Zumari Chatham
Niket Pandya
John K. Kimutai
James Okullo
Youa Vang

Covid-19 CT Scan Prediction with Deep Learning

Introduction

COVID-19, from coronavirus disease 2019, is a contagious disease caused by viruses in the coronavirus subfamily. These viruses cause respiratory tract infections, which can range from mild to lethal. Some of the mild illnesses associated with human beings are the common cold (which can be caused by other viruses like rhinoviruses). More severe illnesses include Severe Acute Respiratory Syndrome (SARS), Middle East Respiratory Syndrome (MERS), and SARS-Cov-2. These viruses are known for their crown shape.

The first case of COVID-19 was discovered in Wuhan, China in December 2019. According to the CDC, the US's first patient identified with a confirmed COVID-19 virus was in Washington State on January 21, 2020. Although the US outbreak appeared to be contained in February, it rapidly accelerated in the following months. Since the first discovery in the United States, the CDC tracker shows that the US is the leading country with over 17 million confirmed positive cases. The total number of reported cases is greater than 330,000. Some of the factors that contributed to the accelerated spread, especially during the months of February-March 2020 and at the end of summer and the beginning of fall, include continued travel-associated importations, larger gatherings, introductions into high-risk workplaces and densely populated areas, and cryptic transmission resulting from limited testing and asymptomatic and pre-symptomatic spread.

The COVID-19 virus causes viral pneumonia in the lungs, which results in acute respiratory syndrome. The virus affects different people in different ways. The symptoms which may appear in 2-14 days may include fever or chills, cough, shortness of breath, fatigue, muscle or body aches, headache, new loss of test or smell, sore throat, congestion or runny nose, nausea or vomiting, and diarrhea.

There are two types of tests: diagnostic tests and antibody tests. The diagnostic test shows if the patient has an active coronavirus infection and therefore needs to isolate or quarantine. Diagnostic tests include molecular (RT-PCR) tests that detect the virus' genetic material and antigen tests that detect specific proteins on the virus's surface. On the other hand, an antibody test looks for the antibodies created by the immune system once someone has been infected with the virus. These antibodies take several weeks to develop after infection and also stays for several weeks after recovery.

Medical imaging devices are being used in most treatment centers by researchers who analyze the CT scans and X-rays to detect COVID-19. Most patients with COVID-19 showed infections in their lungs, and this can be used for an early diagnose of the virus but has not been recommended for definitive diagnosis. We, therefore, decided to come up with an optimal Convolution neural network model that will be able to identify CT-scans for patients with Covid-19.

Dataset

We obtained our dataset from Kaggle and it is stored in a shared drive which can be accessed using the link below:

https://drive.google.com/drive/folders/1xdk-mCkxCDNwsMAk2SGv203rY1mrbnPB?usp=sharing

The CT scans in our dataset were collected in March and April from Negin radiology, a medical center located in Iran. This medical center uses SOMATOM Scope model and syngo CT VC30-easyIQ software to capture and visualize the lung HRCT radiology imaged from the patients. The format of the radiology images was 16-bit grayscale DICOM format with 512*512 pixels resolution. The images were not converted to 8bit data since it might cause losing some data. Instead, the images were converted to TIFF format, 16 grayscale images. In this format, the images cannot be viewed with standard monitors as they will appear black. However, they can be visualized using Visual.py to convert the dataset images to a visualizable format. The dataset consists of 15,589 CT scan images captured from 95 patients infected with COVID-19 and 48,460 CT scan images of 282 of patients who were not infected with COVID-19.

For our experiment, we are using the COVID-CTset. The image dataset consists of two sections: training and validation sets for training and testing the neural network and a set of raw data. To emulate real and accurate results, the dataset was separated into five folds with 20% of the patients with COVID-19 allocated for testing the model in each fold. The number of images for normal patients was higher than the number of images for infected patients. Therefore, we chose only a subset of the normal images, similar to the number of COVID-19 images, to have a more balanced testing and validation dataset. This resulted in a higher number of images for normal patients considered for network testing compared to the training images. The table below lists the number of images and patients.

COVID-19 Patients	Normal Patients	COVID-19 Images	Normal Images
95	282	15,589	48,260

Table 1 Patient and Image Numbers

Each patient has three folders (SR_2, SR_3, SR_4), with each folder show a sequence of the lung HRCT scan images of that patient when the lungs open and closes. These folders have CT scans for the same patient with different recorded thickness.

The data was separated into Train, Validation, and Test sets, shown below.

Train Set				
Fold	**COVID-19 Patients**	**COVID-19 Images**	**Normal Patients**	**Normal Images**
1	77	1820	45	1916
2	72	1817	37	1898
3	77	1836	53	1893
4	81	1823	76	1920
5	73	1832	71	1991

Table 2 Train Set

Validation Set				
Fold	**COVID-19 Patients**	**COVID-19 Images**	**Normal Patients**	**Normal Images**
1	18	462	22	450
2	23	465	22	450
3	18	446	22	450
4	14	459	22	450
5	22	450	22	450

Table 3 Validation Set

Test Set				
Fold	**COVID-19 Patients**	**COVID-19 Images**	**Normal Patients**	**Normal Images**
1	18	462	237	7860
2	23	465	245	7878
3	18	446	229	7883
4	14	459	206	7856
5	22	450	211	7785

Table 4 Test Set

The test set is raw data and is unbalanced. We chose not to balance the test set, so that it mimics real world environment. Furthermore, we wanted to see how each model would perform under this condition.

LSTM Model

One of our goals for this project was to create our own model based off the most accurate pretrained model and see if our model could be better. We decided to combine the CNN model we used for pretraining and combine it with LSTM. CT Scans are taken from the front, back, and side to side and are done in layers. All the patient in our data set have a multiple CT Scans regardless of whether they are positive or negative for COVID. Although, CT scans are a series of images, the images are taken in one sitting and therefore only capture different views of the chest, but do not show progression of the disease over time. We therefore did not use the sequence relationship of the images as a feature of importance in our model.

LSTM is great for classifying, processing, and making predictions on information that is important and passing relevant information forward in a sequence. While CNN is great for feature extraction. From the research we have conducted LSTM is primarily used for classifying and processing time series data whether its images, videos, and stock market data. The challenge was how will we make both models function together.

In order to address this challenge, we used functional API architecture and concatenate them together, which allowed us to extract features from the CNN and pass on that information to the LSTM for sequence processing. The other challenge we faced was passing the correct images from one model to the next. We were able to accomplish that by reshaping the dimensions of the CT Scans by averaging out the pixel size across 1 RGB layer using a softmax activation.

Below is the architecture of the ResNet152-LSTM where the CNN is concatenated with the LSTM

```
conv5_block3_add (Add) (None, 16, 16, 2048) 0 conv5_block2_out[0][0] conv5_block3_3_bn[0][0]
lambda_4 (Lambda) (None, 512, 512) 0 input_1[0][0]
conv5_block3_out (Activation) (None, 16, 16, 2048) 0 conv5_block3_add[0][0]
reshape_5 (Reshape) (None, 512, 512) 0 lambda_4[0][0]
average_pooling2d (AveragePooli (None, 8, 8, 2048) 0 conv5_block3_out[0][0]
lstm_4 (LSTM) (None, 512, 2048) 20979712 reshape_5[0][0]
flatten_layer (Flatten) (None, 131072) 0 average_pooling2d[0][0]
lstm_5 (LSTM) (None, 2048) 33562624 lstm_4[0][0]
concatenate (Concatenate) (None, 133120) 0 flatten_layer[0][0] lstm_5[0][0]
dense (Dense) (None, 2) 266242 concatenate[0][0]
==================================================================================
Total params: 113,173,250 Trainable params: 113,021,826 Non-trainable params: 151,424
```

Resnet152=grey LSTM=blue Merged=red

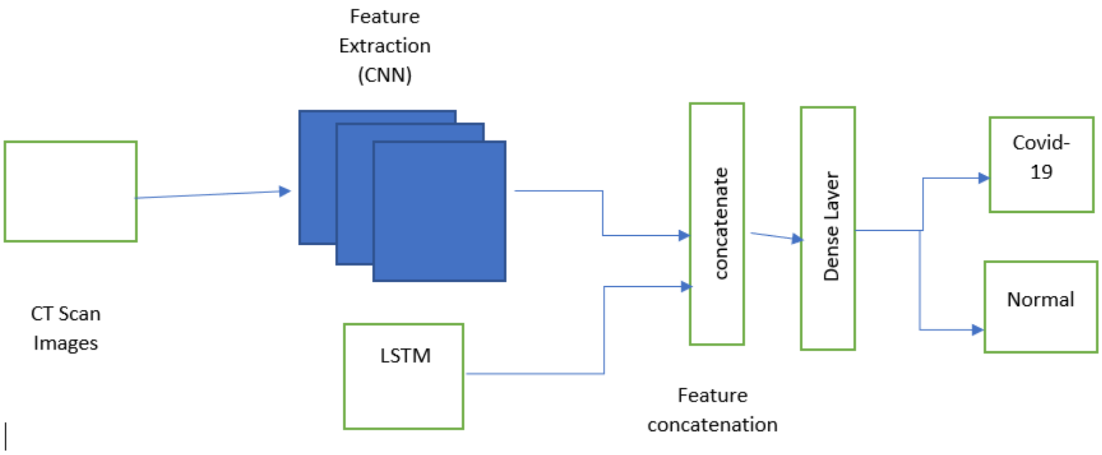


Figure 1: Visual of the ResNet152-LSTM model

Model Comparison

We used the same parameters in all the models listed in the following table (Table 5). To determine our champion model, we looked at the average accuracies of each of the CNN transfer learning models. The average accuracies were calculated from five folders of training and testing data. See Table 6.

Training Parameters	**Value**
Learning Rate	1.00E-04
Batch Size	10
Optimizer	Nadam
Loss Function	Categorical Crossentropy
Epochs	100
Steps Per Epochs	100
Horizontal/Vertical flipping	Yes
Zoom Range	5%
Rotation Range	0 - 360 degree
Width / Height shifting	5%
Shift Range	5%

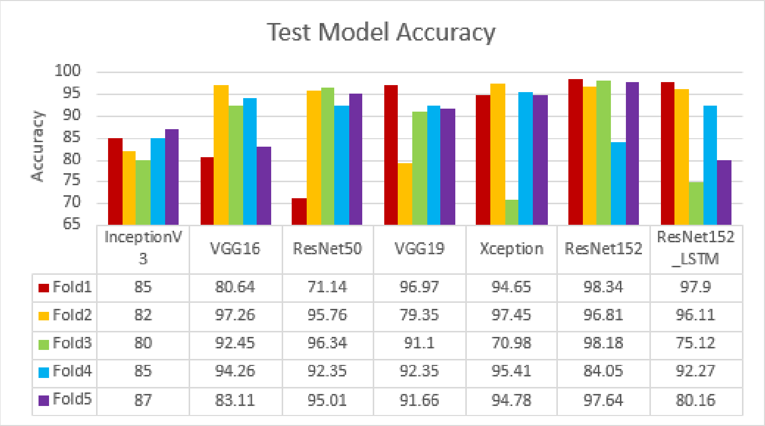
Table 5 Training Parameters

All the models performed about the same on the training data, varying between 96.7-98.8%. The Xception model had the best accuracy on the training data at 98.8%, in red, but the Xception fold3 testing data had the lowest test accuracy of 71%, also in red. For this reason, we did not use the Xception model as our champion model but instead used the ResNet152 model. It had the highest testing accuracy of 95%, highlighted in yellow. We then created our own model by following RestNet152 transfer learning CNN model with a LSTM model. As you can see the ResNet152_LSTM model's accuracy was not better than the champion model.

	Fold1		**Fold2**		**Fold3**		**Fold4**		**Fold5**		Ave Train	Ave Test
Model	**train**	**test**	**train**	**test**	**train**	**test**	**train**	**test**	**train**	**test**		
InceptionV3	98.7	85.0	99.4	82.0	98.5	80.0	97.5	85.0	97.6	87.0	98.4	83.8
VGG16	98.6	80.6	98.5	97.3	99.0	92.5	98.9	94.3	96.7	83.1	98.3	89.5
ResNet50	99.0	71.1	98.6	95.8	99.2	96.3	97.2	92.4	98.2	95.0	98.5	90.1
VGG19	98.7	97.0	97.4	79.4	98.4	91.1	97.2	92.4	97.4	91.7	97.8	90.3
Xception	99.7	94.7	97.5	97.5	97.7	71.0	99.3	95.4	99.6	94.8	98.8	90.7
ResNet152	94.8	98.3	99.0	96.8	98.4	98.2	96.1	84.1	96.1	97.6	96.9	95.0
ResNet152_LSTM	96.1	97.9	95.2	96.1	97.7	75.1	98.2	92.3	96.4	80.2	96.7	88.3

Table 6 Fold Accuracy

Figure 2. shows the test model accuracy for each fold, of each model. Figure 3. shows the average test accuracy of each model. As you can see, the champion model has the best overall test accuracy of all the models. This includes our model, ResNet152_LSTM. The accuracy from the champion model jumps down from 95% to 88.3% in our model, underperforming the champion model.



	InceptionV3	VGG16	ResNet50	VGG19	Xception	ResNet152	ResNet152_LSTM
🟥 Fold1	85	80.64	71.14	96.97	94.65	98.34	97.9
🟨 Fold2	82	97.26	95.76	79.35	97.45	96.81	96.11
🟩 Fold3	80	92.45	96.34	91.1	70.98	98.18	75.12
🟦 Fold4	85	94.26	92.35	92.35	95.41	84.05	92.27
🟪 Fold5	87	83.11	95.01	91.66	94.78	97.64	80.16

Figure 2 Test Model Accuracy

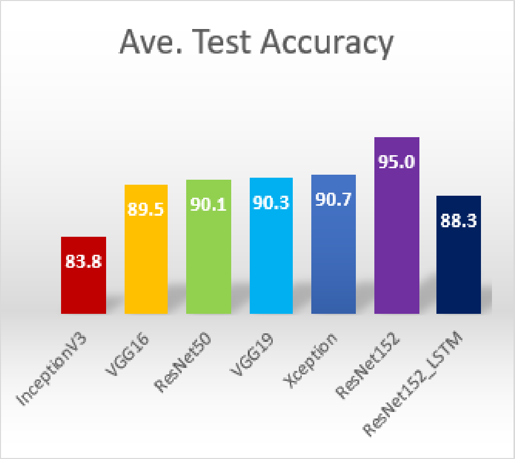


Figure 3: Average. Test Accuracy

Figure 4. displays the average training time each model took to run. This time does not include the time it took to run the entire models. The complete models ran at least 2X longer than the training time. Using training time to pick the champion model was an afterthought. Adding the LSTM model to the champion

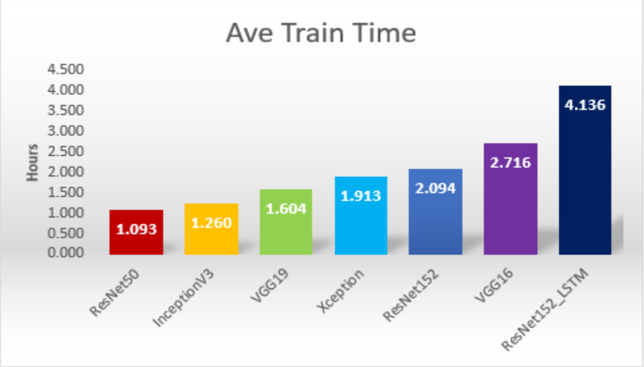


Figure 4 Average Train time

model increased the training time on average from 2.1 to 4.1 hours. Having considered the increase in time of running our model we may have considered choosing another champion model.

We then evaluated the trained networks using four different metrics for each of the classes and the overall accuracy for all the classes as follows:

$$Accuracy\ (for\ each\ class) = \frac{TP + TN}{TP + FP + TN + FN} \tag{1}$$

$$Specificity = \frac{TN}{TN + FP} \tag{2}$$

$$sensitivity = \frac{TP}{TP + FN} \tag{3}$$

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$Accuracy\ (for\ all\ the\ classes) = \frac{Number\ of\ Correct\ Classified\ Images}{Number\ of\ All\ Images} \tag{5}$$

The Average Evaluation results for each network are described in following 2 tables. These results were hand calculated from the confusion matrix that was generated from each model. The RestNet152 model still is the best model based on the average evaluation results of each network. It had the best overall accuracy of 95% and COVID-19 sensitivity of 52.5%. It correctly identified the greatest number of correct patient IDs. The Resnet50 model had the best sensitivity for normal patient IDs and predicted the most correct COVID-19 IDs. but it still did not outperform our champion model. It only had slightly higher numbers in terms of correct COVID-19 IDs. It mistakenly identifies more normal images as COVID-19. Our model, ResNet152_LSTM correctly identified, on average, one more correct COVID-19 ID but still predicted less normal correct IDs and had worse accuracy than our champion model.

Fold	**Network**	**Overall Accuracy**	**COVID Sensitivity**	**Normal Sensitivity**	**COVID Specificity**	**Normal Specificity**	**COVID Precision**	**Normal Precision**
Average	ResNet50	90.11%	35.53%	99.88%	99.88%	35.53%	98.20%	89.64%
	VGG19	81.80%	5.71%	94.54%	94.54%	5.71%	14.90%	85.69%
	Xception	90.64%	31.75%	99.84%	99.84%	31.75%	96.96%	90.22%
	InceptionV3	83.91%	5.04%	94.45%	94.45%	5.04%	10.82%	88.15%
	VGG16	80.57%	5.41%	94.49%	94.49%	5.41%	15.38%	84.36%
	ResNet152	95.00%	52.52%	99.65%	99.65%	52.52%	94.22%	95.05%
	ResNet152_ LSTM	88.34%	31.36%	99.64%	99.64%	31.36%	94.52%	87.98%

Table 7 Average Evaluation Results for Each Network Part 1

Fold	Network	Correct patients ID	Wrong patient ID	Covid Correct ID (TP)	Wrong ID as Normal (FN)	Normal Correct ID (TN)	Wrong ID as Covid (FP)
Average	ResNet50	7487	822	448	813	7039	8
	VGG19	6797	1512	68	1123	6729	388
	Xception	7531	777	357	768	7086	11
	InceptionV3	6972	1337	49	930	6922	407
	VGG16	6694	1615	70	1228	6624	386
	ResNet152	7894	415	430	389	7464	26
	ResNet152_LSTM	7340	969	431	944	6908	25

Table 8 Average Evaluation Results for Each Network Part 2

Comparing average results in previous two tables to the average results generated in what was called the FPN (Folds per network) table, in the following two tables, we noticed that our results were different. We were unable to correctly identify the cause of the discrepancy, but we noticed that the conclusion remained the same. The Resnet152 model was still the best model.

Fold	Network	overall_acc	csens	nsens	cspec	nspec	cprec	nprec
Average	ResNet50	90.72%	85.07%	91.05%	91.05%	85.07%	50.81%	99.07%
	VGG19	79.36%	79.84%	79.33%	79.33%	79.84%	28.94%	98.41%
	Xception	86.94%	84.64%	87.07%	87.07%	84.64%	41.76%	99.00%
	InceptionV3	83.90%	82.30%	83.99%	83.99%	82.30%	36.67%	99.03%
	VGG16	78.78%	84.30%	78.46%	78.46%	84.30%	30.01%	98.67%
	ResNet152	91.50%	87.48%	91.73%	91.73%	87.48%	57.62%	99.23%
	ResNet152_LSTM	89.75%	87.51%	89.88%	89.88%	87.51%	45.07%	99.16%

Table 9 FPN Part 1

Fold	Network	tp	fp	ctp	cfn	cfp	ntp	nfn	nfp
Average	ResNet50	7542	772	389	68	704	7153	704	68
	VGG19	6591	1714	364	92	1622	6227	1622	92
	Xception	7222	1086	386	70	1016	6836	1016	70
	InceptionV3	6974	1338	377	81	1257	6597	1257	81
	VGG16	6551	1762	386	72	1690	6165	1690	72
	ResNet152	7604	706	400	57	648	7204	648	57
	ResNet152_LSTM	7457	851	400	57	794	7057	794	57

Table 10 FPN Part 2

The Confusion Matrices are listed below for the champion model and the ResNet152_LSTM model.

Total	**Resnet152 Average**			
456.4	436.4	20	Covid-19	Ground
7852.2	644.6	7207.6	Normal	Truth
8308.6	Covid-19	Normal		Labels

Predicted Labels

Table 11 ResNet152 Confusion matrix

Total	**Resnet152-LSTM Average**			
456.4	431.4	25	Covid-19	Ground
7852.4	944.2	6908.2	Normal	Truth
8308.8	Covid-19	Normal		Labels

Predicted Labels

Table 12 ResNet152-LSTM Confusion matrix

The ROC curves, shown below, of the true positive and true negative rate of the combined classes of the test data for the champion model and the Resnet152_LSTM model. We realized that we should have separated the classes but ran out of time. The champion model has an overall better ROC average of 0.9462. Our model has an average ROC average of 0.915. The champion model continues to beat our model.

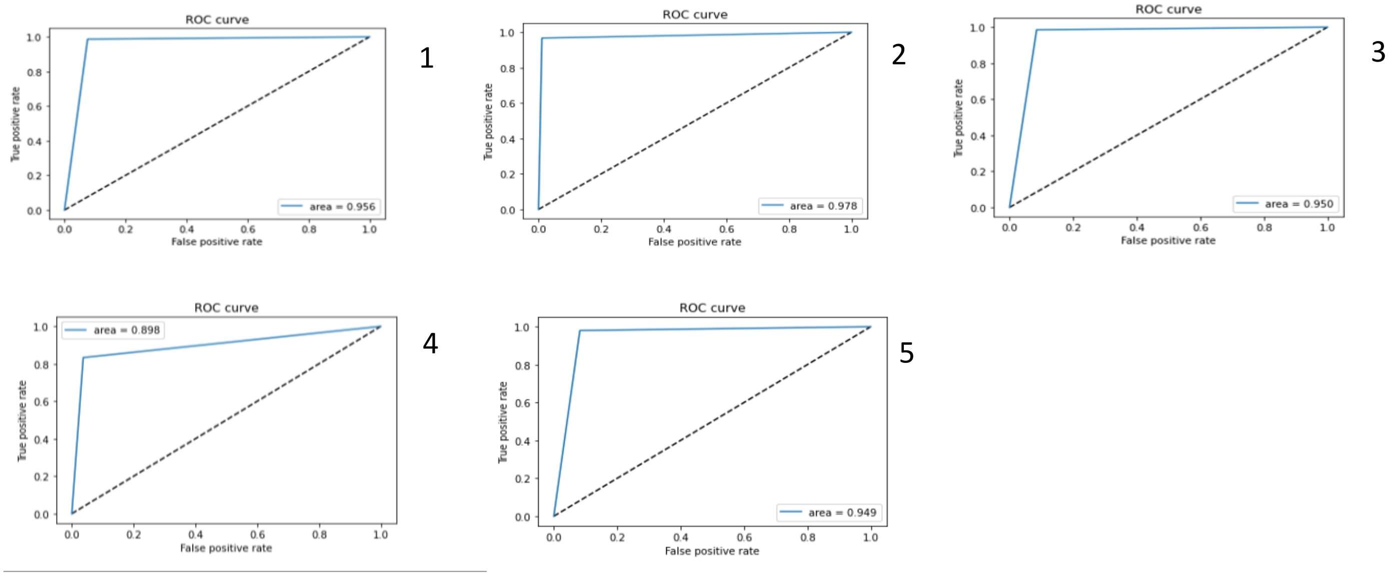


Figure 5 ResNet152 ROC Folds 1-5, Average 0.9462

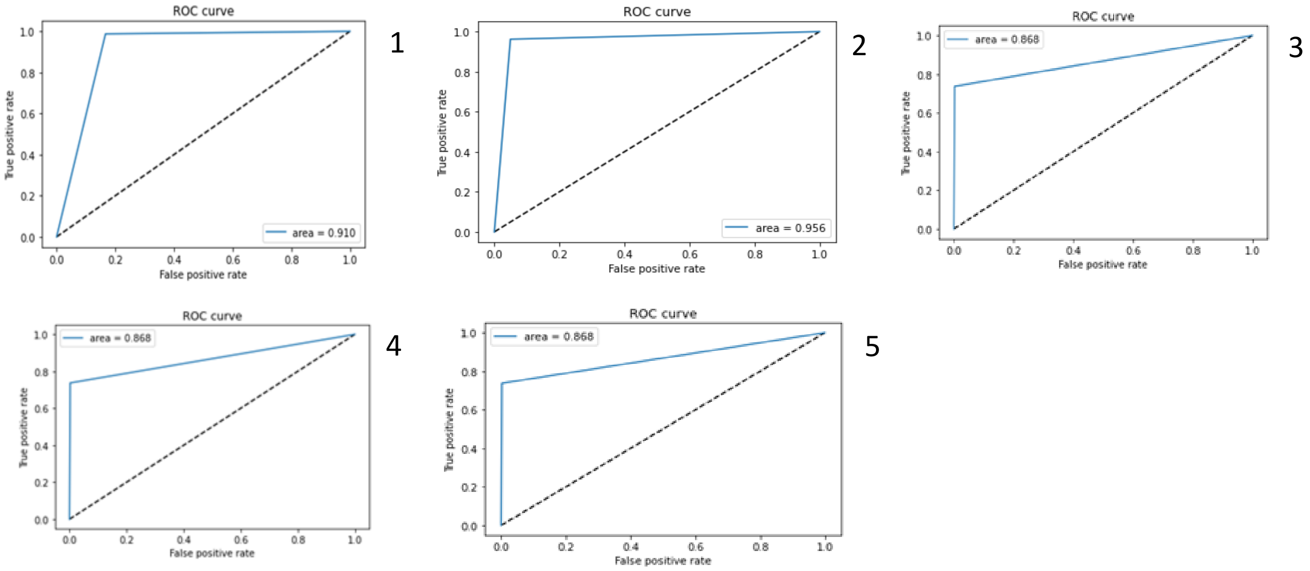


Figure 6 ResNet152-LSTM ROC Folds 1 -5. Average 0.914

Tf-Explain/Grad CAM

COVID 19 infects several organs within the body, causing severe inflammation. In particular, the lungs' infection results in fluid build up in the tissues, similar to other pneumonia cases. However, unlike bacterial-pneumonia, viral pneumonia causes lesions in the lungs . In CT-Scans images, which are significantly more detailed than x-rays, these lesions look opaque. These patchy white structures only obscure underlying structures but do not entirely hide them. They look like ground-glass and are thus commonly referred to as Ground Glass Opacities (GGO). Several studies have shown that a majority of patients with COVID-19 pneumonia have GGOs. In fact, cases have been documented where the patient has a false negative RT-PCR test (the standard test), but with chest CTs showing progression of the disease. The shape, location, and spread of GGO's also help to distinguish COVID- 19 from other conditions like lung cancer, heart disease, and other parasitic, autoimmune, fungal, and mycobacterial diseases.

A normal lung CT scan appears mostly black (Fig 7 B). The white translucent patches (GGO's) in the lungs are indicative of COVID-19 pneumonia (Fig 7 A). The models are able to correctly detect lesions on the lungs of COVID-19 positive patients.

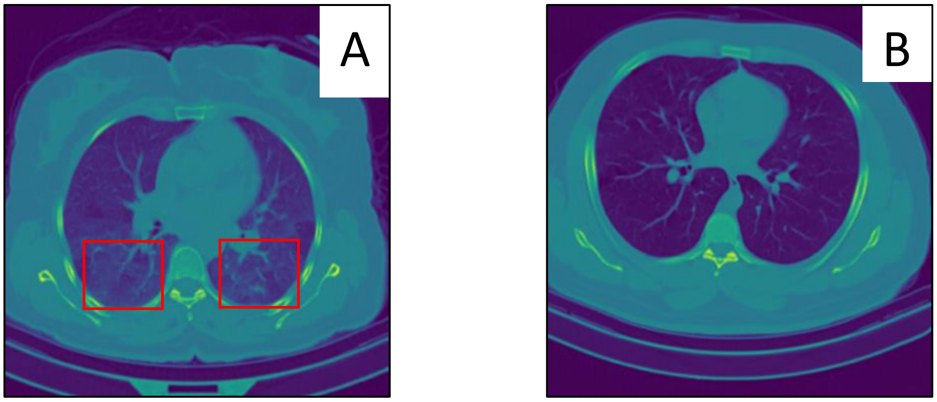


Figure 7: COVID-19 Positive lung CT image. B. Normal lung CT image

However, one of the criticisms most often raised about NNs is that they are 'black-boxes.' Therefore, a big challenge with these models is model interpretability: "how are the "guts" of the model functioning, and are they functioning correctly?".

Tf-explain is a framework that offers several methods to interpret and explain the behavior of a network. It therefore can serve several purposes: visualizing and debugging the model; build trust in prediction of the model; and detection of otherwise unknown important features. The methods offered in Tf-explain APIs are Activation Visualization, Vanilla Gradients, Gradient Inputs, Grad CAM, Occlusion Sensitivity, Smooth Grad, and Integrated Gradients. For our project, we focused on utilizing Grad CAM to help us understand how our model analyzes the images and to see where COVID-19 is localized in patients who are COVID-19 positive.

Gradient-weighted Class Activation Mapping (Grad CAM) uses the gradients in the final convulsion layer to produce a localization map that highlights important features the model is using for prediction. Grad CAM is commonly used for detecting issues with object detection, classification and other computer vision tasks. Grad CAM can be implemented at different layers of a model to analyze what the model sees at each layer and during training by utilizing GradCAMCallbacks.

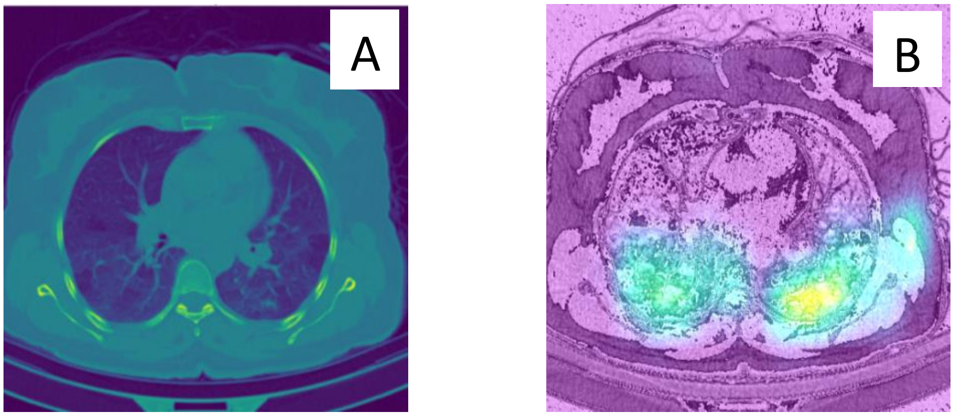


Figure 8: COVID-19 positive image. B. GradCAM image of COVID-19 positive

GradCAM helps to visualize what areas or features the model uses to classify an image as either positive or normal. Using the gradients of specific classes flowing into the last convolution layer, Grad-CAM creates a heatmap that shows the regions of the image that are important for correct classification (Figure 8).

A further investigation of the Grad-CAM heatmaps shows how the model distinguishes between a positive COVID-19 image and a normal chest CT. While the positive images (Fig 9 A,B) show concentrated activations in the part of the lungs with GGOs, there is no such activation in the normal lungs (Fig 9 C,D).

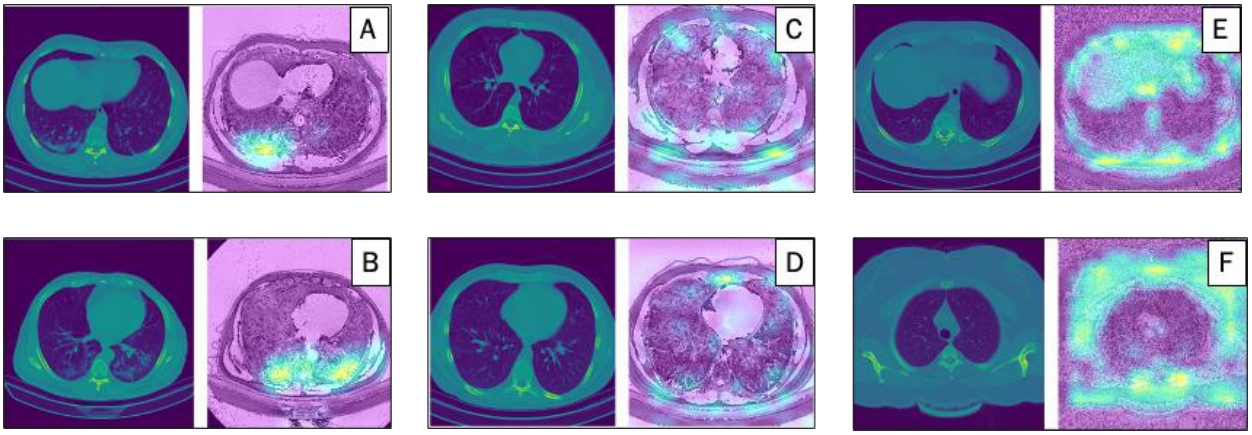


Figure 9: A,B COVID-19 Positive. C,D True Negative. E-F False positives

In a few instances, the heatmaps of normal scans showed activations across the entire image, not just the infected area. Similarly, images wrongly classified as positive (false positives) showed activations in hazy areas of the images, but not necessarily in the lungs (Fig 9 E,F). This suggests that the model has learned, particularly well, to look for the features similar to GGOs; however, it needs to be fine-tuned to look for these patterns only within the lungs. The false positives also appear to be prominent in scans in which the other organs are prominent, or the lungs are partially closed. This further suggests that the image quality is essential; thus, several of the CT scans would be needed to make a correct classification/diagnosis.

Conclusion

When comparing the statistics of Resnet152, our champion model, and Resnet152-LSTM, the Resnet152 is still better at classifying COVID-19. There are a few reasons why we think the Resnet152-LSTM model did not perform as well. First, the Resnet152-LSTM is a much deeper model, so training took much longer. In our case, on average the training time doubled compared to the champion model. We built the ResNet152-LSTM model, hoping to get better results by combining the best model for our dataset with an LSTM model, but overall, the LSTM model yielded poor results.

Second, LSTM is generally used to address sequence prediction problems in which the order on the observations must be preserved when training models and making prediction. The images collected from patients are taken in sequence from top-to-bottom and side-to-side of the diaphragm. These images are not a sequence of a progressing disease or event and, therefore, our dataset is not necessarily a sequence problem. If our dataset were images taken in consecutive days and showing progression of a disease, then it would be more of a sequence problem and therefore, the LSTM model may perform better, but that is not our case.

Due to time constraint, we used the same parameters on all the models. Also, we didn't have the time to make any modifications to any layers of ResNet152-LSTM model. We believe that if we had more time to adjust the model and test other parameters, then we could improve the Resnet152-LSTM model to have better accuracy on classification. Overall, the experiment was successful because we were able to identify the best model at classifying COVID-19 for the dataset given the same conditions.

References:

1. Mayoclinic.org/diseases-conditions/coronavirus/symptoms-causes/syc-20479963
2. https://www.sciencedirect.com/science/article/pii/S0960077920306081
3. https://www.cdc.gov/mmwr/volumes/69/wr/mm6918e2.htm
4. Rahimzadeh, M.; Attar, A.; Sakhaei, M. A Fully Automated Deep Learning-based Network For Detecting COVID-19 from a New And Large Lung CT Scan Dataset. Preprints 2020, 2020060031 (doi: 10.20944/preprints202006.0031.v3).
5. https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/
6. https://sebastianraschka.com/deep-learning-resources.html
7. https://towardsdatascience.com/understanding-and-coding-a-resnet-in-keras-446d7ff84d33
8. https://www.analyticsvidhya.com/blog/2018/04/solving-an-image-captioning-task-using-deep-learning/
9. https://machinelearningmastery.com/cnn-long-short-term-memory-networks/
10. https://medium.com/analytics-vidhya/cnn-lstm-architecture-and-image-captioning-2351fc18e8d7
11. https://www.kaggle.com/rahulbagga/blood-cell-xception-lstm-merger
12. https://www.tensorflow.org/api_docs/python/tf/keras/layers/Lambda
13. Selvaraju, Ramprasaath R., Cogswell, Michael, Das, Abhishek, Vedantam, Ramakrishna, Parikh, Devi, Batra, Dhruv. Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. 2016 Oct. doi: 10.1007/s11263-019-01228-7. Epub 2016 Oct. https://ui.adsabs.harvard.edu/link_gateway/2016arXiv161002391S/arxiv:1610.02391
14. Emerging 2019 Novel Coronavirus (2019-nCoV) Pneumonia Fengxiang Song, Nannan Shi, Fei Shan, Zhiyong Zhang, Jie Shen, Hongzhou Lu, Yun Ling, Yebin Jiang, and Yuxin Shi. Radiology 2020 295:1, 210-217
15. Feng H, Liu Y, Lv M, Zhong J. A case report of COVID-19 with false negative RT-PCR test: necessity of chest CT. Jpn J Radiol. 2020 May;38(5):409-410. doi: 10.1007/s11604-020-00967-9. Epub 2020 Apr 7. PMID: 32266524; PMCID: PMC7136155.
16. https://www.researchgate.net/publication/330710656_CNN-LSTM_Cascaded_Framework_For_Brain_Tumour_Classification