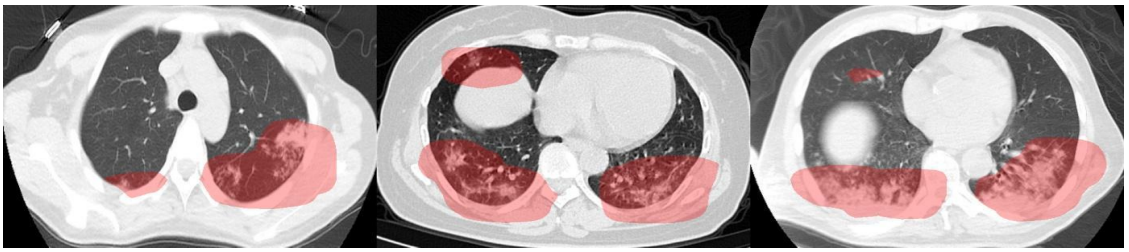
Applied ML Project Final Report
(Team 13)

COVID-19 Detection using Computer Vision, Machine Learning and Deep Learning on Chest CT Scan data

By- Anbang Wang, Jinyu Wang, Megan Zhou, Shubhangi Sengar, Zhengyi Chen

1. Project background

On March 11, 2020, the WHO declared COVID-19 as a global pandemic, and ever since its inception the disease has taken more than 4.5 million lives worldwide. In the fight against this novel disease, there is a pressing need for rapid and effective screening tools to identify patients infected with COVID-19, and to this end CT (Computed Tomography) imaging has been proposed as a key screening method which may be used as a complement to RT-PCR testing, particularly in situations where patients undergo routine CT scans for non-COVID-19 related reasons. Our team's aim is to detect the presence/ absence of COVID using chest CT images.



2. Exploratory data analysis

2.1 Data description

The dataset was taken from COVID CT data and consists of two variants: "A" and "B". The "A" variant consists of cases with confirmed diagnoses (i.e., RT-PCR confirmed). The "B" variant contains all of the "A" variant and adds some cases which are assumed to be correctly diagnosed but are weakly verified. "B" variant data is more extensive, so it is used for initial exploration. The data has 2 components i.e., metadata and image data, the details for both are as follows-

Metadata – This is a patient-level data for 4501 patients (3055 COVID-19 patients, 873 pneumonia patients, 573 normal individuals) and has 10 variables based on demographics & data collection. Few values are missing. The variables include Patient ID, Data source, Country, Age & sex, Finding (Normal, Pneumonia, or COVID-19), Verified finding (Yes or No), Slice selection (either Expert, Non-expert, or Automatic), View and modality (all are axial, CT respectively). More details about the variables have been covered in Deliverable 2 submitted before.

Image data – This is an image level data with 201,103 CT slices collected from 4501 patients in variant 'B' (we will see later that only the 194,922 CT slices of variant 'A' is chosen for model development).

2.2 Data visualization

Several plots were created using metadata to visualize label variable ('finding') in relation to demographic variables like 'country', 'source', 'sex', 'age', and the image data was also visualized for different classes. Detailed visualizations have been covered in deliverable 2 before, and a few noteworthy ones are included below-

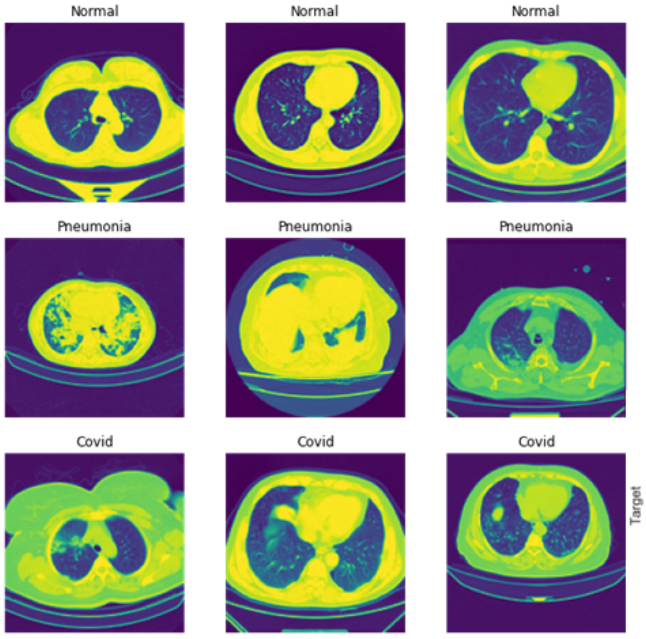


Fig 2a

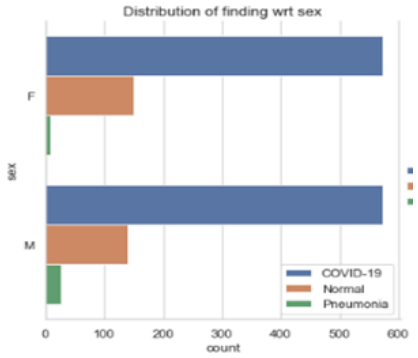


Fig 2b

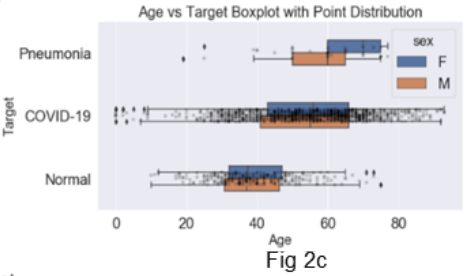


Fig 2c

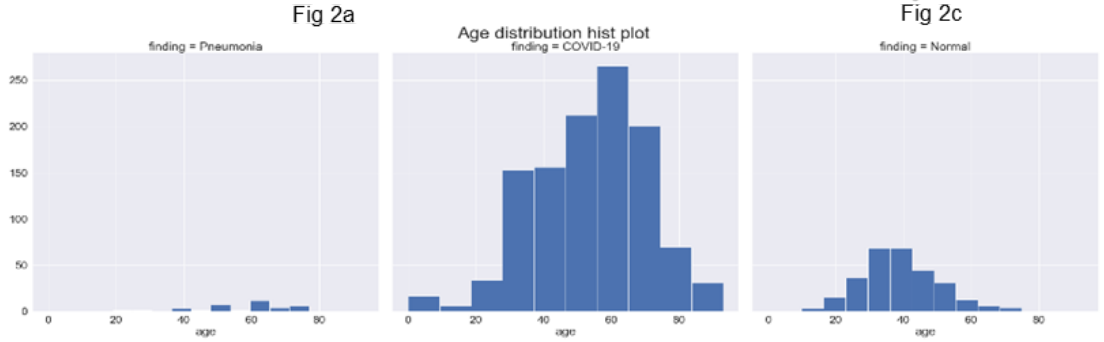


Fig 2d

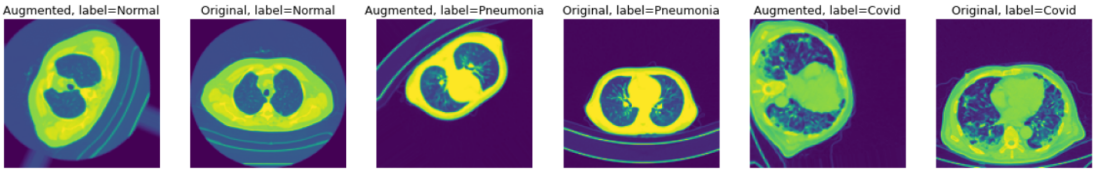
2.3 Insights from exploration

One of the first insights we found was that the size of image data (~28 GB) is too big for processing and collaborative model development and this motivates data resizing or sampling. Secondly, the data is imbalanced in terms of the label variable with a ratio of COVID:Normal:Pneumonia :: 7:2:1, and this inspires us to balance the data before moving with model development. Upon looking further, we realized that the additional observations (~17%) in variant 'B' dataset which do not come from variant 'A' dataset are not verified for COVID presence, as this is also indicted by a 'No' value for 'verified_finding' variable for such observations, implying that the label variable in for those data points can't be trusted completely. Moreover, the visualization of label variable in relation with demographic and data collection variables show that the label classes are almost equally distributed between male & female (*Fig 2b*). As a next step, we wanted to see the distribution of 'age' across different classes of label variable and upon doing so we found out that the distribution

is close to normal and the data has relatively low incidents for pneumonia (*Fig 2d*). A deeper look confirmed that there is a high concentration of covid cases within the age interval 50-70, implying that elders are more likely to be infected by covid, and that the 'age' distribution between the two 'sex' values is also very similar (*Fig 2c*).
The images are grayscale with a size of 512x512 pixels and have only one channel. Also, through visual inspection of x-ray images for each of the labels we can see that it is very hard for people who are not professional to distinguish between the three cases (*Fig 2a*) and, since the difference among categories is not huge, the project is challenging.

3. Data pre-processing

Inspired by our exploration, the non-verified data in variant 'B' was removed, leaving behind only the data from variant 'A' i.e., 194,922 CT slices of cases with confirmed COVID-19 diagnoses collected from 3745 patients. Out of these 94,548 images are for COVID-19, 60,083 are for normal and 40,291 are for pneumonia patients. Further, owing to the processing issues and in order to bring all the images to the same size, the raw image data that ranged between 256x256 to 1024x1024 pixels was resized to 128x128 pixels for each image and this exercise compressed the size of data from ~28 GB to ~2 GB.
After making the above adjustments, we realized that the processing issues are still persistent due to huge data size and also keeping in mind the imbalance in data, a random balanced sample was taken, resulting in 12,600 images which was then used for further pre-processing and model development.
As a next step for pre-processing, the data was split into train, val and test, resulting in 9000 images in train data, 1800 in val data and 1800 in test data. The label ratio was validated to be 1:1:1 for COVID:Normal:Pneumonia classes in each of these datasets.
Additionally, data augmentation for images was done as training deep learning neural network models on more data can result in more skilful models, and the augmentation techniques can create variations of the images that can improve the ability of the fitted models to generalize what they have learned to new images (refer below visualization to the effects of data augmentation).



4. Model development

Once we were done with data pre-processing and had our train, val and test datasets ready, we tried fitting different machine learning frameworks which included deep convolutional neural network, transfer learning using VCG16 & ResNet50 and other traditional models like Naive Bayes, Linear SVC and Random Forest to see which one performs the best. Also, we did an additional exercise of training each of the models (except traditional models) with and without data augmentation to find out whether the augmentation actually boosts performance in case of medical image dataset.

4.1 Deep Convolutional Neural Network

A deep CNN model was developed by adding iterations of Conv2D, MaxPooling2D layers, followed by Flatten and three Dense layers. Activation functions used were relu and softmax (for output layer). We trained two models, one with and another without data augmentation for 30 epochs. The optimized structures and optimized learning rate for the two models are found by using keras tuner. The below model summary presents the optimized model structure with data augmentation:

```
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_9 (Conv2D)           (None, 126, 126, 32)      896

 max_pooling2d_9 (MaxPooling  (None, 63, 63, 32)       0
 2D)

 conv2d_10 (Conv2D)          (None, 61, 61, 64)        18496

 max_pooling2d_10 (MaxPoolin  (None, 30, 30, 64)       0
 g2D)

 conv2d_11 (Conv2D)          (None, 28, 28, 128)       73856

 max_pooling2d_11 (MaxPoolin  (None, 14, 14, 128)      0
 g2D)

 flatten_3 (Flatten)         (None, 25088)             0

 dense_9 (Dense)             (None, 96)                2408544

 dense_10 (Dense)            (None, 64)                6208

 dense_11 (Dense)            (None, 3)                 195

=================================================================
Total params: 2,508,195
Trainable params: 2,508,195
Non-trainable params: 0
_________________________________________________________________
```

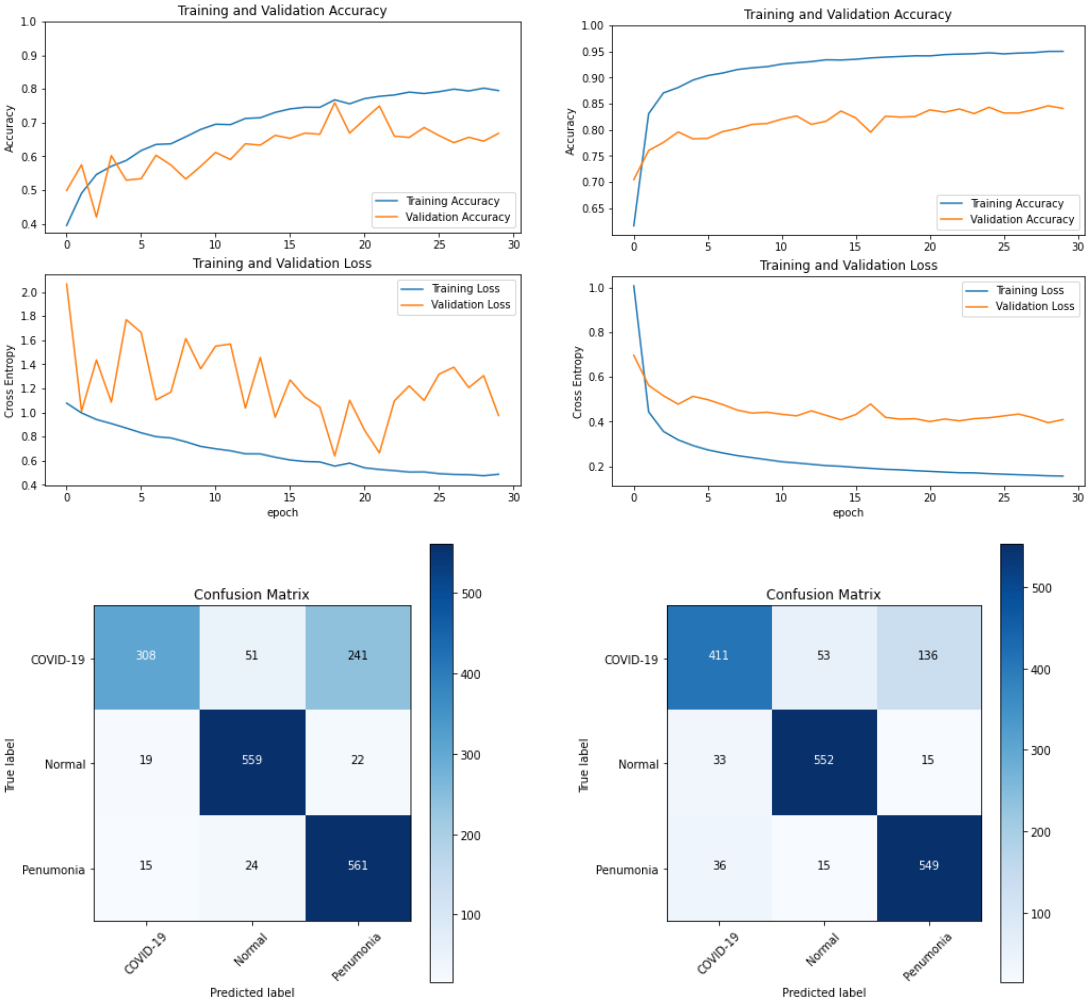
The evaluation on testing data (scores and confusion matrix) for the same are summarized below. Also, the accuracy and loss for training and validation with and without augmentation are plotted below.

With augmentation

CNN with augmentation				
	Precision	**Recall**	**F1-score**	**Accuracy**
COVID-19	0.9	0.51	0.65	
Pneumonia	0.68	0.93	0.79	
Normal	0.88	0.93	0.9	
Overall	0.82	0.79	0.78	0.79

Without augmentation

CNN without augmentation				
	Precision	**Recall**	**F1-score**	**Accuracy**
COVID-19	0.85	0.68	0.76	
Pneumonia	0.78	0.91	0.84	
Normal	0.89	0.92	0.9	
Overall	0.84	0.84	0.84	0.84



It can be noted from the above exercise that the CNN model performs better without image augmentation. It can also be seen that the model has started to overfit before 30 epochs in case of data augmentation, whereas no overfitting is seen for unaugmented data.

4.2 Transfer Learning

In order to improve the model performance further and try our hands on transfer learning, we explored frameworks like VGG16 and ResNet50 which are specialized for image classification. The details of model architecture and performance can be found below.

4.2.1 VGG16

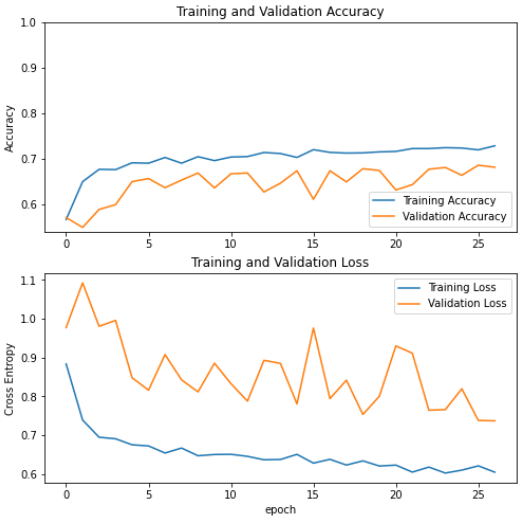
The first approach leveraged VGG16, which is one of the winning models in ILSVRC competitions. The model was created using keras VGG16 which was pre-trained on ImageNet dataset. The model uses a functional layer of VGG16, followed by the layers - GlobalAveragePooling2D, Flatten, a dense layer (with relu activation) and another dense output layer (with softmax activation). The same can be seen in the below model summary-

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 vgg16 (Functional)          (None, 4, 4, 512)         14714688

 global_average_pooling2d (G  (None, 512)              0
 lobalAveragePooling2D)

 dense (Dense)               (None, 64)                32832

 dense_1 (Dense)             (None, 3)                 195

=================================================================
Total params: 14,747,715
Trainable params: 33,027
Non-trainable params: 14,714,688
```

Using the above model architecture, training was done on both augmented and unaugmented data for 30 epochs and the model results are summarized below for both.

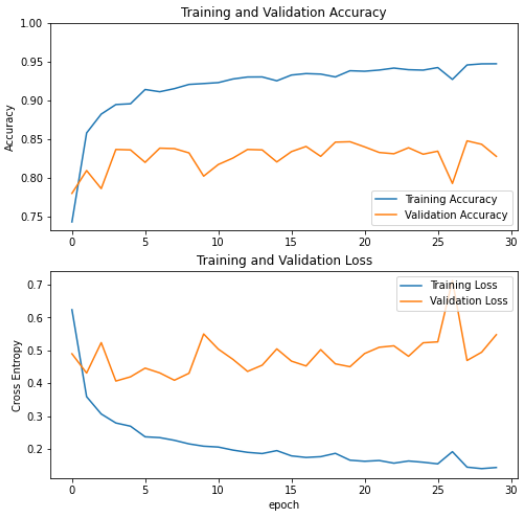
With augmentation

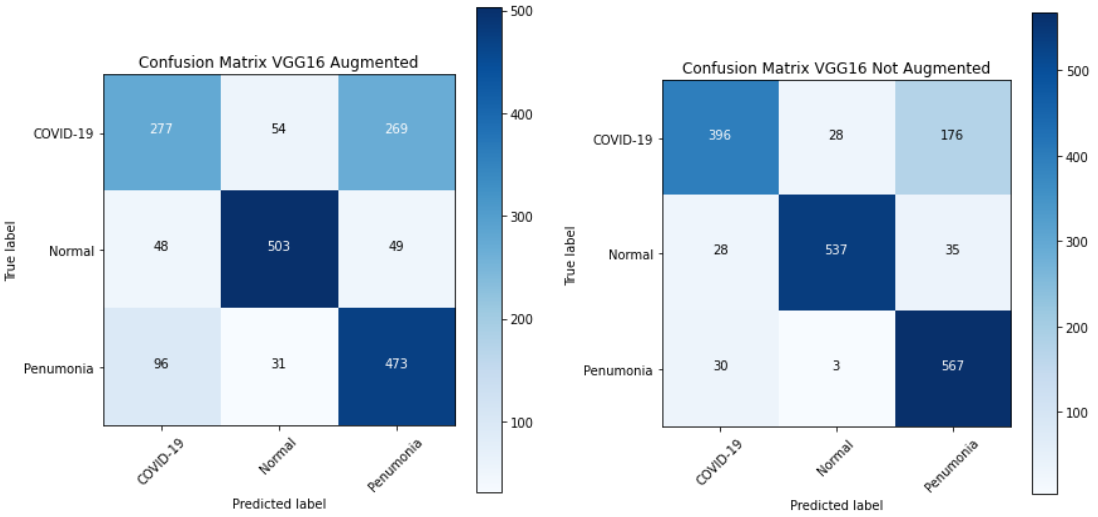
VGG with augmentation				
	Precision	**Recall**	**F1-score**	**Accuracy**
COVID-19	0.66	0.46	0.54	
Pneumonia	0.6	0.79	0.68	
Normal	0.85	0.84	0.85	
Overall	0.7	0.7	0.69	0.7



Without augmentation

VGG without augmentation				
	Precision	**Recall**	**F1-score**	**Accuracy**
COVID-19	0.87	0.66	0.75	
Pneumonia	0.73	0.94	0.82	
Normal	0.94	0.89	0.92	
Overall	0.85	0.83	0.83	0.83





It can be noted that accuracy and other classification metrics (precision, recall, f1-score) are better for unaugmented data. It can also be noted that training and validation performance is also better for unaugmented data.

4.2.2 ResNet50

Another approach that we used leverages ResNet50, which is a 50 layer deep CNN and well known for it's image classification capabilities. The model uses a functional layer of ResNet50 available on keras, followed by the layers GlobalAveragePooling2D, Flatten, a dense layer (with relu activation) and another dense output layer (with softmax activation). The same can be seen in the below model summary-

```
 Layer (type)                 Output Shape              Param #
=================================================================
 resnet50 (Functional)        (None, 4, 4, 2048)        23587712

 global_average_pooling2d_1   (None, 2048)              0
 (GlobalAveragePooling2D)

 dense_2 (Dense)              (None, 64)                131136

 dense_3 (Dense)              (None, 3)                 195

=================================================================
Total params: 23,719,043
Trainable params: 23,665,923
Non-trainable params: 53,120
```

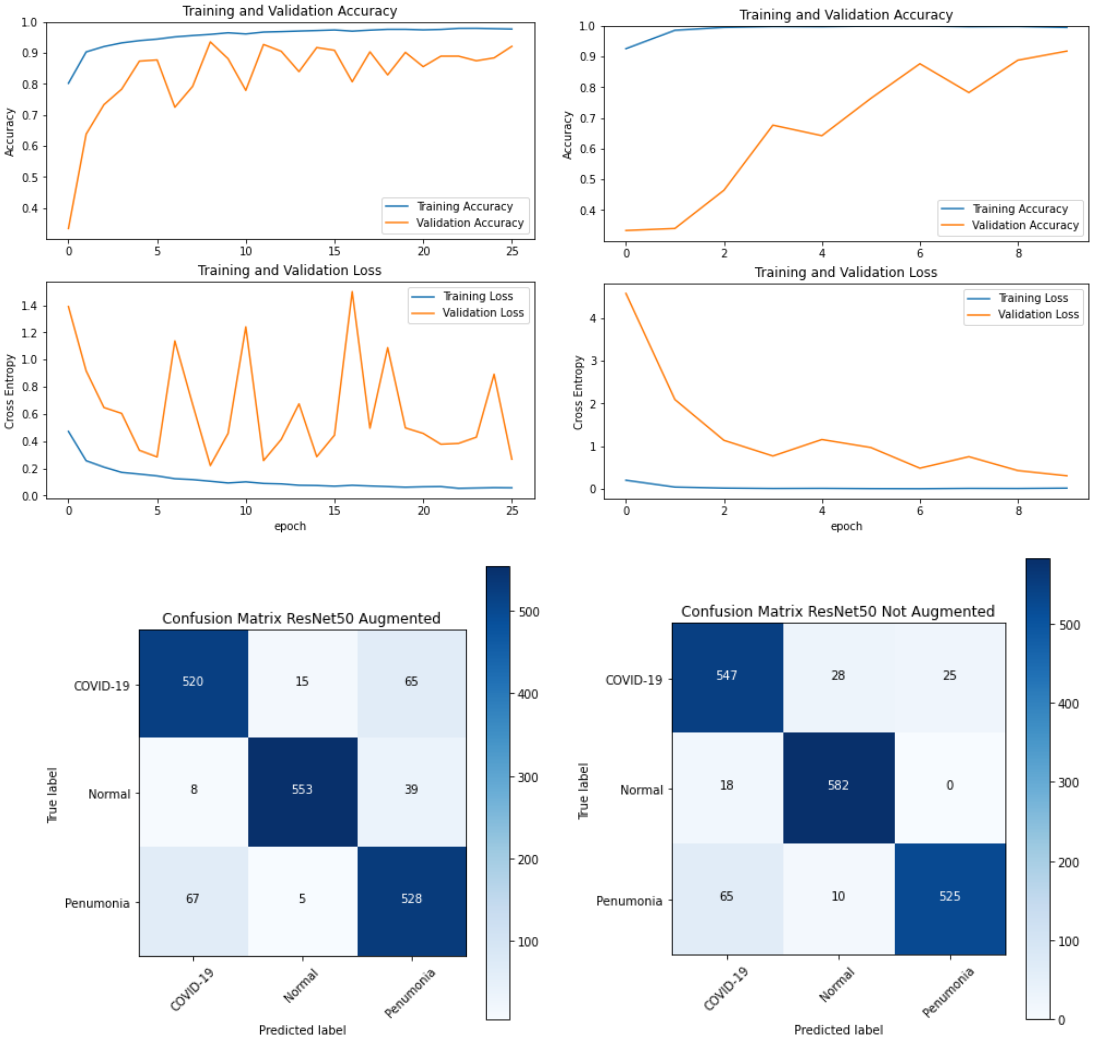
Using the above model architecture, training was done on both augmented and unaugmented data for 30 epochs and model results are summarized below.

With augmentation

Resnet with augmentation				
	Precision	**Recall**	**F1-score**	**Accuracy**
COVID-19	0.87	0.87	0.87	
Pneumonia	0.83	0.88	0.86	
Normal	0.96	0.92	0.94	
Overall	0.89	0.9	0.89	0.89

Without augmentation

Resnet without augmentation				
	Precision	**Recall**	**F1-score**	**Accuracy**
COVID-19	0.87	0.91	0.89	
Pneumonia	0.95	0.87	0.91	
Normal	0.94	0.97	0.95	
Overall	0.92	0.92	0.92	0.92



It can be noted that accuracy and other classification metrics (precision, recall, f1-score) are marginally better for unaugmented data. It can also be noted that training and validation performance is also better for unaugmented data.

4.3 Other Machine Learning Models

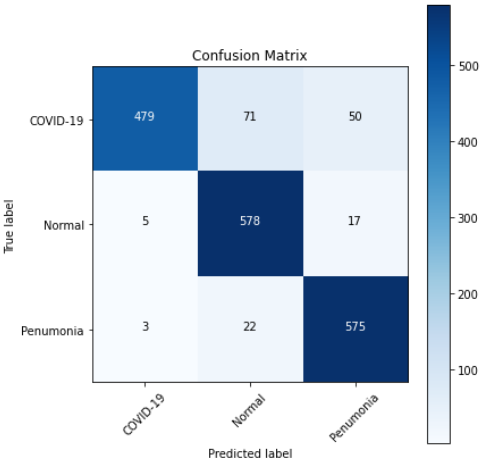
In addition to deep learning and transfer learning, traditional machine learning models like Naive Bayes, Linear SVC and Random Forest Classifier were also applied for classification. As we saw in the case of deep CNN and transfer learning that augmentation didn't improve the accuracy, so we haven't used augmented data to train these models. Instead, the data was fed into the models in two different forms - the raw data and the extracted 3D color histogram from the HSV color space. Below is a summary of performance for each model-

Performance of traditional ML models		
Model	**Raw pixel accuracy**	**Histogram accuracy**
Naïve Bayes	0.47	0.62
Linear SVC	0.58	0.59
Random Forest	0.9	0.81

As can be seen from the above table, Random Forest Classifier performs much better compared to Naive Bayes and Linear SVC for both forms of data. It can also be noted that for Naive Bayes and Linear SVC, the accuracy is better for extracted 3D color histogram data, whereas for Random forest, the accuracy is better with the raw data. In this case, only Random Forest is consistent with our Deep Learning model in the way that we achieve scores with the unprocessed data.

Since, Random Forest Classifier with raw image data was a clear winner, we have visualized the results more elaborately for the same below-

Random Forest Classifier				
	Precision	**Recall**	**F1-score**	**Accuracy**
COVID-19	0.98	0.8	0.88	
Pneumonia	0.9	0.96	0.93	
Normal	0.86	0.96	0.91	
Overall	0.91	0.91	0.91	0.9



5. Conclusion

Below is a summary of the model performance for all the different models that we fit for our use case.

Comparison of different models		
Framework	**Model**	**Accuracy**
Deep Neural Network	CNN with augmentation	79%
Deep Neural Network	CNN w/o augmentation	84%
Transfer Learning	VGG16 with augmentation	70%
Transfer Learning	VGG16 w/o augmentation	83%
Transfer Learning	ResNet50 with augmentation	89%
Transfer Learning	ResNet50 w/o augmentation	92%
Traditional ML model	Naïve Bayes with raw pixel data	47%
Traditional ML model	Naïve Bayes with color histogram data	62%
Traditional ML model	Linear SVC with raw pixel data	58%
Traditional ML model	Linear SVC with color histogram data	59%
Traditional ML model	Random Forest with raw pixel data	90%
Traditional ML model	Random Forest with color histogram data	81%

It can be noted that ResNet50 without augmentation performs the best, followed by Random Forest Classifier on raw pixel data and ResNet50 without augmentation. This

points to the excellent image classification performance of ResNet50, and the generalization capability of Random Forest. It should also be noted that the models performed better on unaugmented data than on augmented data, and hence, we can say that in case of medical image data, image augmentation is not so helpful.

6. Limitations

Known limitation of the models is the size of data. Since the image data was huge, a data sample of 9000 images for training, 1800 images for validation and 1800 images for testing had to be taken to facilitate model processing. This limits our ability to use more data to polish our results and uplift the accuracy.

7. Next Steps

We suggest using GPUs in order to expand the use of data to train the models. This is likely to bump up the accuracy of the models and remove the need for under sampling. Also, if we can get text data for symptoms in addition to chest CT scan images, we can possibly combine text and image analytics to bump up the accuracy. Moreover, we can try adding demographic variables like 'age' which have different distributions for different label classes as this might help increase the accuracy of the model. Lastly, we suggest that since in the medical world the false negatives may cost lives, so we can try to tune our threshold to minimize the false negatives.