Covid Lung Analysis

An Exercise in Image Analysis and Neural Network Modeling

The COVID pandemic is still currently ongoing and even though restrictions have relaxed currently, COVID still affects many people today. Due to the pandemic, there has been a huge influx of patients around the world stretching healthcare providers thin. This in turn makes it difficult for patients with mild ailments/symptoms to get a checkup. This projects is an attempt to create a machine learning model that would be able to predict if a patient has COVID based on lung X-rays in the hopes that it could help healthcare providers and patients alike in their ongoing struggles with COVID.

Audience

Unless a model exists with near 99% accuracy and has been extensively tested on million upon millions of lung images it is unlikely that a healthcare provider would use this model as their only method of diagnosing COVID, however it could be used as a supplement for a provider in deciding their diagnoses. In addition, it could also be used by a patient to determine whether they may need to get a doctor's opinion. However, since this model uses X-ray images (which a patient would not normally get unless a doctor recommends it) it further restricts the use of this model to mostly healthcare providers which limits its useability. In the end, this is mostly an exercise in image analysis with some potential to be further studied by more experienced data scientists and healthcare specialists.

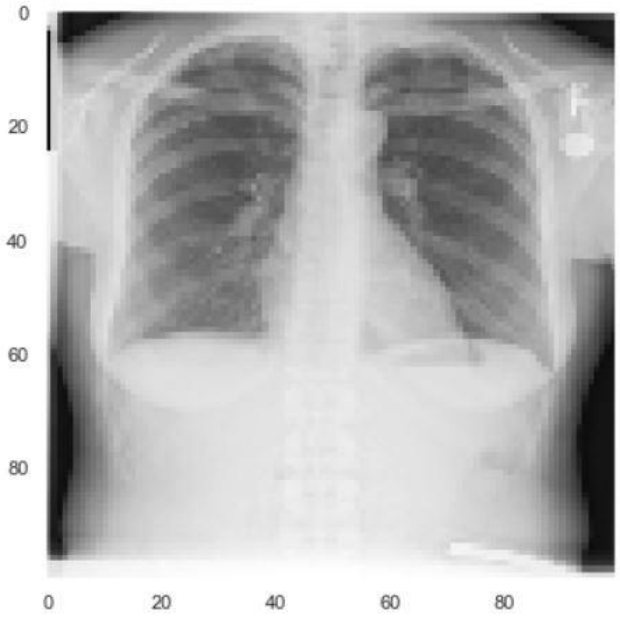
About the Data

The data set originates from Kaggle which contains lung images of both positive and negative COVID patients. There are 13,808 images in total, 10,192 (74%) being negative images and 3,616 (26%) being positive images.

Data Wrangling

To make the images all uniform, I rescaled them to be (100,100) pixels and made sure the images were grayscale. Below is what the image looks like after rescaling.

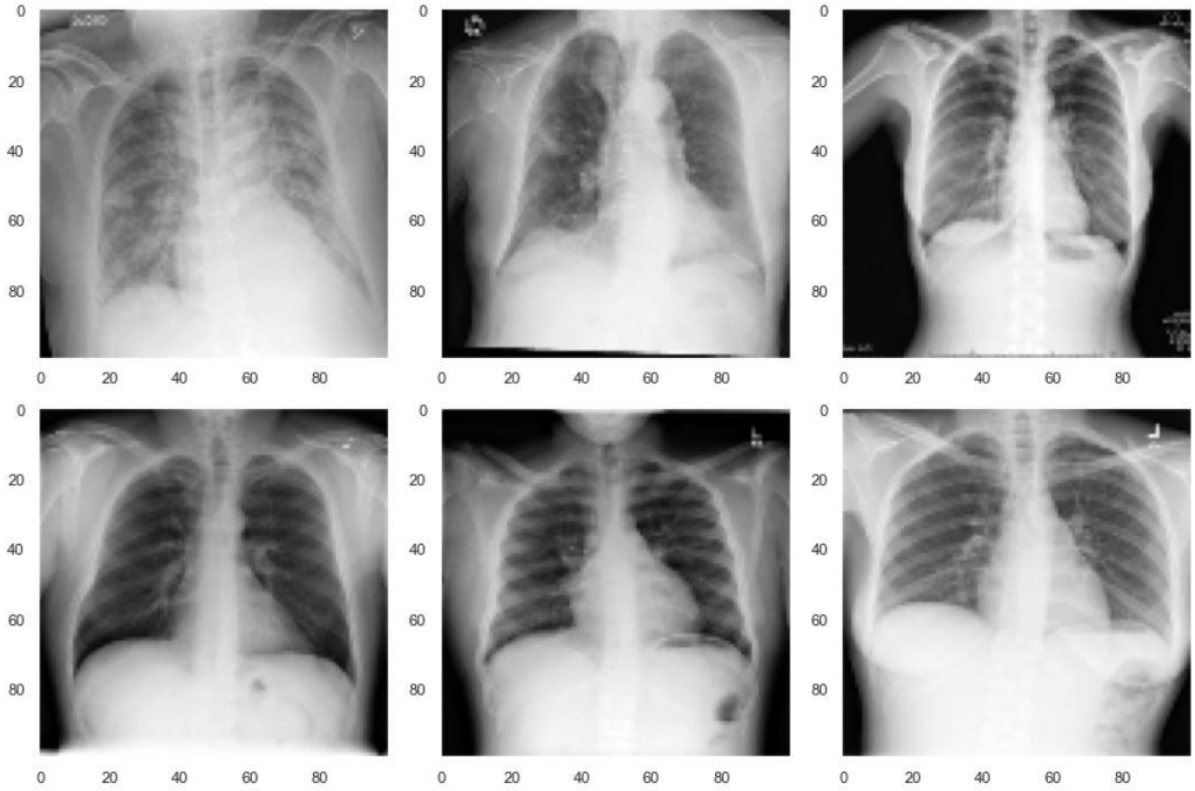
```
Image Dimension    : (100, 100)
Image Height       : 100
Image Width        : 100
Result             : Negative
```



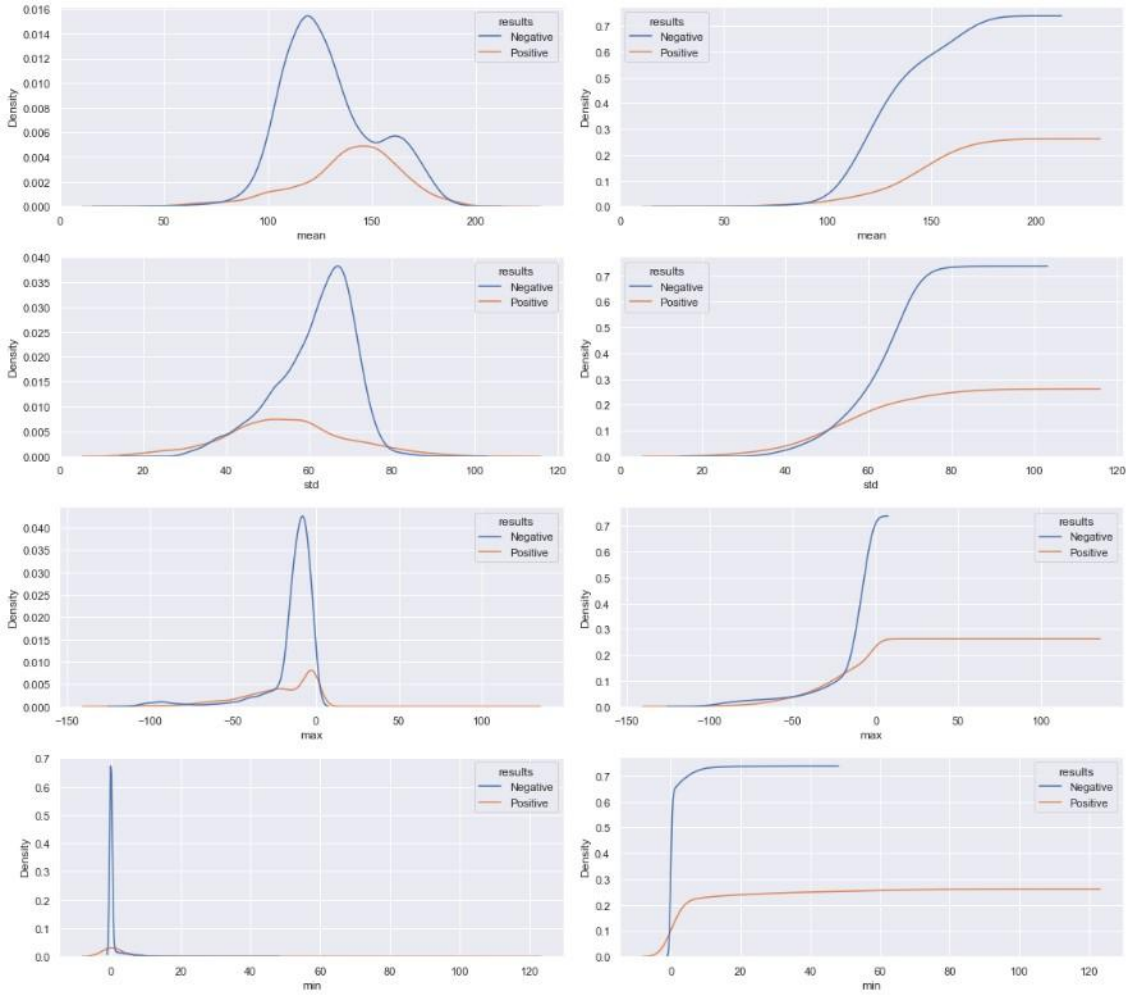
In addition, I also created a set of blurred images and sharpened images to see if they would work better in the machine learning model. The image below shows the three type of images the model would potentially be working with.

EDA

Exploring the data revealed some interesting observations. Below is a set of positive and negative covid lungs.



The top three images are examples of positive lungs, while the bottom three are examples of negative lungs. An interesting observation is that the positive lungs have a higher concentration of white lines across the lungs. This observation is further exemplified in the following KDE plot below.



Looking specifically at the average pixel intensity on the top left graph. It becomes evident that on average, the positive images have higher intensity compared to the negative images which indicates there are more white pixels on average in the positive images.

The graph below shows a scatterplot between the standard deviation and mean of pixel intensities between positive and negative images.

The image shows that positive images tend to stray away from the general cluster while also having a very concentrated cluster. All of these graphs and analysis indicate that there are some differences between positive and negative images which a machine learning model may potentially be able to identify.

Modeling

The first decision required for the model is whether the original image or a manipulated image would be better suited for the model. As mentioned above there were three types of images to choose from, the original, the blurred, and the sharpened image. To test which image was the best, I created a simple neural network model and tested which image type yielded the best validation accuracy.

Image Type	Loss	Accuracy
Original	0.2962	0.86556
Blurred	0.2998	0.86218
Sharpened	0.2445	0.90418

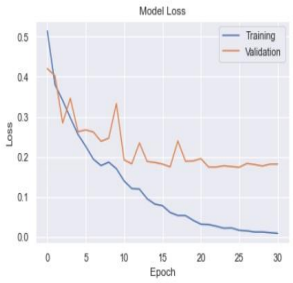
Both the original and blurred images had similar loss and accuracy. Sharpened however had the lowest loss and the highest accuracy which indicates that it is most likely the best type of image to utilize in model building.

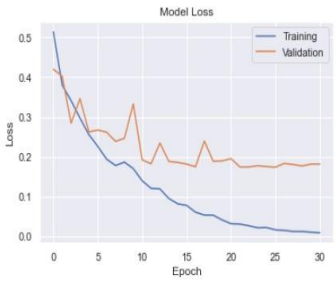
After determining which image type to use, the next step was testing hyperparameters in the model. Given that the test model already yields a 90.4% accuracy, I decided to use that as a base and tune the hyperparameters in that model. The hyperparameters that are going to be tuned in the model are the number of units in the 2D convoluted layer and dense layer, as well as the learning rate and momentum for the SGD optimizer.

The original model had {units: 32, lr:0.001, momentum:0.9, units2:128}.
After tuning the hyperparameters the conclusion was that the best model of the ones that were searched is: {units: 128, lr:0.001, momentum:0.6, units2:128}

Between the two models it was now time to see which one was better after maximizing the validation accuracy of the two models.

Model	Loss	Accuracy
Original	0.187	0.938
Tuned	0.181	0.932

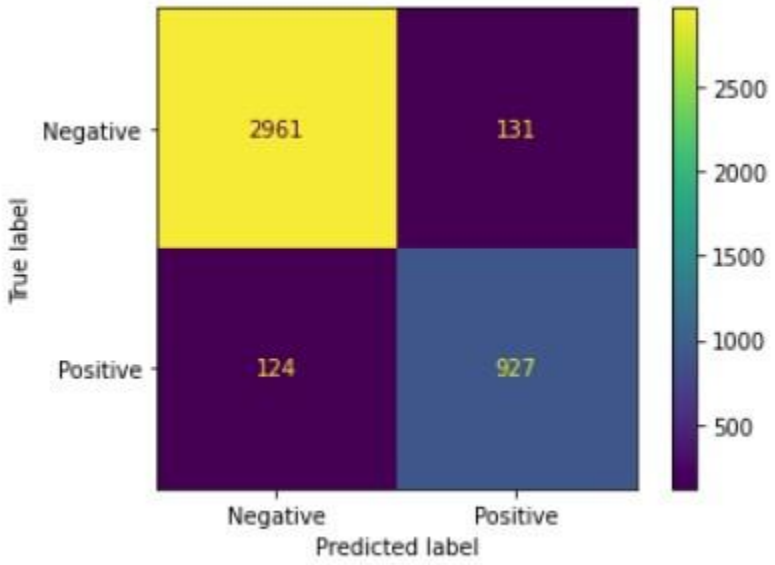




Original

Tune

Given that both models performed similarly, I chose to use the original model given that it contains less units making it a little bit more simple compared to the tuned model. Below is a confusion matrix of the validation set to see how well the model performs.



The confusion matrix shows that the model performs decently well in predicting the true label. We can calculate the precision, recall, and accuracy based on the confusion matrix

Precision	Recall	Accuracy
0.8762	0.8820	0.9385

It is arguably more important that recall is higher than precision since having a false positive diagnosis is less dangerous than a false negative diagnosis. Therefore it is good that the model has a slightly better recall compared to precision in this instance.

Conclusion

- On average there are subtle differences between the lung of a COVID positive patient compared to a COVID negative patient
- We were able to create a model with a 93.8% accuracy in predicting whether a patient is COVID positive

Future Improvements

- In the tuning step I was only able to try out around 5 models due to time. I would have loved to test out whether adding layers, including dropouts, etc. could have created a better model
- I could have potentially done something with the unbalanced data.