50.039 Deep Learning Small Project Report

Ong Li-Chang 1003328

Noorbakht Khan 1003827

Richard Pung 1003523

Github Code Link

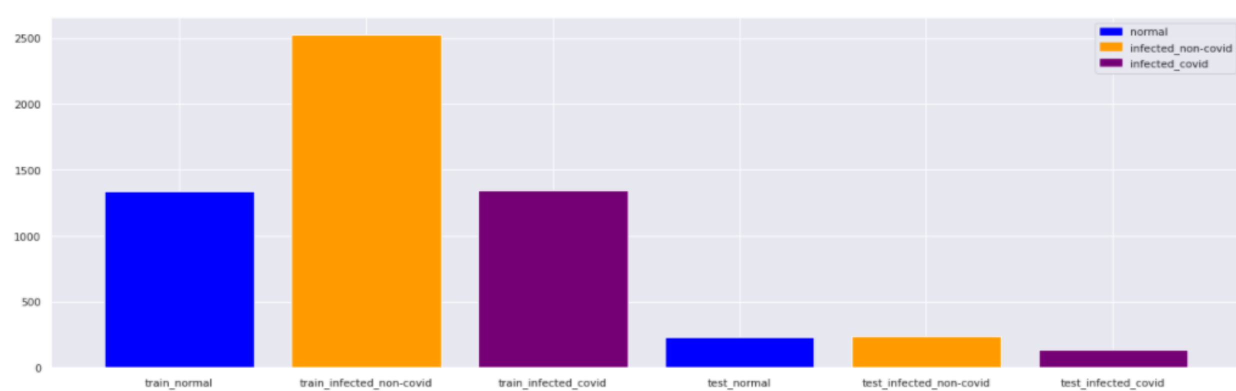
https://github.com/Noorbakht/50.039_Small_Project

Dataset and Dataloader Expectations

We were planning to try both suggested architectures and evaluate the accuracy of our results before finalising on the architecture that we are using. Thus, we conducted dataset exploration on both possible types of datasets. The results of which are shown below.

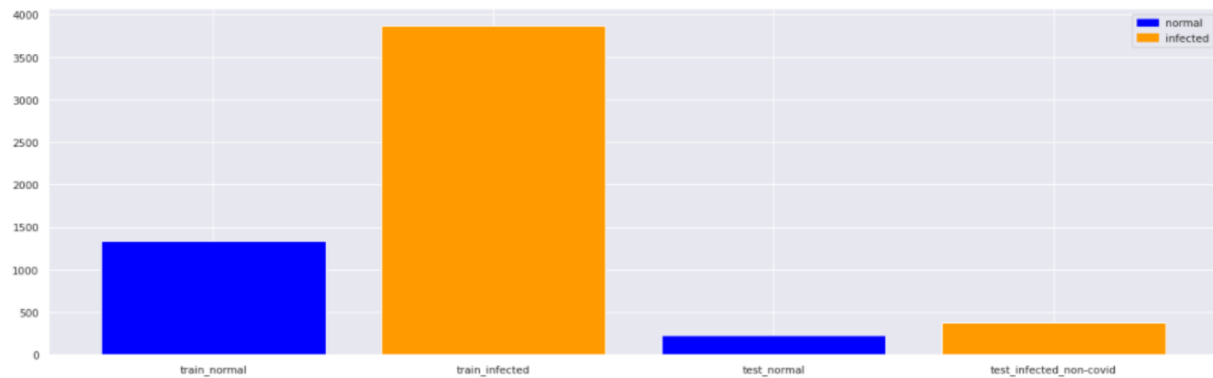
Dataset Exploration

Multi-class Exploration (normal vs infected_non-covid vs infected_covid)



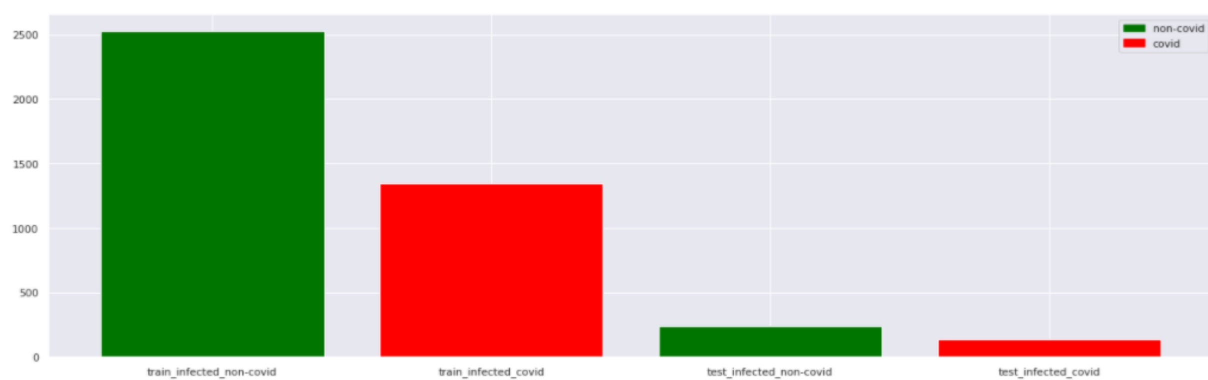
The above graph shows the data distribution for normal, infected_non-covid and infected_covid samples. On the train side, we see that the majority class is the infected_non-covid class which is about 2x bigger than both the normal and infected_covid samples. On the test set we only see a minor imbalance where normal and infected_non-covid are about equal in size and infected_covid is about 0.5x the size of the other two classes.

Binary Class Exploration (normal vs infected)



For the above graph, we combined the infected_non-covid images and infected_covid images to form the larger infected class. As you can see from the graph above, the dataset is unbalanced, skewing heavily towards our infected samples on both the train and test set. The train_infected is about 3.5x the train_normal size whilst the test set are closer in size with about 0.5x discrepancy.

Binary Class Exploration (infected_non-covid vs infected_covid)

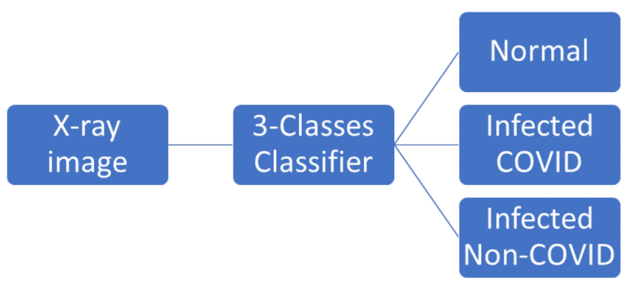


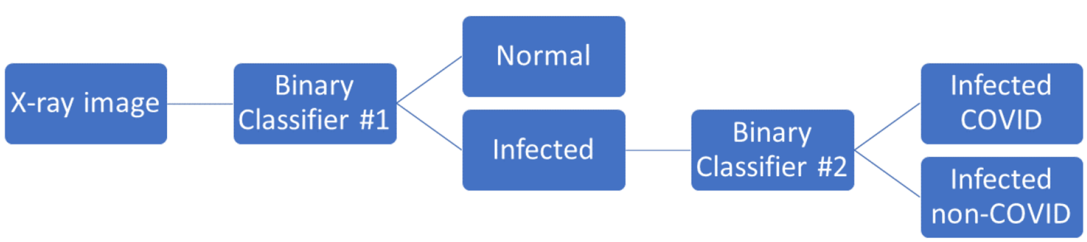
For the above graph, we compare only the distribution of the infected_non-covid images and infected_covid images. As you can see from the graph above, the dataset is unbalanced, skewing heavily towards our non-covid samples on both the train and test set. The non-covid train and test set is about 2x the size of the covid train and test set respectively.

Data Processing/Augmentation Operations

For our pre-processing, we decided to perform 3 operations. Using the CV2 package, we resized our images and removed gaussian noise. We then normalized all our pixels to fit between a range of [0,1]. On all datasets (Train, Test, Val), we resized our images to the standard dimension of 150x150 to ensure that our data input size would match that of the NN input size. Next, based on online resources[1], we found that gaussian blurring was widely used when building computer vision algorithms. It helps to reduce image noise and reduce detail which in turns helps to enhance different structures at different scales. Based on our testing compared to no gaussian blurring, we saw a +1-3% average increase of accuracy over 5 tests. Lastly we applied input normalisation to ensure that the NN received input in [0, 1]. This is done to ensure that each pixel has a similar data distribution and makes convergence faster while training the network.

Proposed Model Expectations





1 https://www.w3.org/Talks/2012/0125-HTML-Tehran/Gaussian.xhtml#:~:text=Gaussian%20blurring%20is%20commonly%20used,the%20downsampled%20image%20(aliasing).

The first architecture is a multi-class classifier that classifies 3 classes into normal, infected COVID and infected non-COVID. On the other hand, the second architecture first classifies our classes into normal or infected. If the class is labelled as infected, it passes through another binary classifier that classifies it as either infected COVID or infected non-COVID.

The first architecture will have the entire training data being passed into our classifier while the second architecture will have the entire training data passed into the first classifier and a smaller subset of training data passed into the second classifier. Given the reduction in the number of samples, the second classifier in the second architecture may be more susceptible to over-fitting.

Thus, we decided to try both architectures and evaluate the accuracy of our results before finalising on the architecture that we are using.

Results:

	Accuracy (%)
Multi-class Classifier Model (Normal vs Infected Covid vs Infected Non-Covid)	67

Table 1: Results of Multi-class Classifier on Test Sets

	Accuracy of First Classifier - Normal vs Infected (%)	**Accuracy of Second Classifier - Non-Covid vs Covid (%)**	**Average Accuracy (%)**
Binary Classifier Model	74	86	80

Table 2: Results of Binary Classifier on Test Sets

Based on the results we obtained, we decided to pursue the second architecture and fine-tune the model that we created.

Architecture

For our model architecture, we took inspiration from the DarkCovidNet[2] model. The DarkCovidNet model uses the Darknet-19 model as their base for research but only consists of 17 convolutional layers instead of 19. In each layer of our model architecture, there is one convolutional layer followed by BatchNorm and ReLu. The batch normalization layer helps to reduce the gradient vanishing problem by normalizing the inputs before being passed onto the next layer.

Our model architecture uses triple convolutions with stride of 1 and kernel size of 3 which empirically performs for our task. Our model is unique in that it uses two fully connected layers for the classifier which would allow it to learn a higher order feature such that it can distinguish between the samples. We added max pooling layers as a form of regularization to prevent the model from overfitting.

We are using Cross Entropy Loss for a binary classification problem like ours. Since we added a LogSoftmax layer in the last layer of our network, we used the inbuilt loss function Negative Log-Likelihood loss function (NLL). This loss function will help to calculate a score that summarizes the average difference between the actual and predicted probability distributions for predicting class 1. The score is minimized and a perfect cross-entropy value is 0.

For our mini-batch size, we first went with the value 8. However, due to the fact that our model took a long time to run, we decided to increase this value to 32 to increase the run time of our models being trained. As for our optimizer, we are using an Adam optimizer with a learning rate of 0.001. We initially trained our model with a learning rate set to 1.0. However, we realised that our training loss decreased to a value before increasing again. Thus, we set our learning rate to 0.001 to allow our model to learn a more optimal set of weights despite taking slightly longer to

---

2 https://medium.com/visionwizard/darkcovidnet-automated-detection-of-covid-19-with-x-ray-images-c4bfc29eb06c

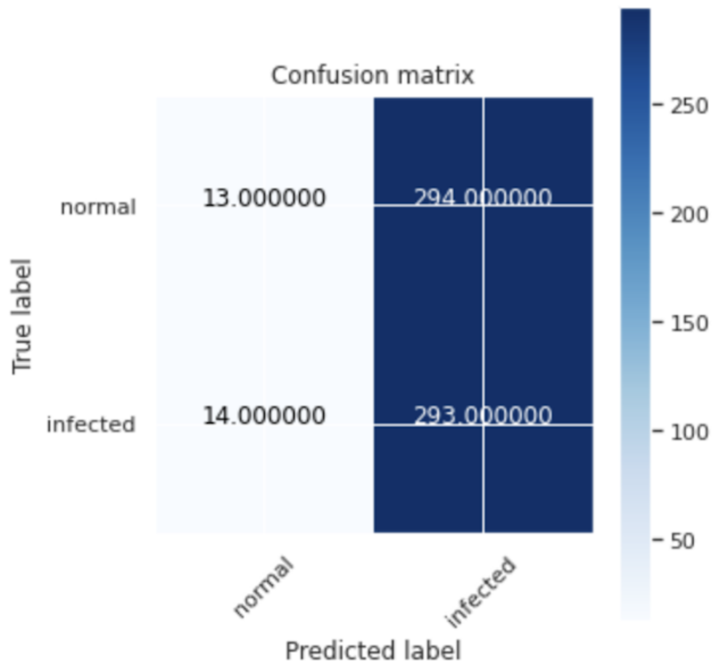
train. We also used the standard random small weight initialization. This is to reduce the likelihood of vanishing/exploding gradient issues that we may have.

Training And Testing Losses

First Binary Classifier:

Training and Testing Losses:

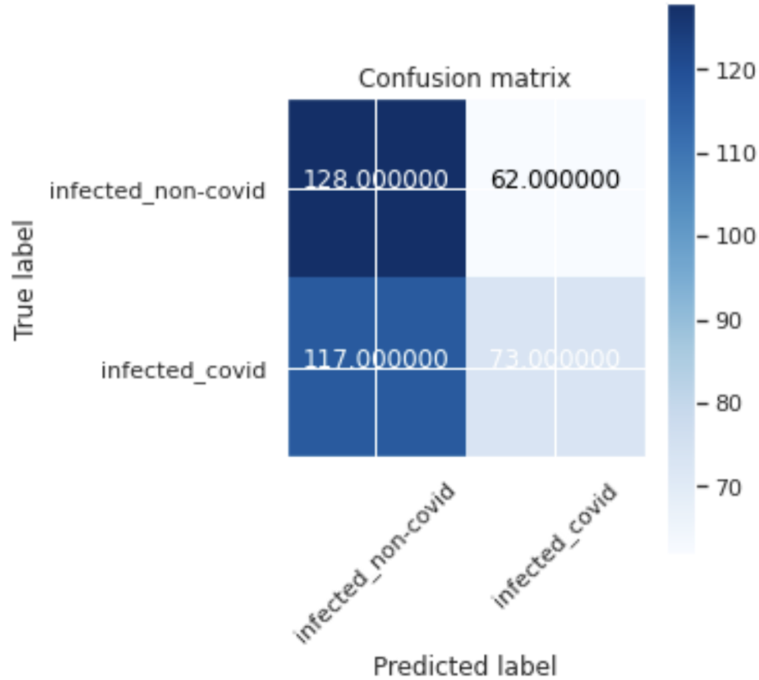
Confusion Matrix:



Second Binary Classifier:

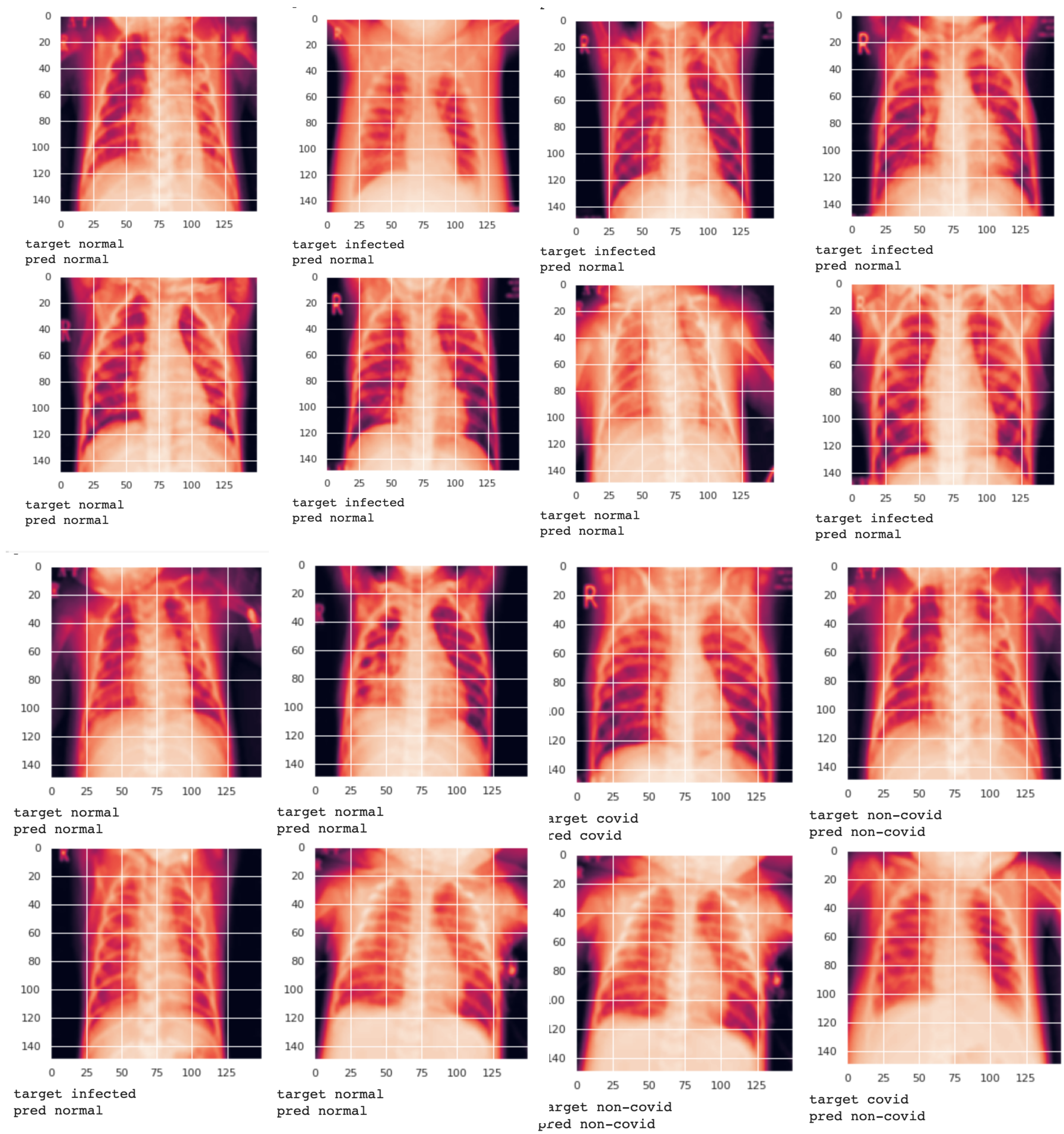
Training and Testing Losses:

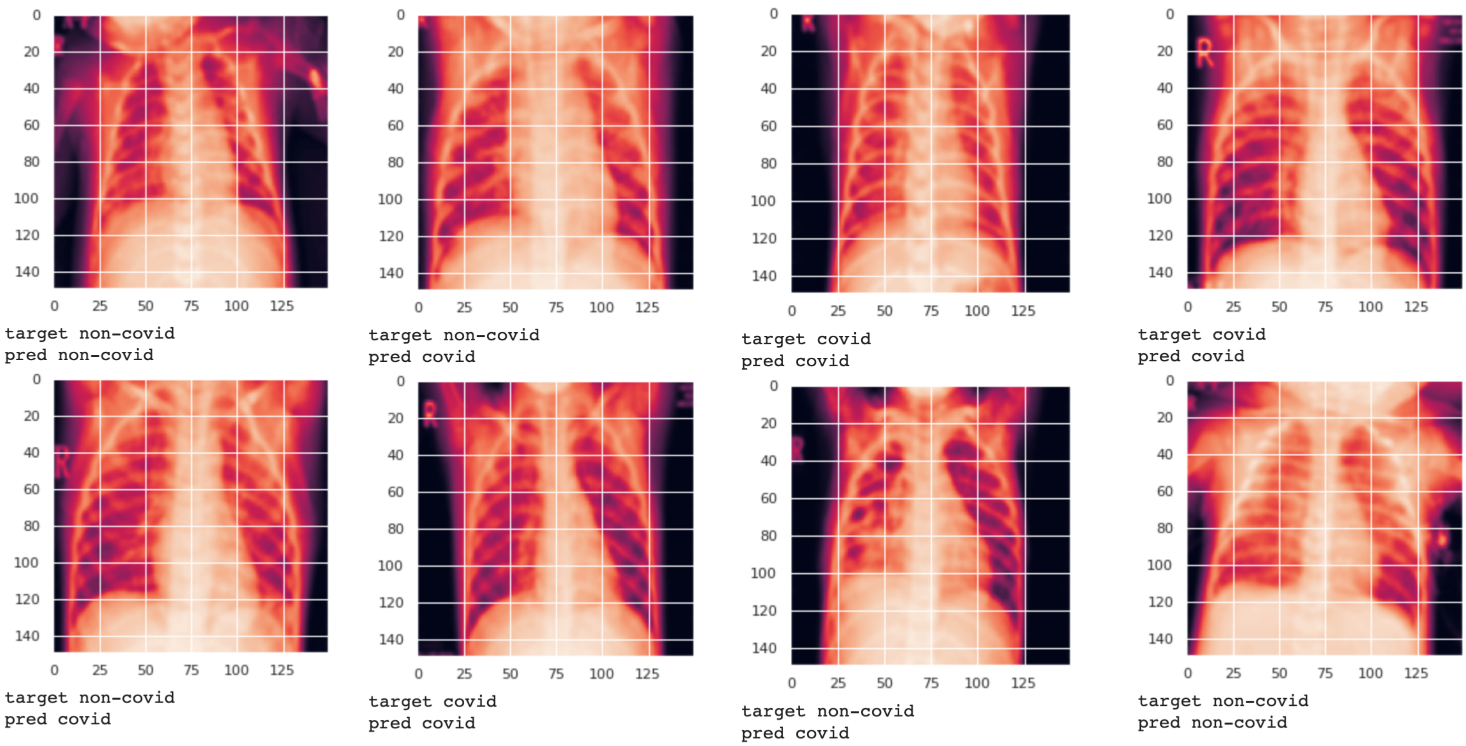
Confusion Matrix:



Validation Set Performance

We obtained a validation accuracy of 50% for our first model that distinguishes between normal and infected. For our second model, we obtained an accuracy of 68.75%.


target normal
pred normal
target infected
pred normal
target infected
pred normal
target infected
pred normal
target normal
pred normal
target infected
pred normal
target normal
pred normal
target infected
pred normal
target normal
pred normal
target normal
pred normal
arget covid
red covid
target non-covid
pred non-covid
target infected
pred normal
target normal
pred normal
arget non-covid
pred non-covid
target covid
pred non-covid



Explanation of why identifying Normal/Infected is easier than Non-Covid/Covid

Predicting normal from infected is easier for the model to learn than predicting infected_non-covid from infected_covid. This is due to the fact that an infection stemming from non-covid and covid cases both leads to inflammations in the lungs which can be seen in the x-ray. As such, the images for infected_non-covid and infected_covid look similar, making it more difficult for the model to distinguish between the two.

Would it be better to have a model with high overall accuracy or low true negatives/false positives rates on certain classes? Discuss.

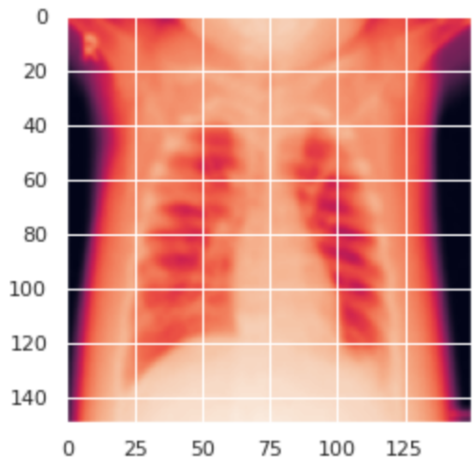
Given the time critical nature of treating Covid, if a patient is diagnosed as infected, we would rather classify said patient as having Covid than missing it out completely. Thus, between the 2 choices, if we take the infected_covid class as our positive class, we would rather have a model that has high overall accuracy rather than one that has low true negatives. Low true negatives signifies that our model is good at not incorrectly predicting Covid exists when it does not. However, it does not mean that our model will be good at predicting Covid which is the most important thing. Thus, we would prefer the more accurate model over the low true negatives rates model (given that infected_covid is our positive class).

Briefly look up online how doctors diagnose infections based on x-rays. Does our AI seem to be able to reproduce this behavior correctly? Show typical samples of the dataset on which your AI failed and discuss what might have been the reasons.

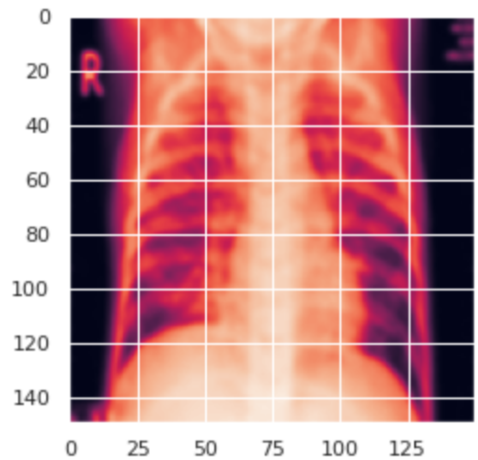
Based on online resources[3], we learnt that 'Chest X-rays produce images of your heart, lungs, blood vessels, airways, and the bones of your chest and spine. Chest X-rays can also reveal fluid in and around your lungs or air surrounding a lung.' Doctors use these X-ray images to determine whether you have a specific condition such as heart problems, a collapsed lung or pneumonia. Specifically for lung infections, such as pneumonia, 'when interpreting the x-ray, the radiologist will look for white spots in the lungs (called infiltrates) that identify an infection. This exam will also help determine if you have any complications related to pneumonia such as abscesses or pleural effusions (fluid surrounding the lungs).'[4]

Our AI is not able to detect these white spots in the lungs with such specificity, often confusing infected images as normal as shown by the image below:

[3] https://www.mayoclinic.org/tests-procedures/chest-x-rays/about/pac-20393494
[4] https://www.radiologyinfo.org/en/info.cfm?pg=pneumonia



target infected
pred normal



target infected
pred normal

As seen from above, our AI is not capable of recognising such patterns (white spots), which a trained expert could easily diagnose upon viewing the x-ray scans. One of the factors that could have led to this is the small size of our dataset, which would have reduced our CNN layers ability to form pattern recognisers. If we had a larger dataset, it is possible that our model would be able to recognise such patterns in due time.

Annex

Resources Used:

1. https://www.w3.org/Talks/2012/0125-HTML-Tehran/Gaussian.xhtml#:~:text=Gaussian%20blurring%20is%20commonly%20used,the%20downsampled%20image%20(aliasing).
2. https://medium.com/visionwizard/darkcovidnet-automated-detection-of-covid-19-with-x-ray-images-c4bfc29eb06c
3. https://machinelearningmastery.com/one-vs-rest-and-one-vs-one-for-multi-class-classification/
4. https://machinelearningmastery.com/how-to-choose-loss-functions-when-training-deep-learning-neural-networks/
5. https://datascience.stackexchange.com/questions/10523/guidelines-for-selecting-an-optimizer-for-training-neural-networks
6. https://www.lexjansen.com/nesug/nesug10/hl/hl07.pdf
7. https://www.mayoclinic.org/tests-procedures/chest-x-rays/about/pac-20393494
8. https://www.radiologyinfo.org/en/info.cfm?pg=pneumonia