           Conv2d-38            [-1, 512, 9, 9]       2,359,808
      BatchNorm2d-39            [-1, 512, 9, 9]           1,024
             ReLU-40            [-1, 512, 9, 9]               0
           Conv2d-41            [-1, 512, 9, 9]       2,359,808
      BatchNorm2d-42            [-1, 512, 9, 9]           1,024
             ReLU-43            [-1, 512, 9, 9]               0
        MaxPool2d-44            [-1, 512, 4, 4]               0
           Linear-45                 [-1, 4096]      33,558,528
      BatchNorm1d-46                 [-1, 4096]           8,192
             ReLU-47                 [-1, 4096]               0
           Linear-48                 [-1, 4096]      16,781,312
      BatchNorm1d-49                 [-1, 4096]           8,192
             ReLU-50                 [-1, 4096]               0
           Linear-51                    [-1, 3]          12,291
================================================================
Total params: 65,090,499
Trainable params: 65,090,499
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.09
Forward/backward pass size (MB): 142.46
Params size (MB): 248.30
Estimated Total Size (MB): 390.85
----------------------------------------------------------------
```

Fig.3 Model summary

Considering the size of the training set and the validation set, we choose 64 as the batch size for the training set, and 32 as the batch size for the validation set.

As the model is a multi-class classifier, Cross entropy loss function is unsuitable to use in this case. Hence we choose Negative log-likelihood (NLL) loss function, which helps to prevent vanishing gradient problems.

We choose Adam optimizer as it is computationally efficient, and well suited for problems that are large in terms of data parameters. We have tried a list of learning rates ranging from 0.001 to 1.0, and 0.01 learning rate provides the most promising result.

Default initialization of pytorch is used for our model, as it is good enough to get random weights from normal distribution and train from scratch.

The loss and accuracy plot is shown in Fig.4, the model accuracy steadily climbed up over iterations, the highest spike is around 0.8. The validation loss is still a bit higher than training loss, suggesting overfitting which is already addressed by tuning the parameters and adjusting our learning rate.

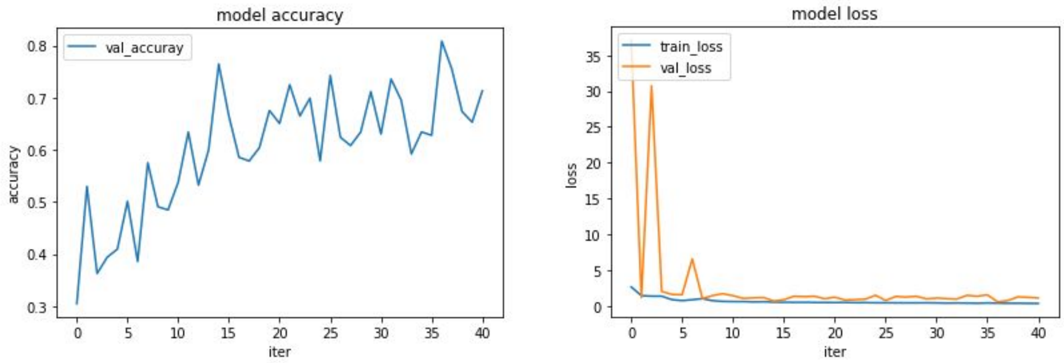


Fig.4 Loss and accuracy v.s. iterations

Results, Discussion and Analysis

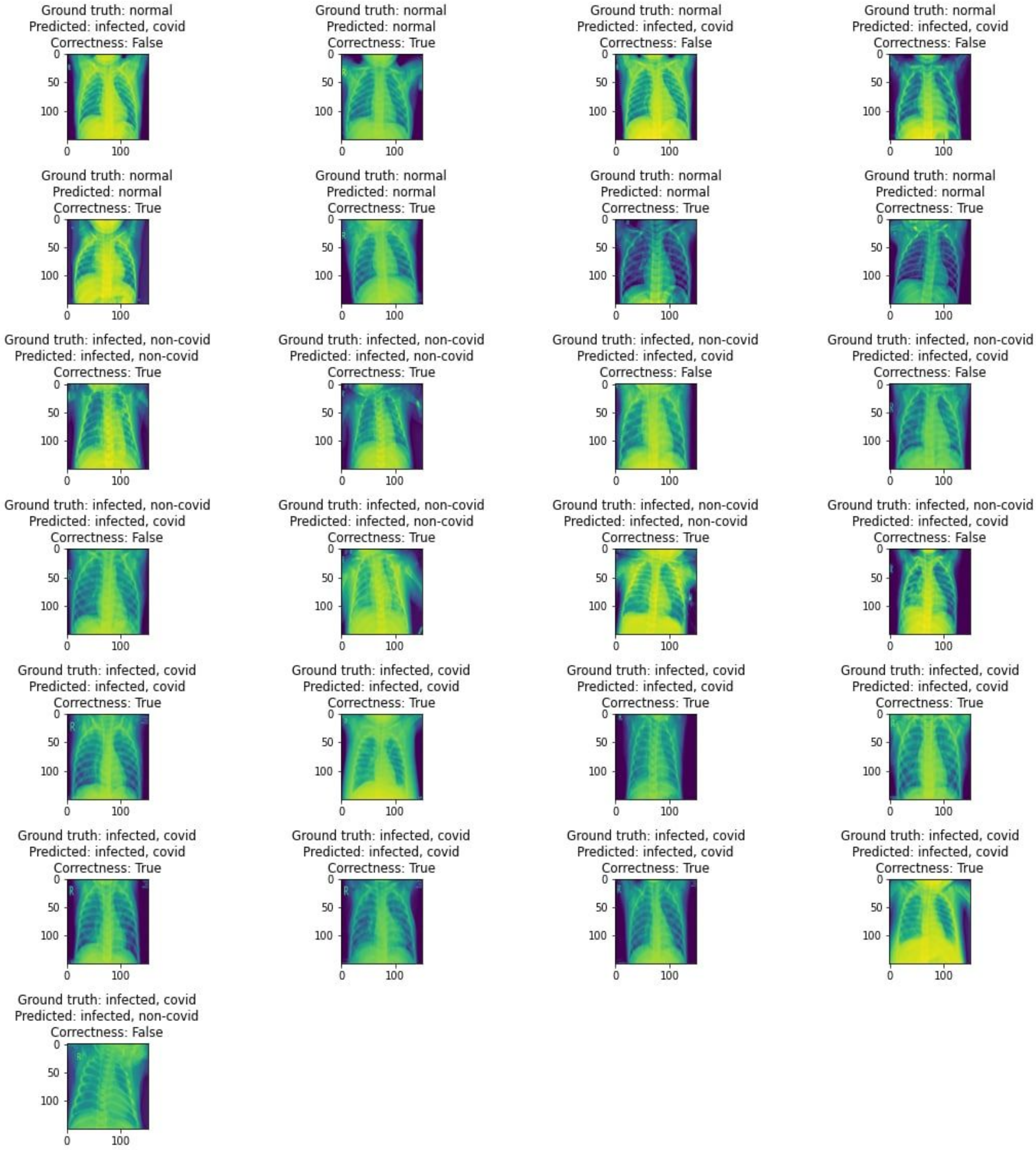

Ground truth: normal
Predicted: infected, covid
Correctness: False
Ground truth: normal
Predicted: normal
Correctness: True
Ground truth: normal
Predicted: infected, covid
Correctness: False
Ground truth: normal
Predicted: infected, covid
Correctness: False
Ground truth: normal
Predicted: normal
Correctness: True
Ground truth: normal
Predicted: normal
Correctness: True
Ground truth: normal
Predicted: normal
Correctness: True
Ground truth: normal
Predicted: normal
Correctness: True
Ground truth: infected, non-covid
Predicted: infected, non-covid
Correctness: True
Ground truth: infected, non-covid
Predicted: infected, non-covid
Correctness: True
Ground truth: infected, non-covid
Predicted: infected, covid
Correctness: False
Ground truth: infected, non-covid
Predicted: infected, covid
Correctness: False
Ground truth: infected, non-covid
Predicted: infected, covid
Correctness: False
Ground truth: infected, non-covid
Predicted: infected, non-covid
Correctness: True
Ground truth: infected, non-covid
Predicted: infected, non-covid
Correctness: True
Ground truth: infected, non-covid
Predicted: infected, covid
Correctness: False
Ground truth: infected, covid
Predicted: infected, covid
Correctness: True
Ground truth: infected, covid
Predicted: infected, covid
Correctness: True
Ground truth: infected, covid
Predicted: infected, covid
Correctness: True
Ground truth: infected, covid
Predicted: infected, covid
Correctness: True
Ground truth: infected, covid
Predicted: infected, covid
Correctness: True
Ground truth: infected, covid
Predicted: infected, covid
Correctness: True
Ground truth: infected, covid
Predicted: infected, covid
Correctness: True
Ground truth: infected, covid
Predicted: infected, covid
Correctness: True
Ground truth: infected, covid
Predicted: infected, non-covid
Correctness: False
0
50
100
0
100

Fig.5 Test set presentation

	Expected					
Predicted		**Normal**	**Infected (covid)**	**Infected (non-covid)**	**Classification Overall**	**Accuracy (precision)**
	Normal	5	0	0	5	100.0%
	Infected (covid)	3	8	4	15	53.3%
	Infected (non-covid)	0	1	4	5	80.0%
	Truth overall	8	9	8	-	-
	Accuracy (recall)	62.5%	88.9%	50%	-	-
Overall accuracy		68%				

Fig.6 Result Matrix

Out of the 8 False cases: 3 normal cases predicted as covid patients, 4 non-covid patients predicted as covid patients and 1 covid patient predicted as non-covid patient.

“You might find it more difficult to differentiate between non-covid and covid x-rays, rather than between normal x-rays and infected (both covid and non-covid) people x-rays. Was that something to be expected? Discuss.”

Like other pneumonias, covid-19 pneumonia causes the density of the lungs to increase. This may be seen as whiteness in the lungs on radiography which, depending on the severity of the pneumonia, obscures the lung markings that are normally seen; however, this may be delayed in appearing or absent. When lung markings are partially obscured by the increased whiteness, a ground glass pattern occurs. But sometimes benign calcifications can also present themselves as such, hence those cases take experienced radiologists to differentiate them. Since many lung markings are representative for both covid and non-covid pneumonias, to differentiate them is obviously harder compared to differentiate between them and the normal cases in which the image showed way less lung disease markers.

Other than normal pneumonias, conditions such as other atypical pneumonias and the early stages of community acquired pneumonias, pulmonary aspiration, pulmonary oedema, lung cancer etc can also cause ground glass appearance, consolidation, and linear opacities.

Hence, due to the shared pathological symptoms and how they are presented by the chest X-ray, we may find it harder to differentiate the non-covid pneumonia cases and covid cases.

“**Final question: would it be better to have a model with high overall accuracy or low true negatives/false positives rates on certain classes? Discuss.”**

Throughout this project we kept reminding ourselves about what we are trying to achieve. We believe that during a pandemic situation, it would be more reasonable to develop a model which sometimes provides more false positive results than having a model which would be more likely to give false negative results, and accuracy in this case can be sacrificed a bit in order to achieve that.

The underlying reason for the above statement is that we are dealing with highly infectious diseases, it would be more acceptable for our model to misclassify a healthy person as a covid patient than to have a model which has a higher chance to provide a false negative result, which is allowing a infected patient to pass the test and spread the disease.

To avoid that, we decided to use the set of parameters which provide lowest false negative numbers. **For the validation set, it has only 1 case of false negative, which misclassified a covid patient as non-covid pneumonia patient; this is still a better result since that patient will still not be able to spread the disease**. Although the overall accuracy may suffer from that, we are sure that this is the best approach to solve the real-life problem.

References

1. Zhang, X., Zou, J., He, K., & Sun, J. (2016). Accelerating very deep convolutional networks for classification and detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 38(10), 1943-1955. doi:10.1109/tpami.2015.2502579
2. Yim, M. (n.d.). Msyim/vgg16. Retrieved March 21, 2021, from https://github.com/msyim/VGG16
3. Elgendi M, Nasir MU, Tang Q, Smith D, Grenier J-P, Batte C, Spieler B, Leslie WD, Menon C, Fletcher RR, Howard N, Ward R, Parker W and Nicolaou S (2021) The Effectiveness of Image Augmentation in Deep Learning Networks for Detecting COVID-19: A Geometric Transformation Perspective. Front. Med. 8:629134. doi: 10.3389/fmed.2021.629134