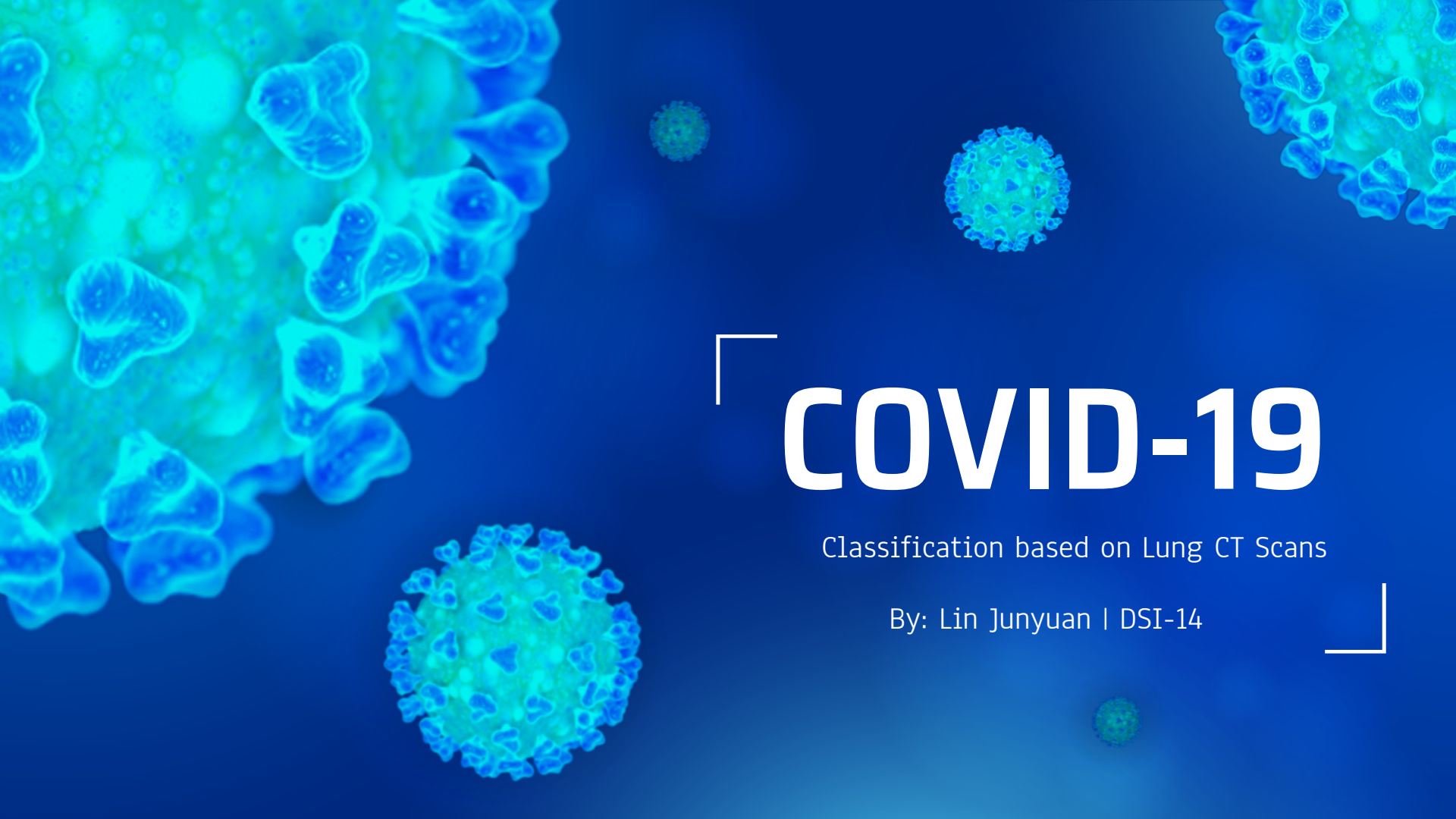
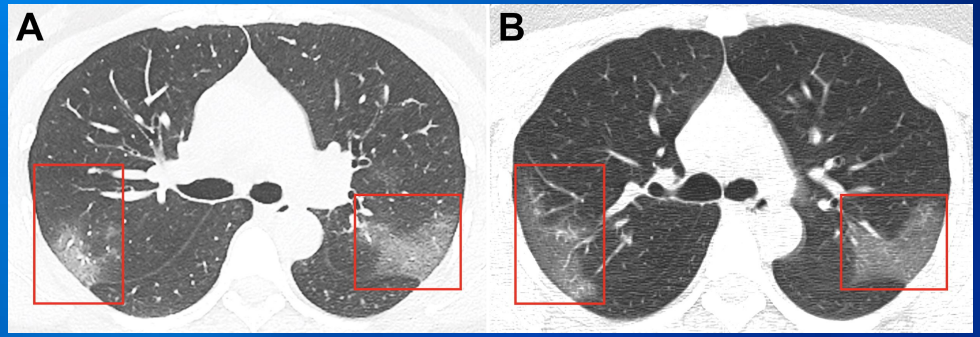


COVID-19

Classification based on Lung CT Scans

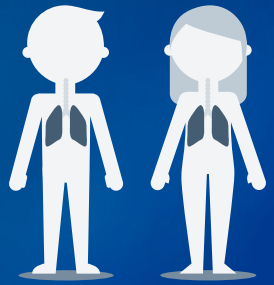
By: Lin Junyuan | DSI-14

INTRODUCTION



Credit: CT Imaging of the 2019 Novel Coronavirus (2019-nCoV) Pneumonia
Junqiang Lei, Junfeng Li, Xun Li, Xiaolong Qi

- Ground Glass Opacities that grows with severity

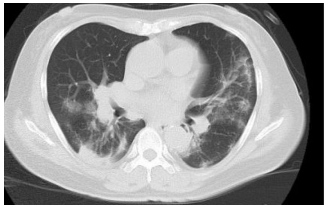


Problem Statement

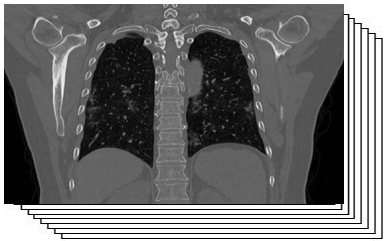
To develop a Convolutional Neural Network-based classification(CNN) model, based on Chest CT Scan data in the axial view, to determine whether Covid infection is present, with a targeted performance of **80%** for both accuracy and sensitivity.

Data preprocessing

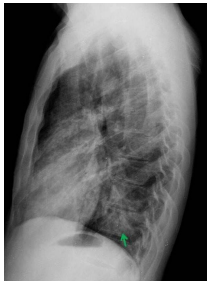
Gathered images from 6 sources, coming in a variety of file formats, views and image resolutions



Axial view
.jpg/.png format



Frontal view
.nii format

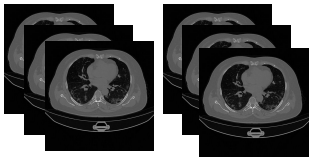


Supine view
*This is actually X-ray

Preprocessing Steps

Load Image

6 slices from each nii file

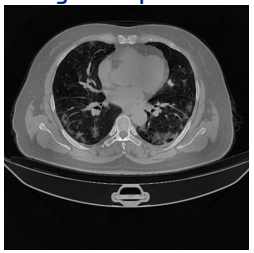


.jpg/.png

Process image

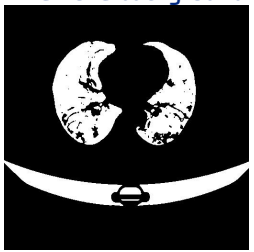
1. Median denoising

2. Histogram equalization

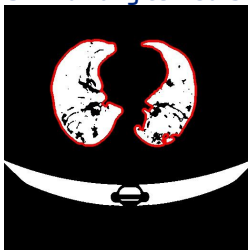


3. Thresholding

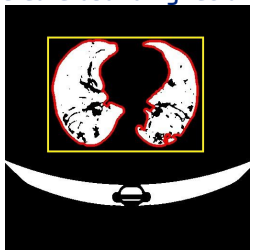
4. Remove background



5. Find Lung contours

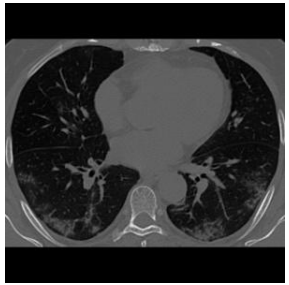


6. Create bounding rectangle

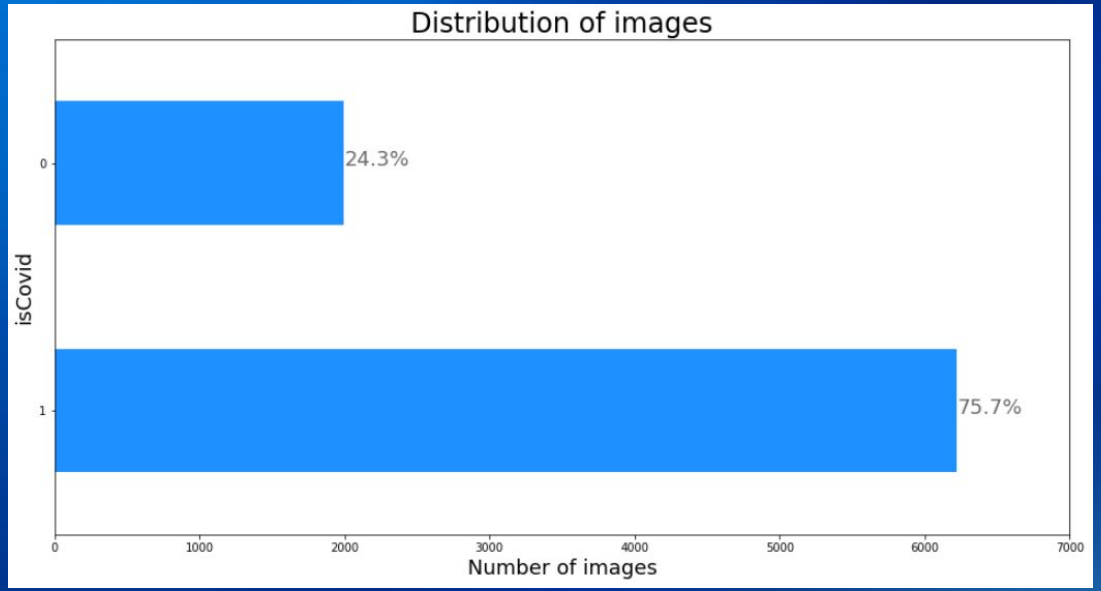


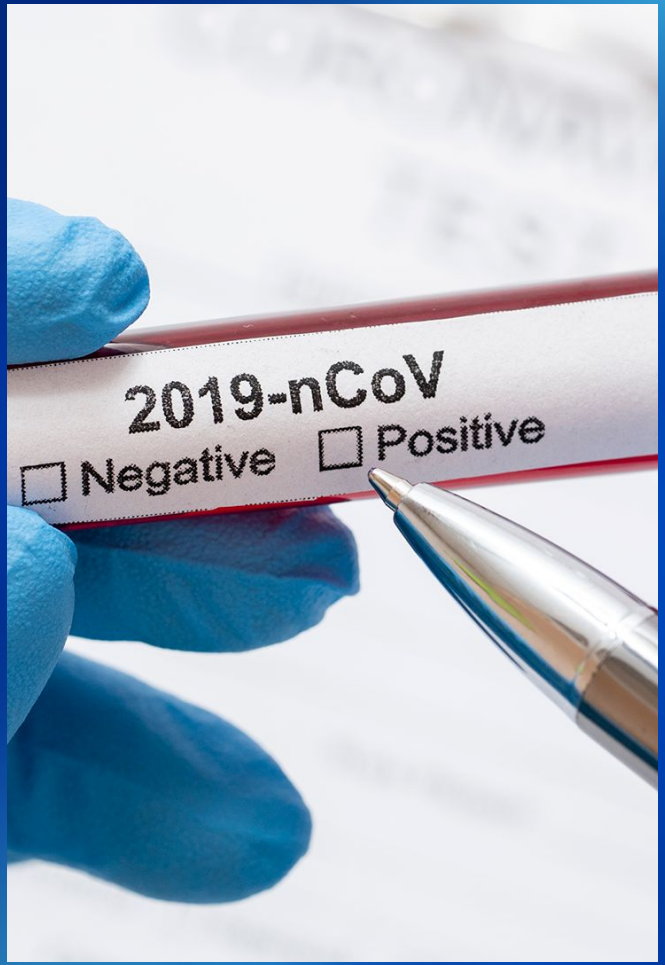
Output image

- Crop original image using bounded region
- Zero pad and resize to (224, 280)



Data distribution





Data preparation

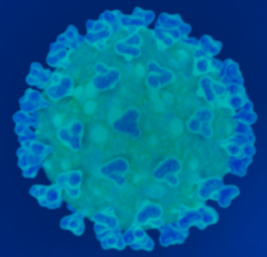
STEP 01 - Train-Test-Split

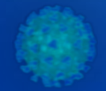
- There are multiple images from each patient.
- Will need to split on patientID to prevent data leakage

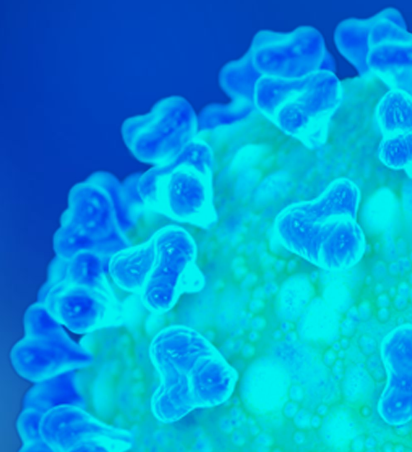
STEP 02 - Data augmentation

- Keras / ImageDataGenerator
- Randomization Parameters
  - Horizontal shift +/- 5%
  - Vertical shift +/- 5%
  - Horizontal flip
  - fill_mode='nearest',
  - Brightness variation +/- 20%

MODELING







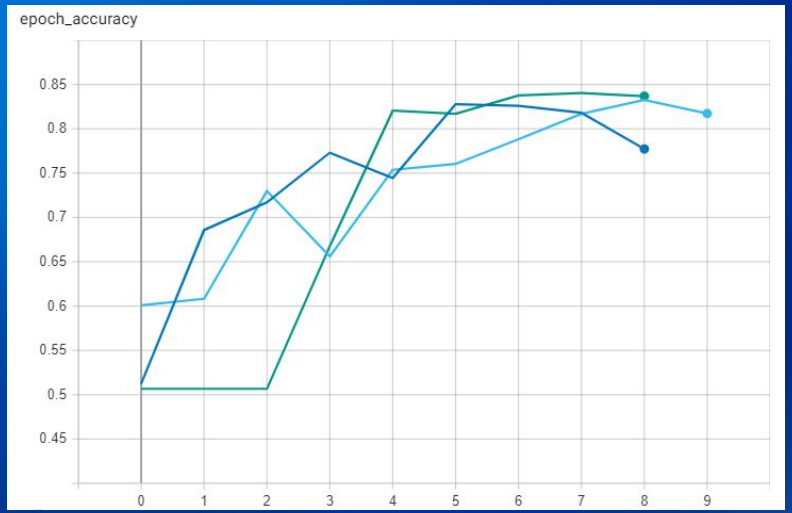
Initialize model structure

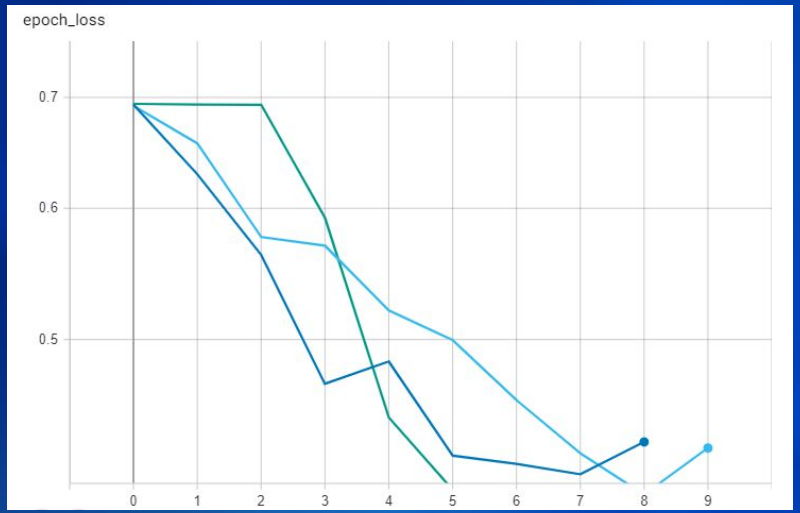
- Conv2d (filters = 32, kernel size = 3, ‘relu’)
- Conv2d (filters = 64, kernel size = 3, ‘relu’)
- Max Pooling (size = (2,2))
- Dropout (ratio=0.25)
- Conv2d (filters = 32, kernel size = 3, ‘relu’)
- Max Pooling (size = (2,2))
- Global Average Pooling
- Dense (filters=256, ‘relu’)
- Dense (filters=128, ‘relu’)
- Dropout (ratio=0.5)
- Dense (filters=64, ‘relu’)
- Dense (filters=1, ‘sigmoid’)

```
Model: "model1_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 222, 278, 32)      320
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 220, 276, 64)      18496
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 110, 138, 64)      0
_________________________________________________________________
dropout (Dropout)            (None, 110, 138, 64)      0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 108, 136, 32)      18464
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 54, 68, 32)        0
_________________________________________________________________
global_average_pooling2d (Gl (None, 32)                0
_________________________________________________________________
dense (Dense)                (None, 256)               8448
_________________________________________________________________
dense_1 (Dense)              (None, 128)               32896
_________________________________________________________________
dropout_1 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_2 (Dense)              (None, 64)                8256
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 65
=================================================================
Total params: 86,945
Trainable params: 86,945
Non-trainable params: 0
```

Hyperparameter tuning [Number of filters]

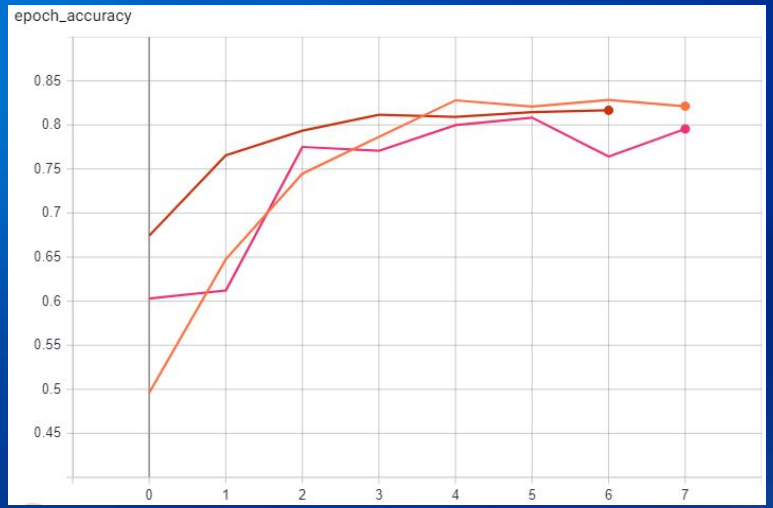
Initial model | model with half number of filters | model with twice number of filters

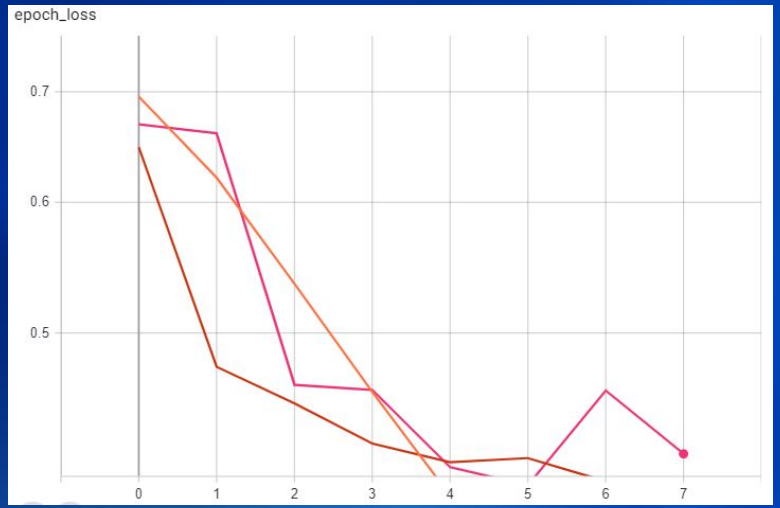




Hyperparameter tuning [Size of kernels]

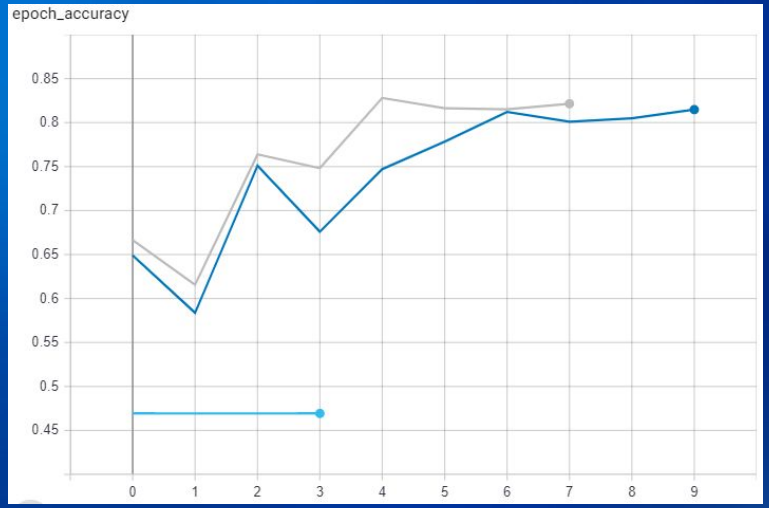
Initial model (size:3) | model with kernel size: 5 | model with kernel size: 7

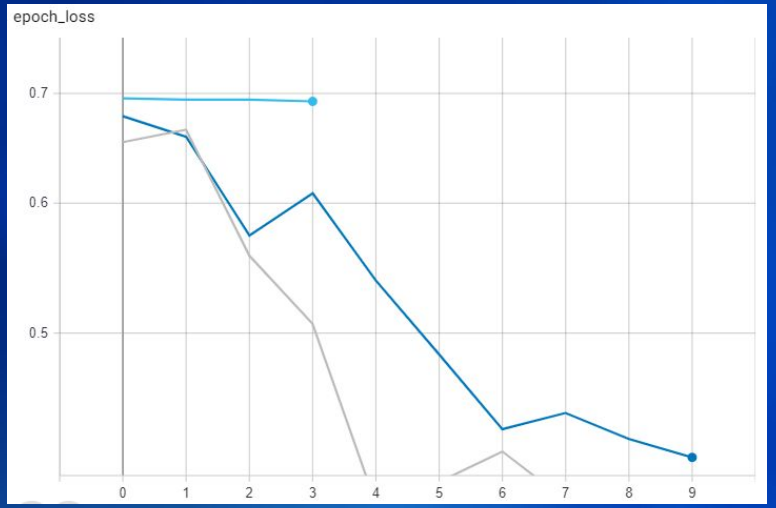




Hyperparameter tuning [Number of layers]

Initial model | model with 1 less conv layer and 1 less dense layer | model with 1 more conv layer and 1 more dense layer





Model evaluation

	model_name	true_negative	false_positive	false_negative	true_positive	accuracy	sensitivity	specificity
0	model1_1	1027	231	337	955	0.78	0.74	0.82
1	model1_2	866	392	74	1218	0.82	0.94	0.69
2	model1_3	947	311	105	1187	0.84	0.92	0.75
3	model2_1	896	395	63	1208	0.82	0.95	0.69
4	model2_2	853	438	32	1239	0.82	0.97	0.66
8	model2_3	917	404	50	1206	0.82	0.96	0.69
5	model3_1	920	321	135	1178	0.82	0.90	0.74
6	model3_2	941	300	173	1140	0.81	0.87	0.76
7	model3_3	1241	0	1313	0	0.49	0.00	1.00

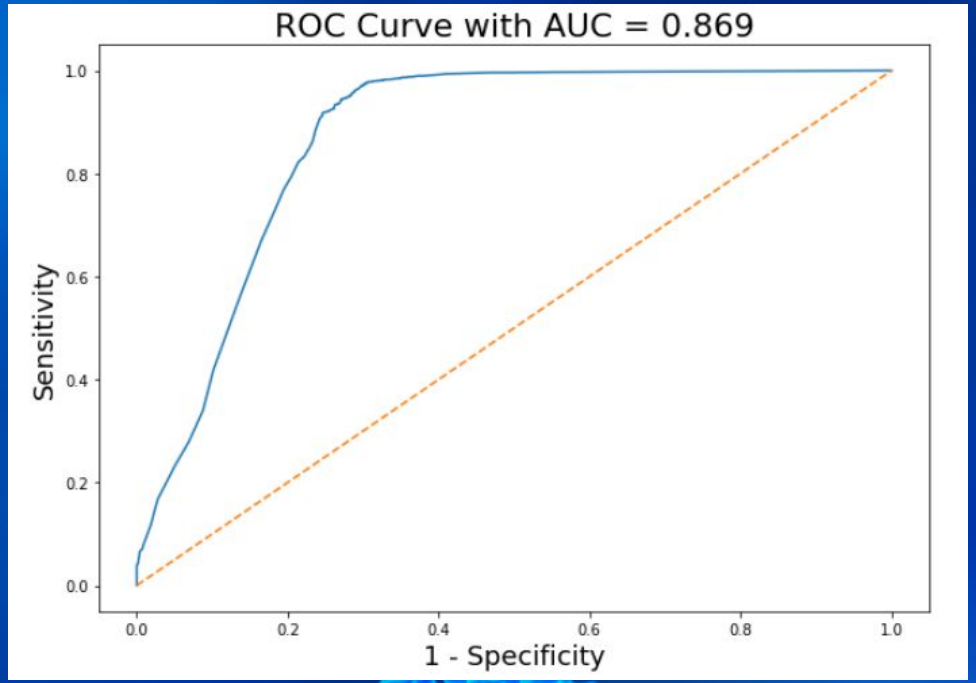
Selected model structure

- Conv2d (filters = 64, kernel size = 5, ‘relu’)
- Conv2d (filters = 128, kernel size = 5, ‘relu’)
- Max Pooling (size = (2,2))
- Dropout (ratio=0.25)
- Conv2d (filters = 64, kernel size = 5, ‘relu’)
- Max Pooling (size = (2,2))
- Global Average Pooling
- Dense (filters=512, ‘relu’)
- Dense (filters=256, ‘relu’)
- Dropout (ratio=0.5)
- Dense (filters=128, ‘relu’)
- Dense (filters=1, ‘sigmoid’)

```
Model: "model_selected"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 220, 276, 64)      1664
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 216, 272, 128)     204928
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 108, 136, 128)     0
_________________________________________________________________
dropout (Dropout)            (None, 108, 136, 128)     0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 104, 132, 64)      204864
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 52, 66, 64)        0
_________________________________________________________________
global_average_pooling2d (Gl (None, 64)                0
_________________________________________________________________
dense (Dense)                (None, 512)               33280
_________________________________________________________________
dense_1 (Dense)              (None, 256)               131328
_________________________________________________________________
dropout_1 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_2 (Dense)              (None, 128)               32896
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 129
=================================================================
Total params: 609,089
Trainable params: 609,089
Non-trainable params: 0
```

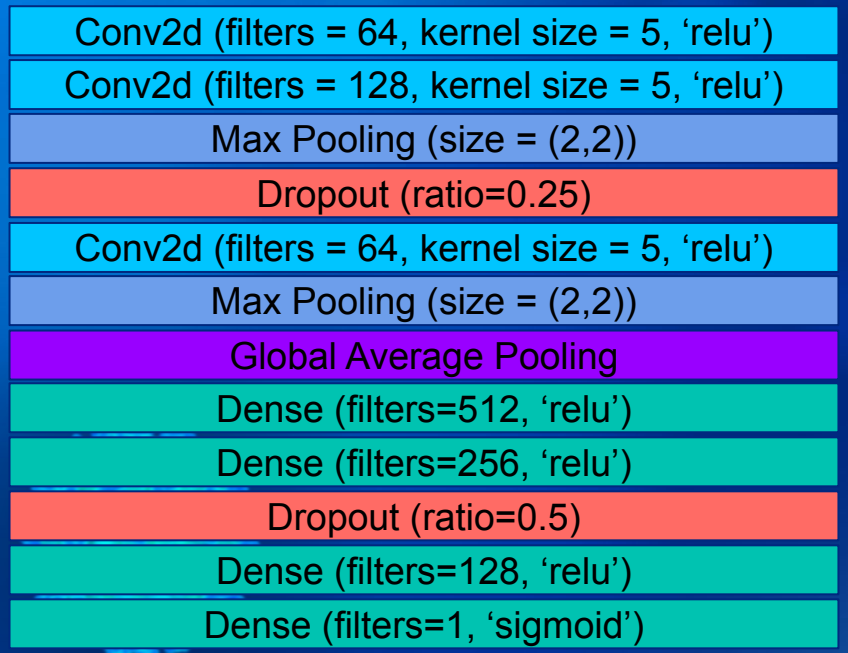
Selected model performance

	model_name	true_negative	false_positive	false_negative	true_positive	accuracy	sensitivity	specificity
0	model1_1	1027	231	337	955	0.78	0.74	0.82
1	model1_2	866	392	74	1218	0.82	0.94	0.69
2	model1_3	947	311	105	1187	0.84	0.92	0.75
3	model2_1	896	395	63	1208	0.82	0.95	0.69
4	model2_2	853	438	32	1239	0.82	0.97	0.66
8	model2_3	917	404	50	1206	0.82	0.96	0.69
5	model3_1	920	321	135	1178	0.82	0.90	0.74
6	model3_2	941	300	173	1140	0.81	0.87	0.76
7	model3_3	1241	0	1313	0	0.49	0.00	1.00
9	model_selected	947	374	61	1195	0.83	0.95	0.72


ROC Curve with AUC = 0.869
Sensitivity
1 - Specificity
0.0
0.2
0.4
0.6
0.8
1.0

Conclusion and Recommendations

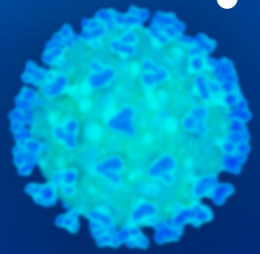
Conclusions



- The selected model was able to perform to our targeted values of minimum 80% for both accuracy and sensitivity.
- It has attained an ROC-AUC score of 0.86.

Recommendations

- To continue to build up the database
  (More quality data -> Better performance)
- To do Multi-Class classification to separate between Covid infection and other types of viral/bacterial infection
- To try other methodologies, i.e. Transfer Learning, 3D CT Scan CNNs



THANKS!

https://www.linkedin.com/in/junyuanlin/

DID YOU KNOW?

CNN was inspired by research done on cat brains.

"Receptive Fields of Single Neurons in the Cat's Striate Cortex" conducted by Hubel and Wiesel