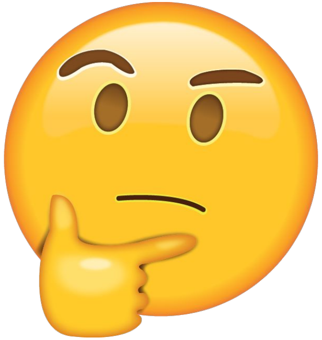
Semantic Segmentation

Covid-19 CT Scans

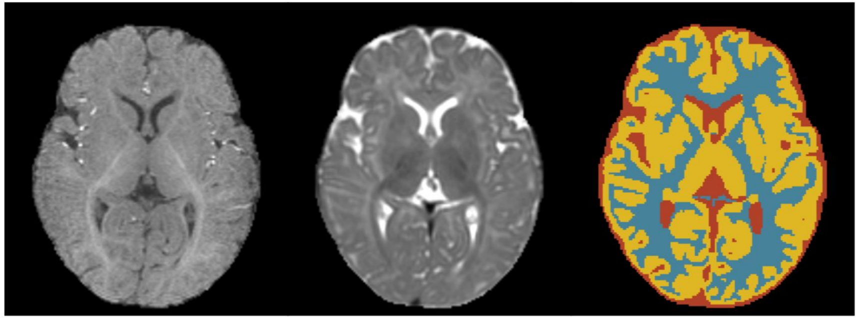
Sheila Monteiro Augusto

What is Semantic Segmentation?



https://bit.ly/2JMSER4

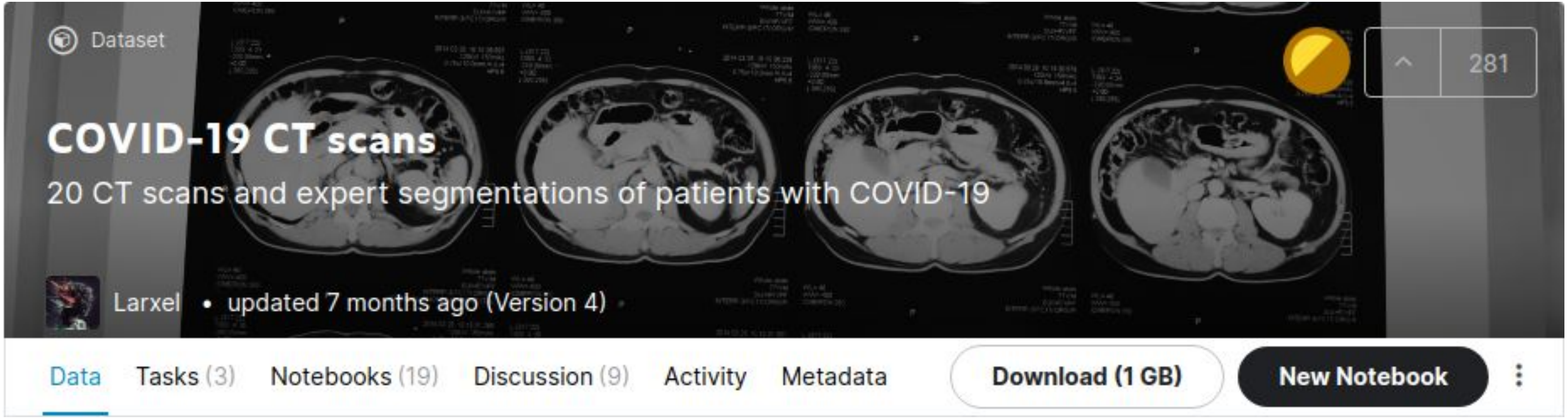
https://bit.ly/40MfeEb



https://arxiv.org/pdf/1712.05319v2.pdf

Problem

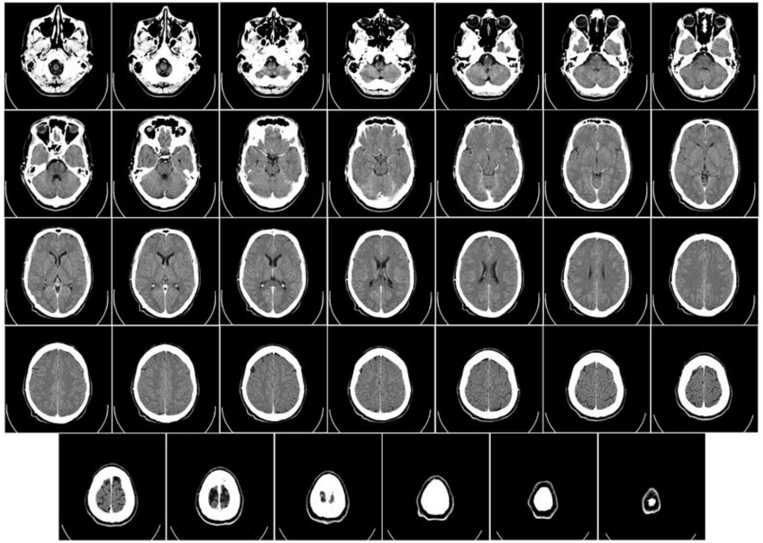
Kaggle Challenge



https://www.kaggle.com/andrewmvd/covid19-ct-scans

CT-Scan

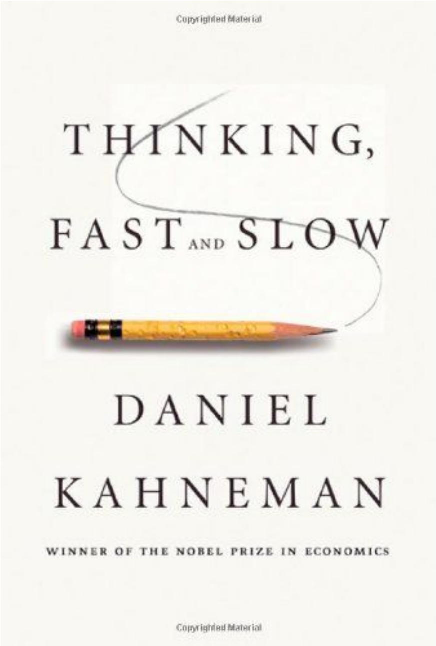
https://bit.ly/37DewGP



https://bit.ly/40Ftw9Q

Radiologic Errors

“When asked to evaluate the same information twice, they frequently give different answers. The extent of the inconsistency is often a matter of real concern. Experienced radiologists who evaluate chest X-rays as “normal” or “abnormal” contradict themselves 20% of the time when they see the same picture on separate occasions.”



Dataset

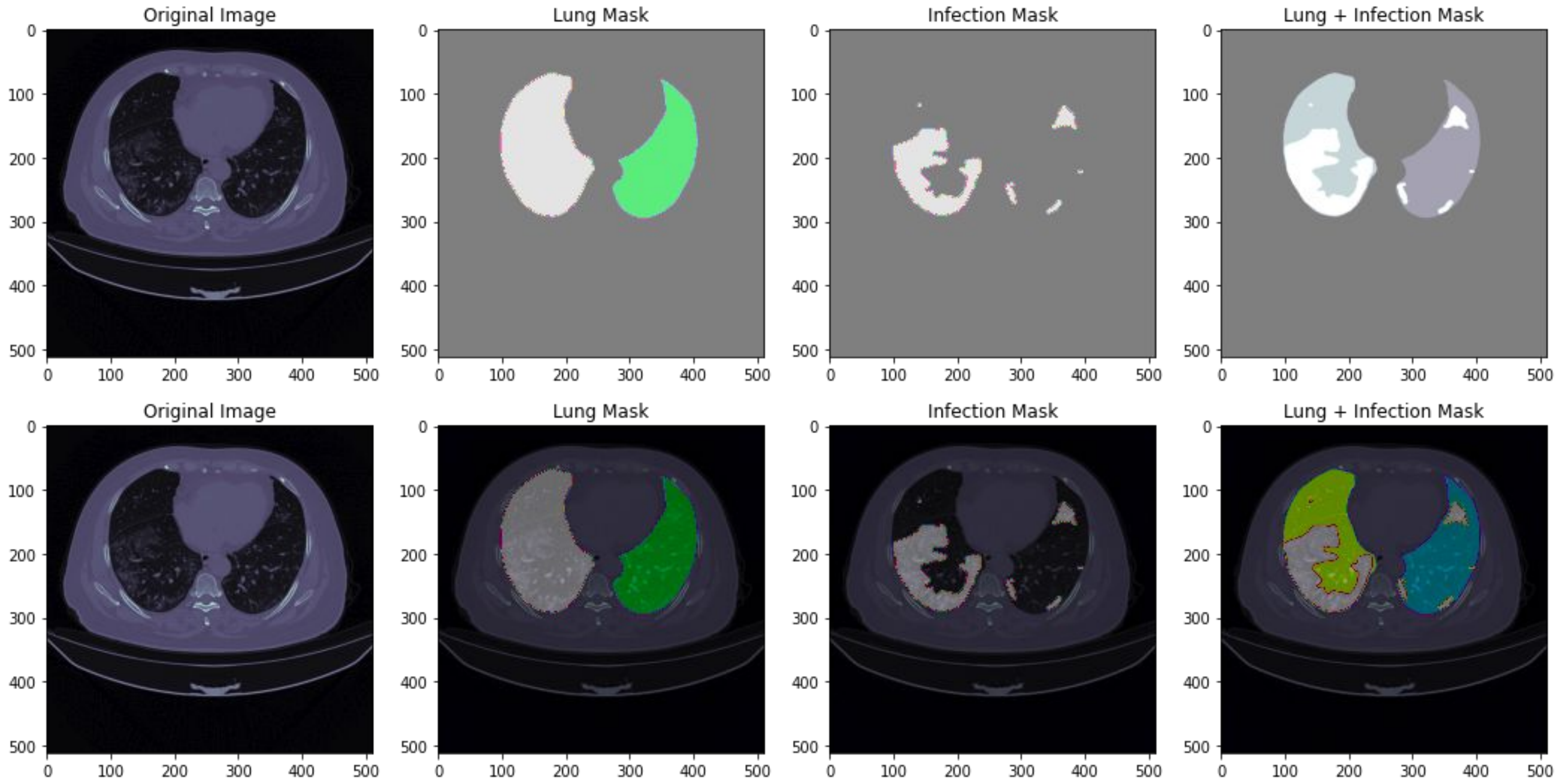
coronacases_001.nii.gz	93.2 MB
coronacases_002.nii.gz	70.1 MB
coronacases_003.nii.gz	67.3 MB
coronacases_004.nii.gz	88.9 MB
coronacases_005.nii.gz	96.5 MB
coronacases_006.nii.gz	71.6 MB
coronacases_007.nii.gz	85.0 MB
coronacases_008.nii.gz	96.9 MB
coronacases_009.nii.gz	84.7 MB
coronacases_010.nii.gz	98.7 MB
radiopaedia_10_85902_1.nii.gz	9.6 MB
radiopaedia_10_85902_3.nii.gz	120.0 MB
radiopaedia_14_85914_0.nii.gz	21.9 MB
radiopaedia_27_86410_0.nii.gz	20.2 MB
radiopaedia_29_86490_1.nii.gz	10.2 MB
radiopaedia_29_86491_1.nii.gz	11.3 MB
radiopaedia_36_86526_0.nii.gz	8.3 MB
radiopaedia_40_86625_0.nii.gz	28.3 MB
radiopaedia_4_85506_1.nii.gz	10.1 MB
radiopaedia_7_85703_0.nii.gz	12.2 MB

http://doi.org/10.5281/zenodo.3757476

```
for i in range(raw_data.shape[0]):
  ct = read_nii(raw_data['ct_scan'][i])
  print(f'{i+1} {ct.shape}')
```

```
1 (512, 512, 301)
2 (512, 512, 200)
3 (512, 512, 200)
4 (512, 512, 270)
5 (512, 512, 290)
6 (512, 512, 213)
7 (512, 512, 249)
8 (512, 512, 301)
9 (512, 512, 256)
10 (512, 512, 301)
11 (630, 630, 39)
12 (630, 630, 418)
13 (401, 630, 110)
14 (630, 630, 66)
15 (630, 630, 42)
16 (630, 630, 42)
17 (630, 630, 45)
18 (630, 630, 93)
19 (630, 630, 39)
20 (630, 630, 45)
```

Dataset Images



Neural Network Architecture

Why U-Net?

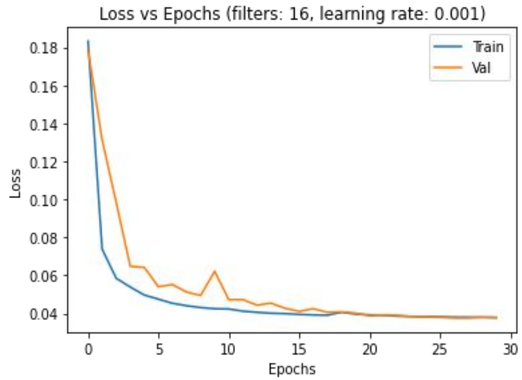
U-Net: Convolutional Networks for Biomedical Image Segmentation

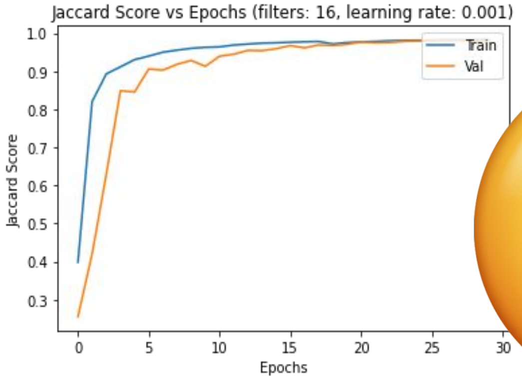
Olaf Ronneberger, Philipp Fischer, and Thomas Brox

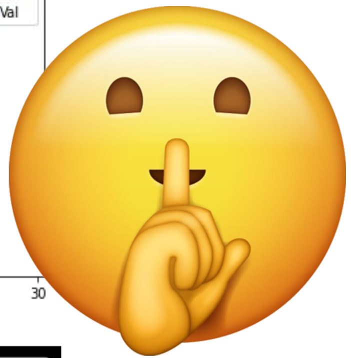
Computer Science Department and BIOSS Centre for Biological Signalling Studies,
University of Freiburg, Germany
ronneber@informatik.uni-freiburg.de,
WWW home page: http://lmb.informatik.uni-freiburg.de/

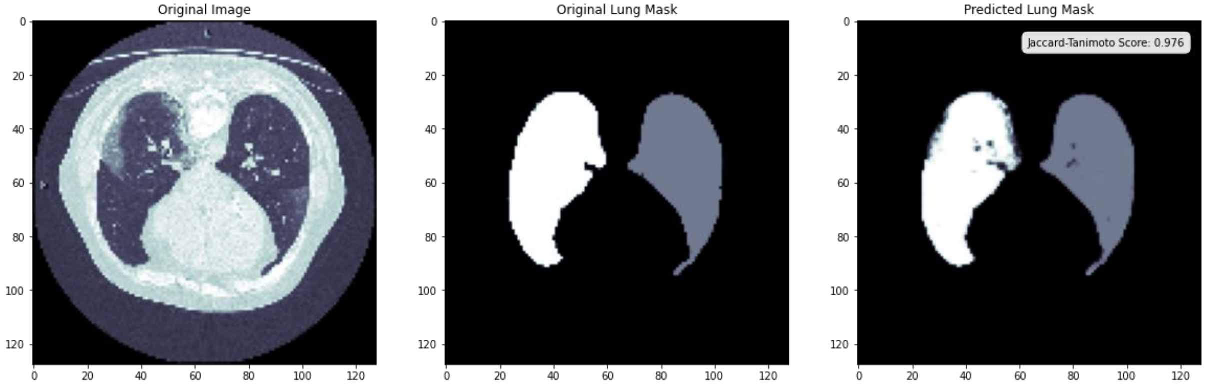
Abstract. There is large consent that successful training of deep networks requires many thousand annotated training samples. In this paper, we present a network and training strategy that relies on the strong use of data augmentation to use the available annotated samples more efficiently. The architecture consists of a contracting path to capture context and a symmetric expanding path that enables precise localization. We show that such a network can be trained end-to-end from very few images and outperforms the prior best method (a sliding-window convolutional network) on the ISBI challenge for segmentation of neuronal structures in electron microscopic stacks. Using the same network trained on transmitted light microscopy images (phase contrast and DIC) we won the ISBI cell tracking challenge 2015 in these categories by a large margin. Moreover, the network is fast. Segmentation of a 512x512 image takes less than a second on a recent GPU. The full implementation (based on Caffe) and the trained networks are available at http://lmb.informatik.uni-freiburg.de/people/ronneber/u-net.

Why U-Net???

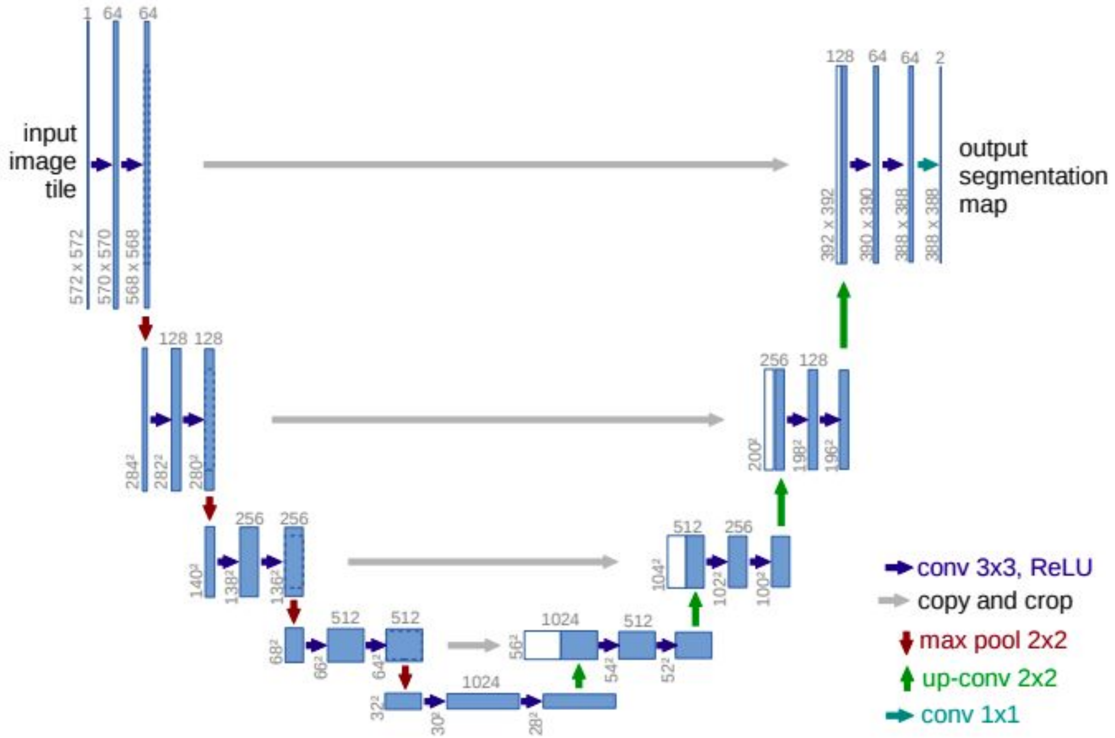

Loss vs Epochs (filters: 16, learning rate: 0.001)
Train
Val
Loss
Epochs


Jaccard Score vs Epochs (filters: 16, learning rate: 0.001)
Train
Val
Jaccard Score
Epochs




Original Image
Original Lung Mask
Predicted Lung Mask
Jaccard-Tanimoto Score: 0.976

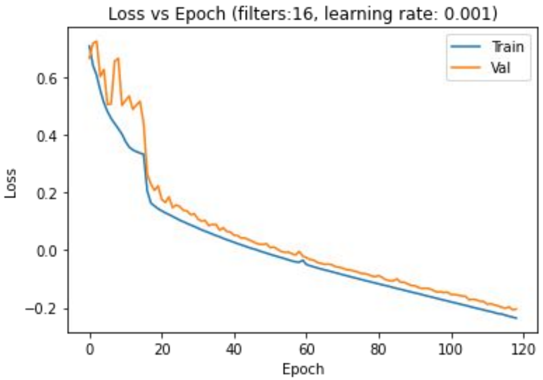
U-Net Architecture

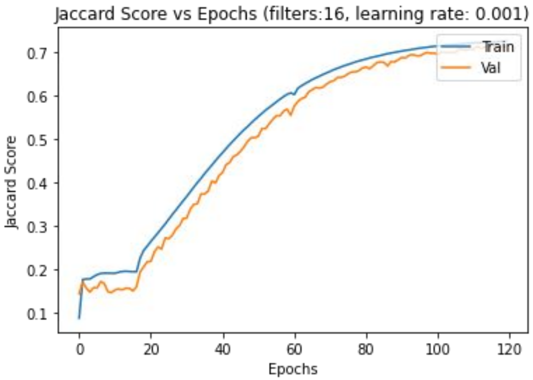


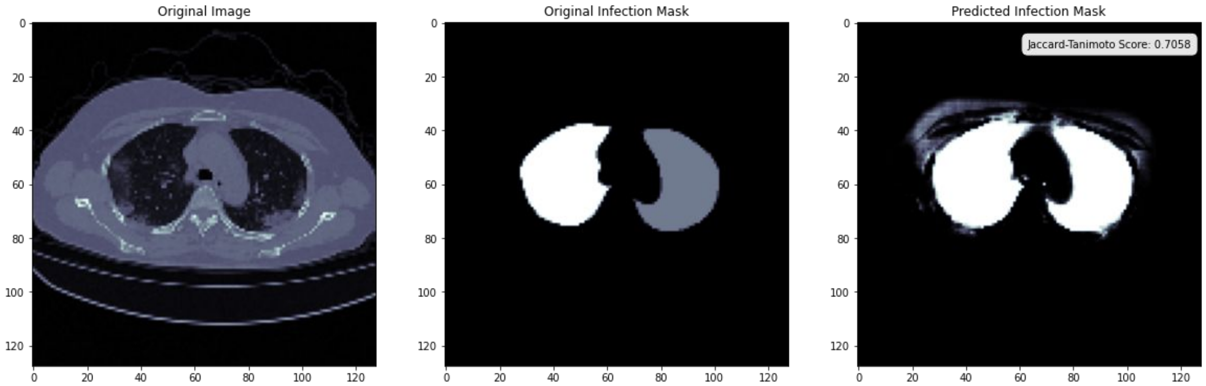
https://arxiv.org/pdf/1505.04597.pdf

Preprocessing

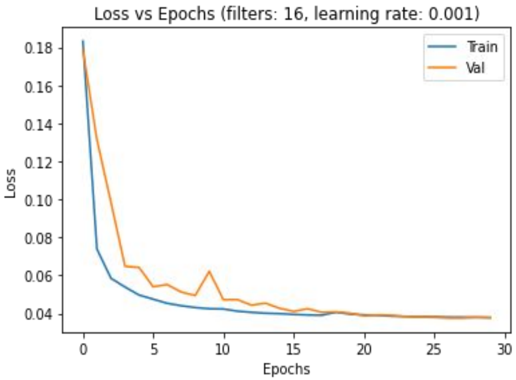
Without Preprocessing

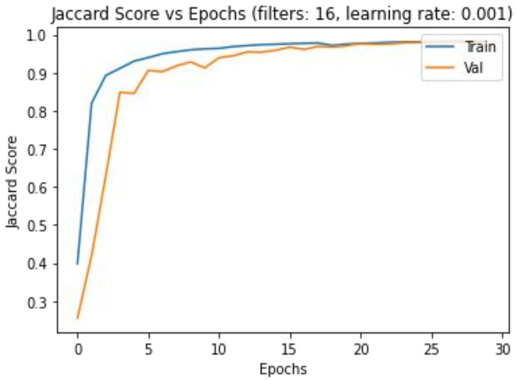


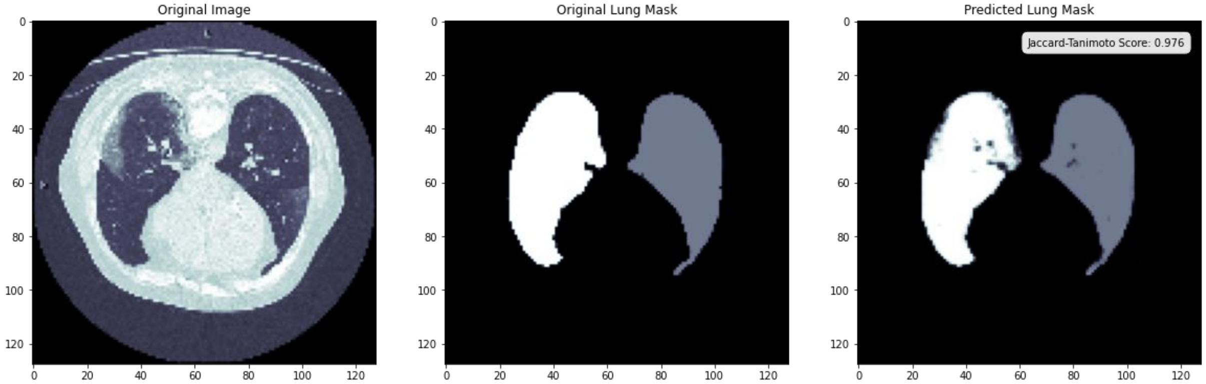




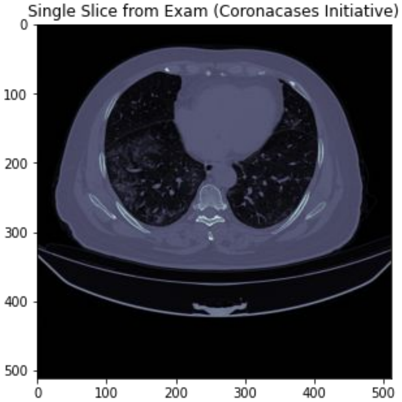
With Preprocessing

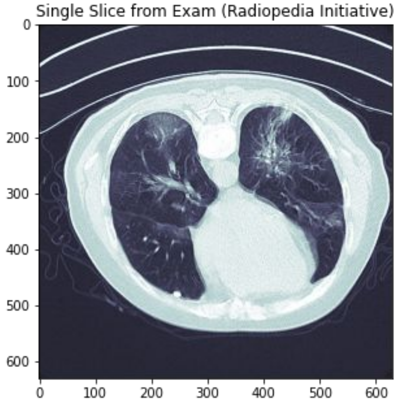


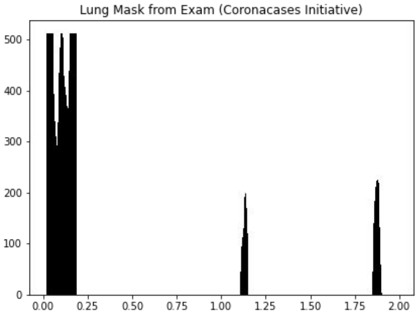


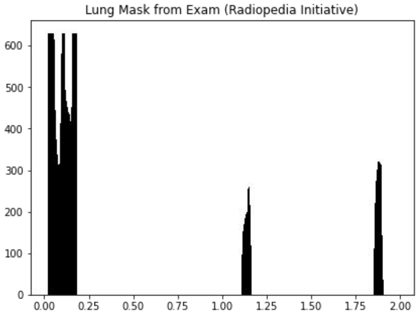


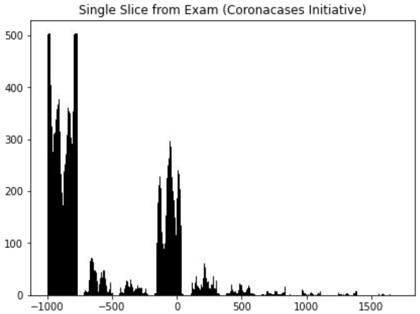
Saturation Level - Images and Masks

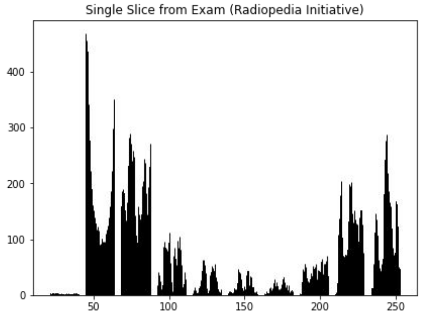


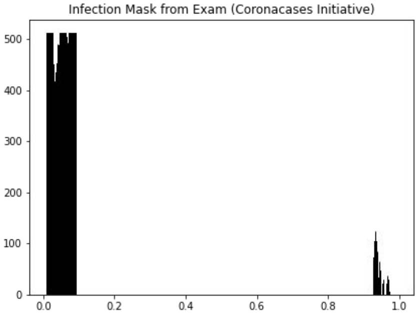


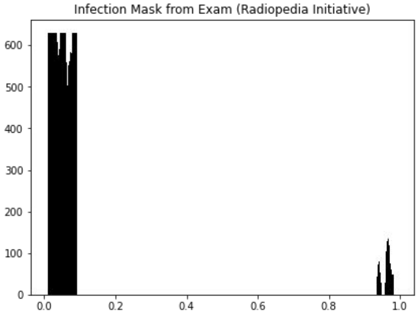






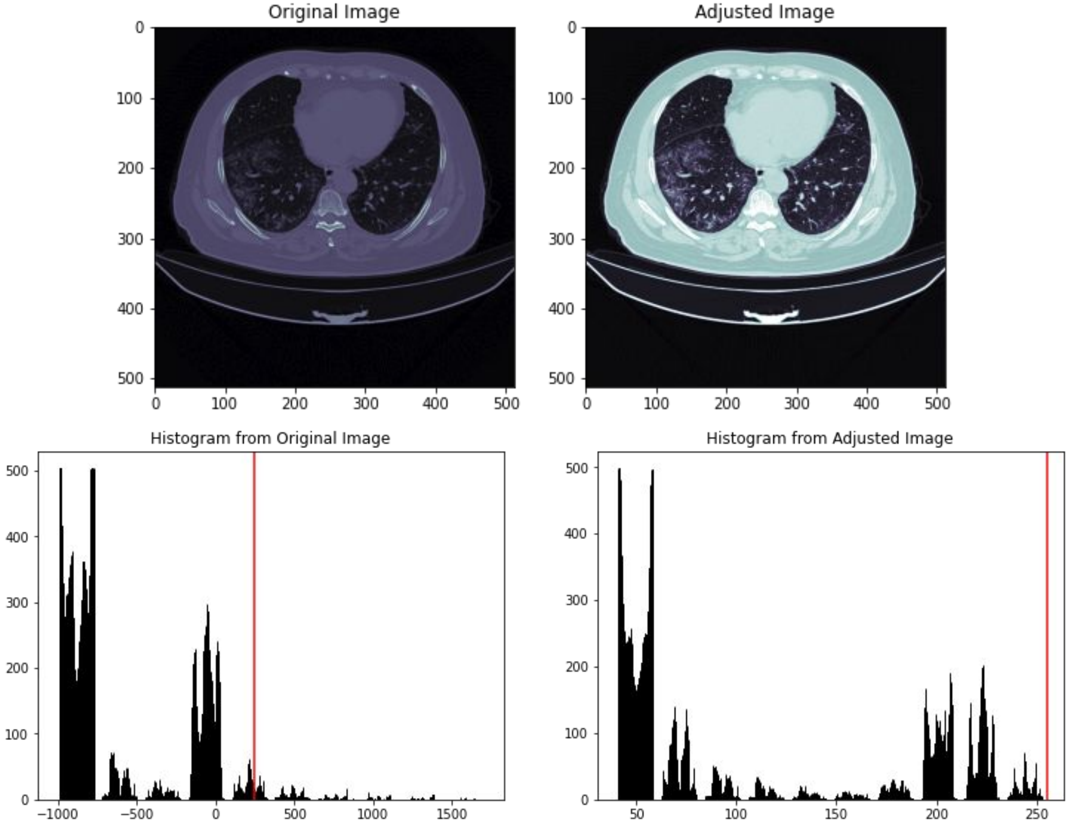






Saturation Level Adjustment - *Coronacases* Exams

1. Adjust saturation level to values between [-1250,250];
2. Normalize pixels to values between [0,255].



https://gitee.com/junma11/COVID-19-CT-Seg-Benchmark

Next Steps...

- Resize ct-scan and masks to 128 x 128 pixels;
- Normalize pixels from exams and masks to values between [0,1];
- Save resized images in *.HDF5 format:

```
with h5py.File('/content/drive/MyDrive/1-Projeto_Final_DL/Covid_Resized.h5','w') as h5f:
  h5f.create_dataset("Resized_Images/ct_scan", data=ct_images, compression="lzf", chunks=(True))
  h5f.create_dataset("Resized_Images/lung_mask", data=lung_mask, compression="lzf", chunks=True)
  h5f.create_dataset("Resized_Images/infection_mask", data=infection_mask, compression="lzf", chunks=True)
h5f.close()
```

Model Evaluation

Jaccard-Tanimoto Score

$$J(A,B) = \frac{A \cdot B}{\|A\|^2 + \|B\|^2 - A \cdot B},$$

$$A \cdot B = \sum_i A_i B_i,$$

$$\|A\|^2 = \sum_i A_i^2.$$

```
smooth = 1e-3 # ~1/255.

def jaccard_coef(y_true, y_pred):

    intersection = K.sum(y_true * y_pred, axis=[0,-2,-3])
    sum_ = K.sum(y_true*y_true + y_pred*y_pred, axis=[0,-2,-3])

    jac = (intersection + smooth) / (sum_ - intersection + smooth)

    return K.mean(jac)
```

Loss: Binary Cross-Entropy (BCE)

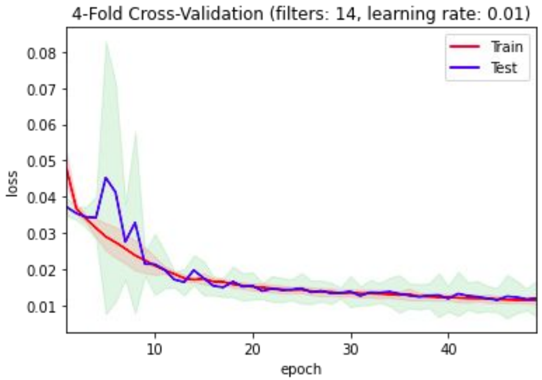
$$BCE = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i),$$

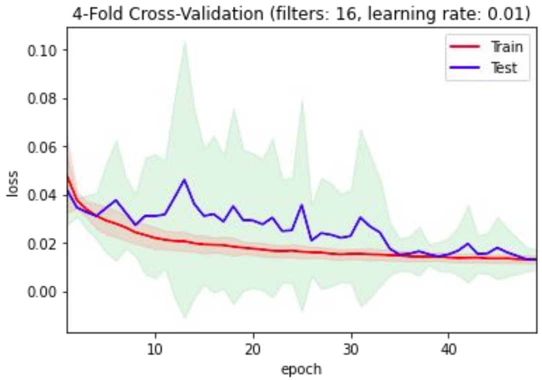
$$N = 2$$
$$y_i = \text{label}$$
$$\hat{y}_i = \text{label's predicted probability}$$

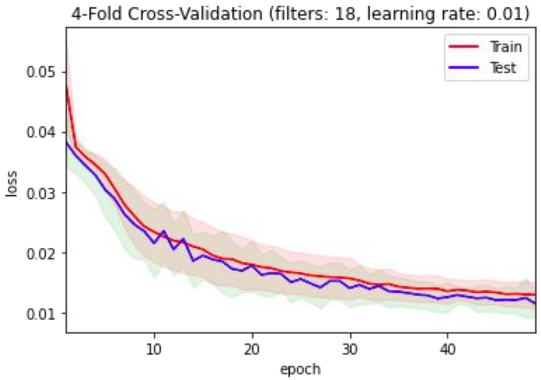
4-fold Cross-Validation (Infection Data)

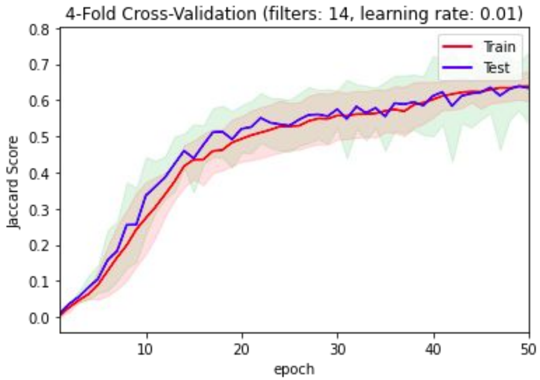
lr = learning rate
dp = dropout

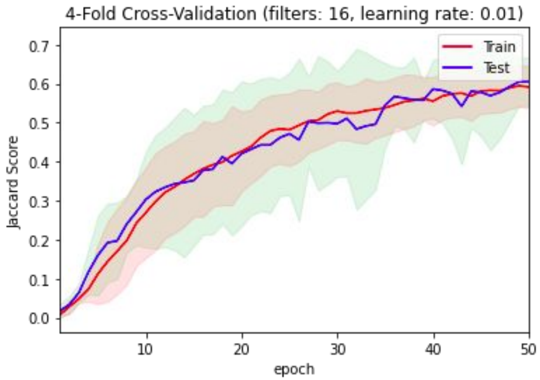
Infection Dataset (lr: 0.01, dp: 0.3)

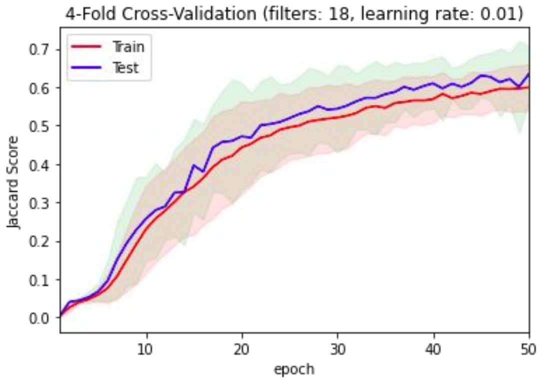




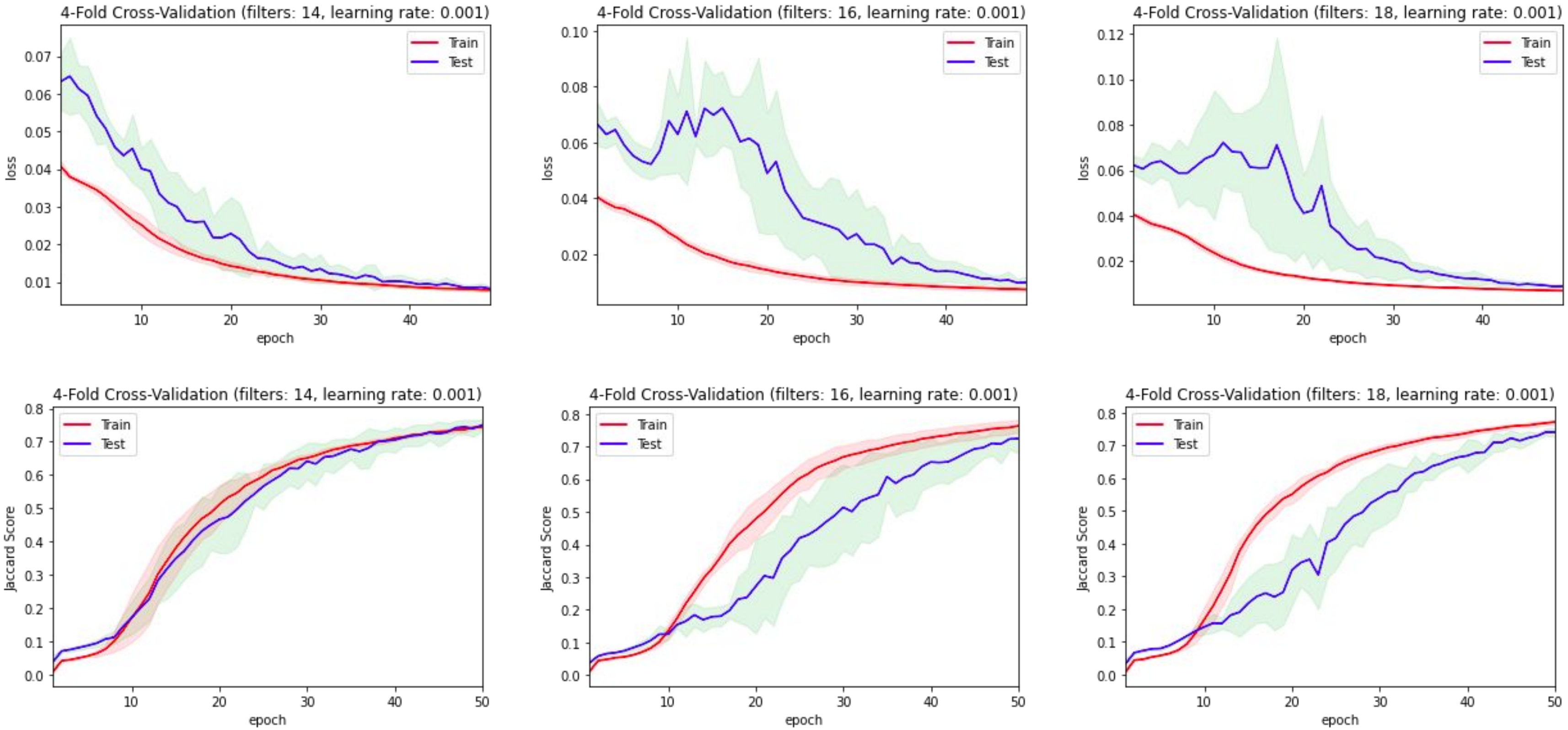




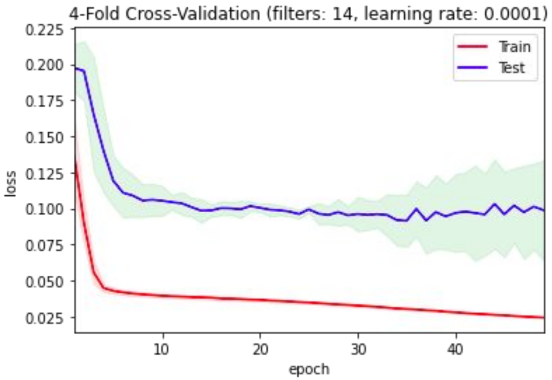


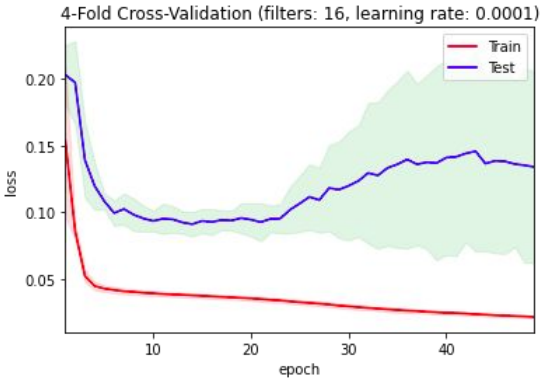


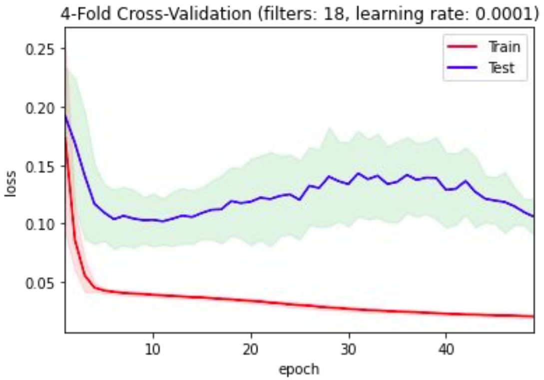
Infection Dataset (lr: 0.001, dp: 0.3)

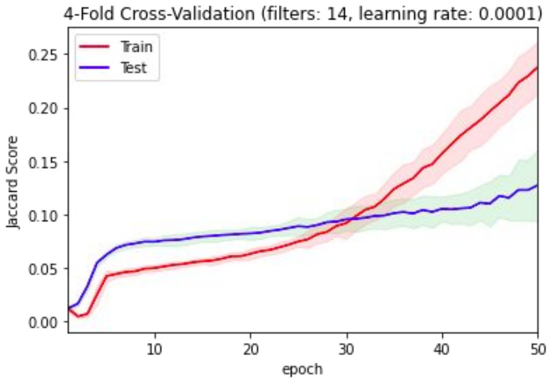


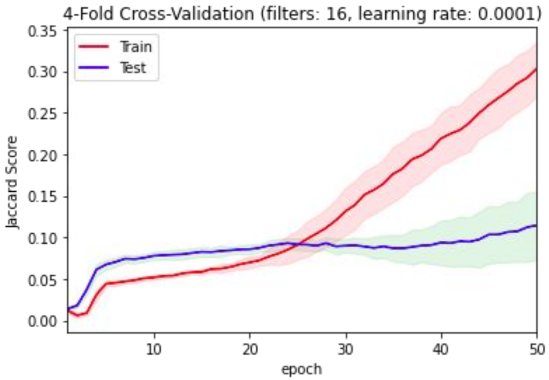
Infection Dataset (lr: 0.0001, dp: 0.3)

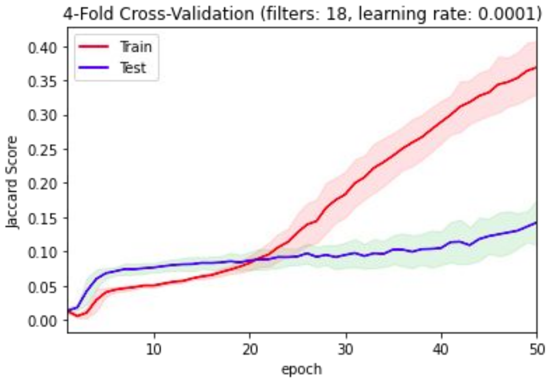








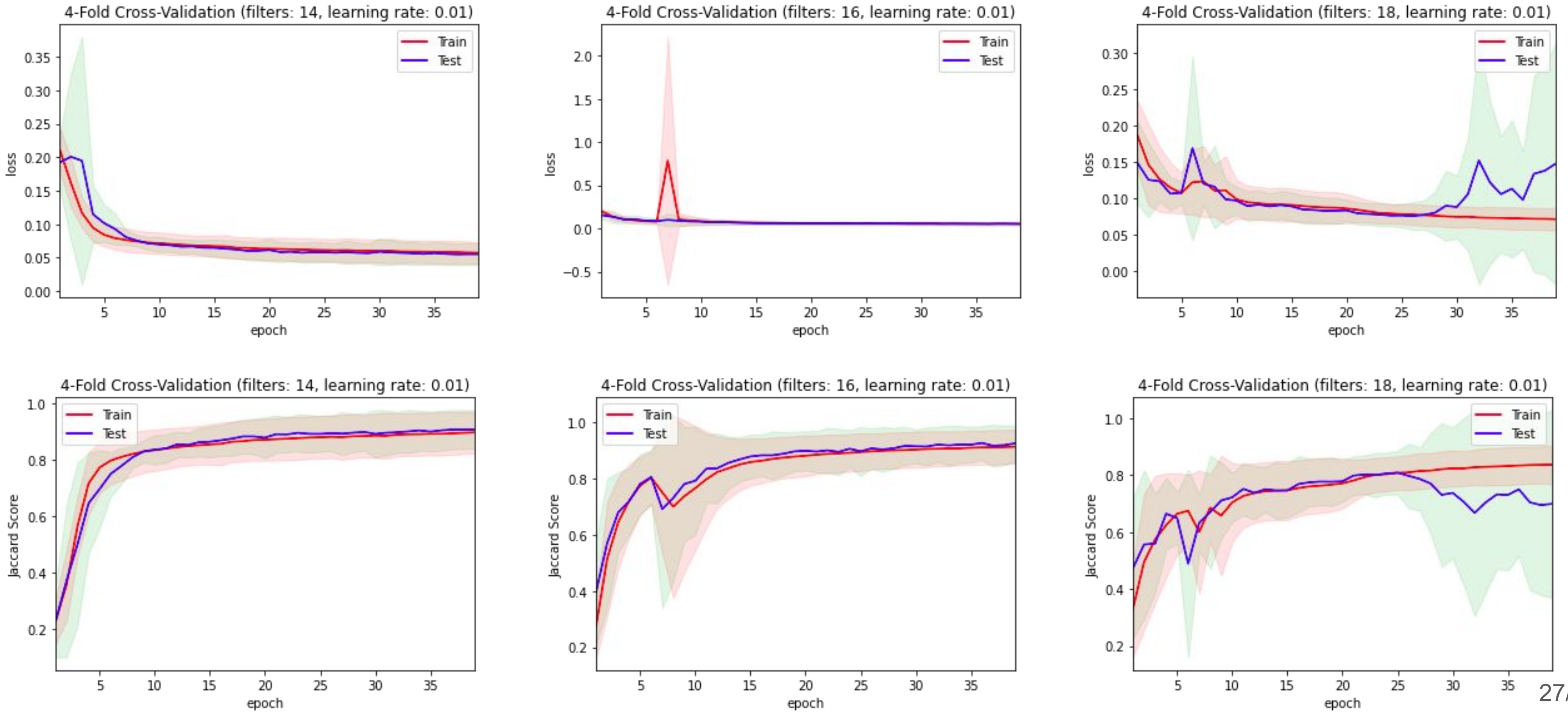




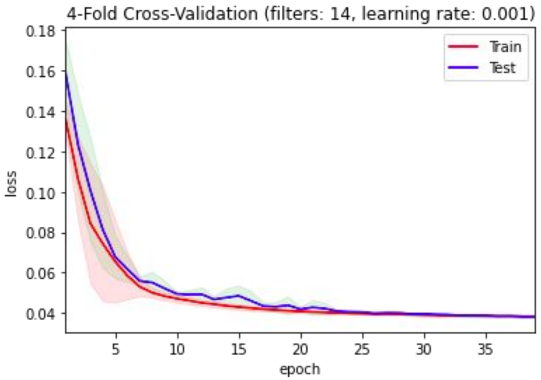
4-fold Cross-Validation (Lung Data)

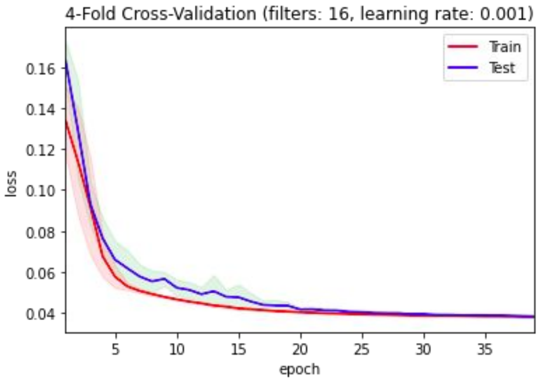
lr = learning rate
dp = dropout

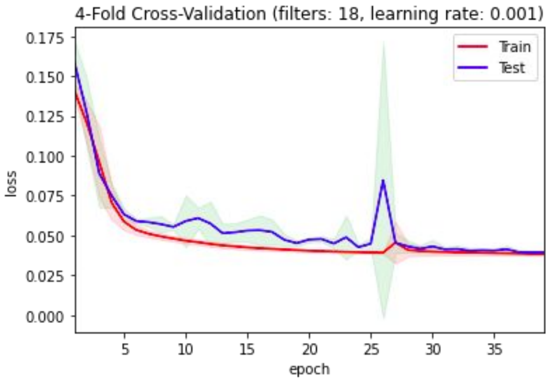
Lung Dataset (lr: 0.01, dp: 0.3)

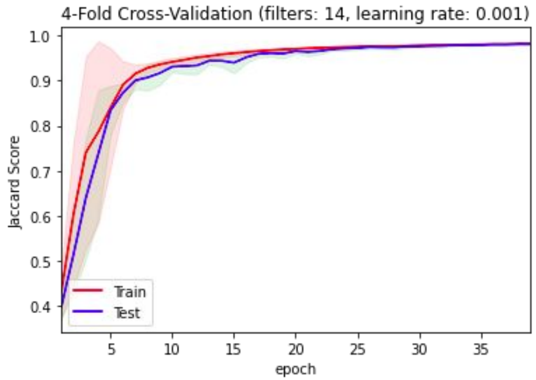


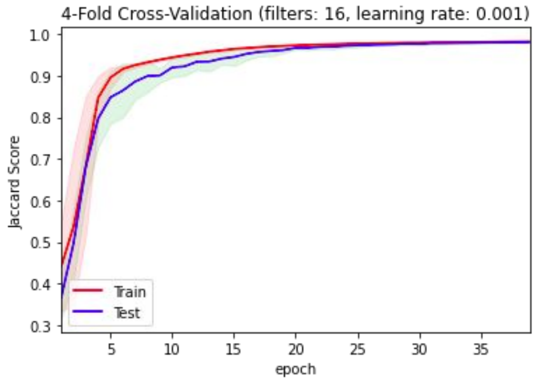
Lung Dataset (lr: 0.001, dp: 0.3)

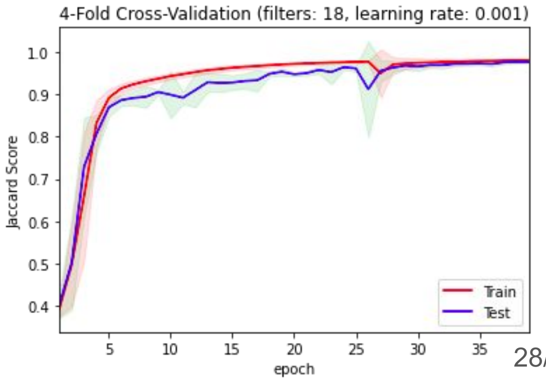




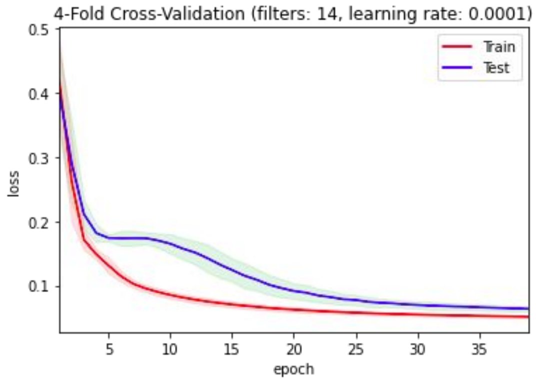


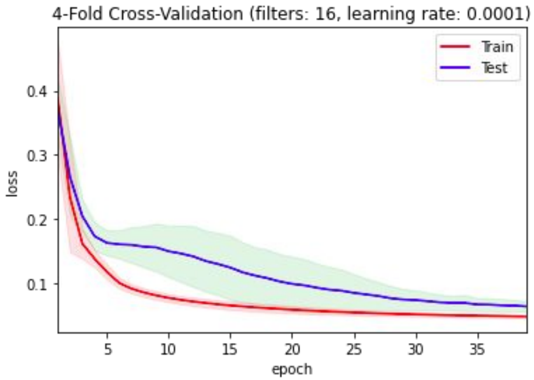


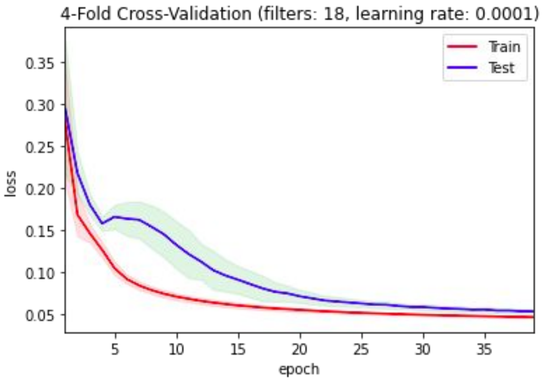


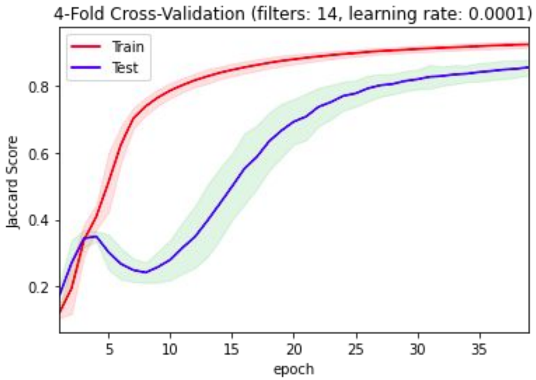


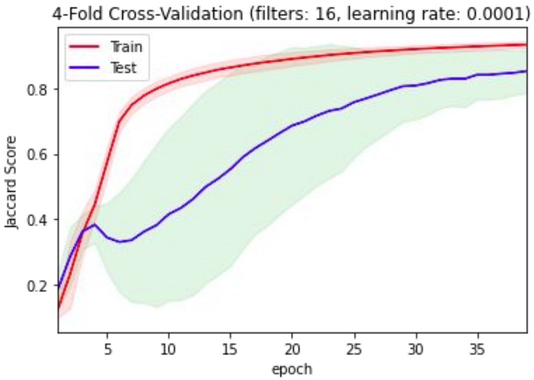
Lung Dataset (lr: 0.0001, dp: 0.3)

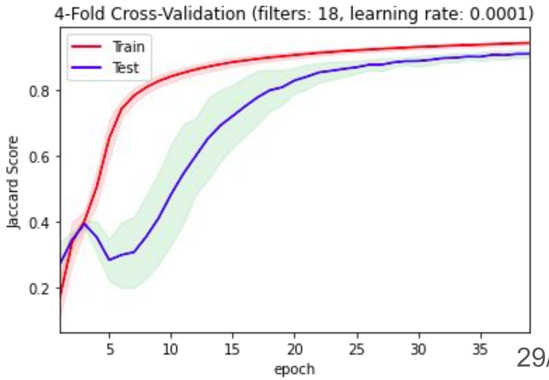












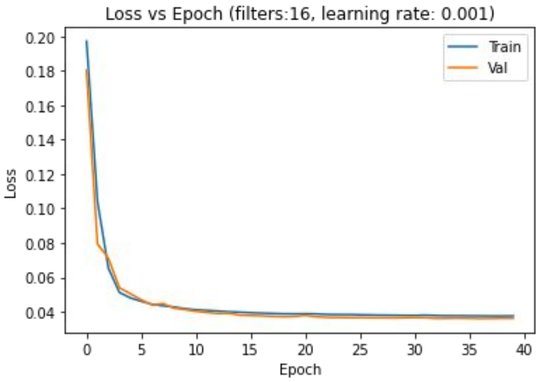
Training and Prediction

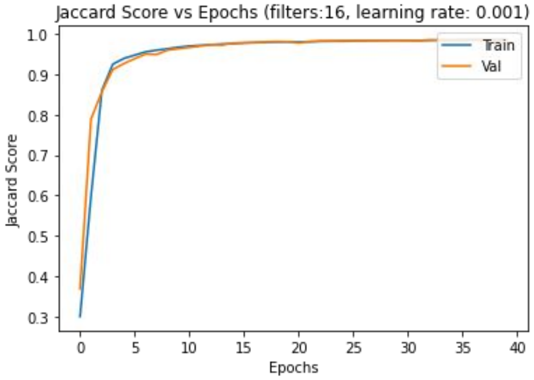
Train, Validation and Test Split

```
from sklearn.model_selection import train_test_split

X = np.expand_dims(ct_data.copy(),axis=-1)
y = np.expand_dims(infection_data.copy(),axis=-1)

X_train, ct_test, y_train, mask_test = train_test_split(X, y,test_size = 0.2)

ct_train, ct_val, mask_train, mask_val = train_test_split(X_train, y_train,test_size = 0.3)
```

```
---------------------- Dimensões dos Datasets ----------------------

Imagens para o Treino: (1971, 128, 128, 1)
Máscaras para o Treino: (1971, 128, 128, 1)

Imagens para o Validação: (845, 128, 128, 1)
Máscaras para o Validação: (845, 128, 128, 1)

Imagens para o Teste: (704, 128, 128, 1)
Máscaras para o Teste: (704, 128, 128, 1)
```

Lung Dataset (dropout: 0.3)

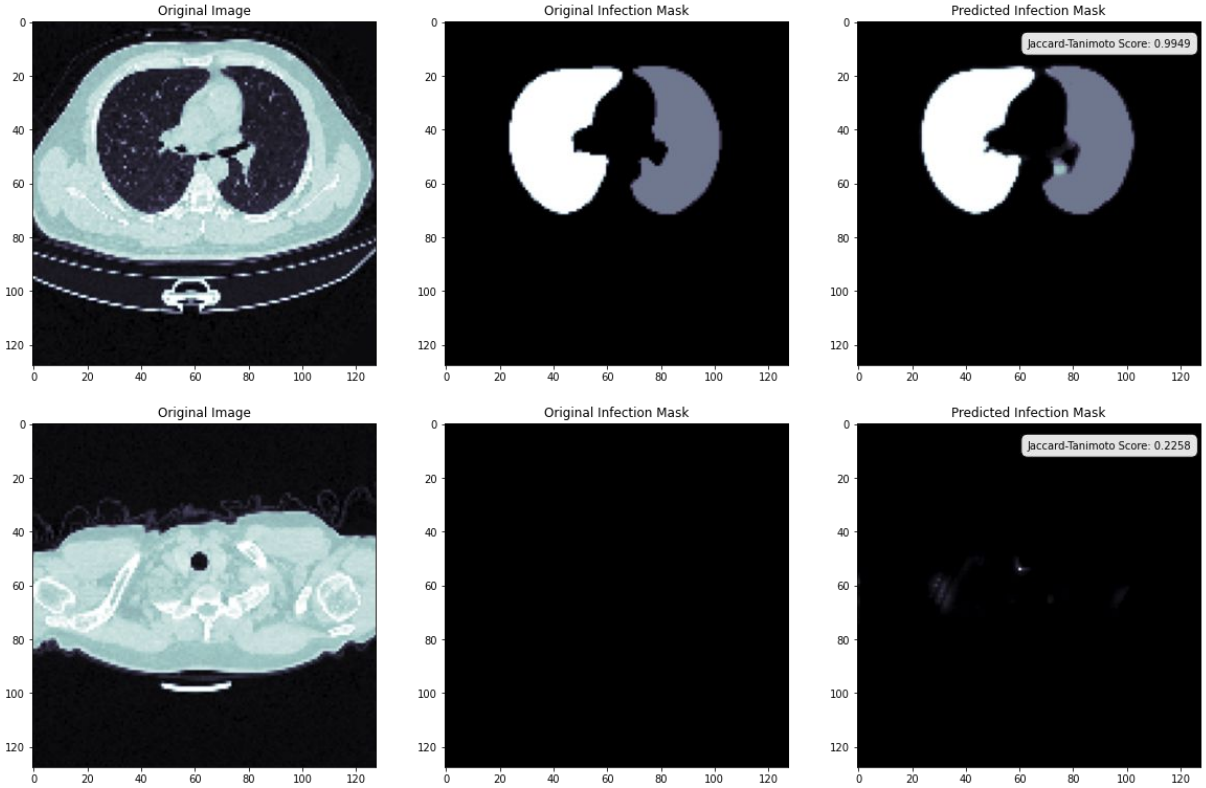




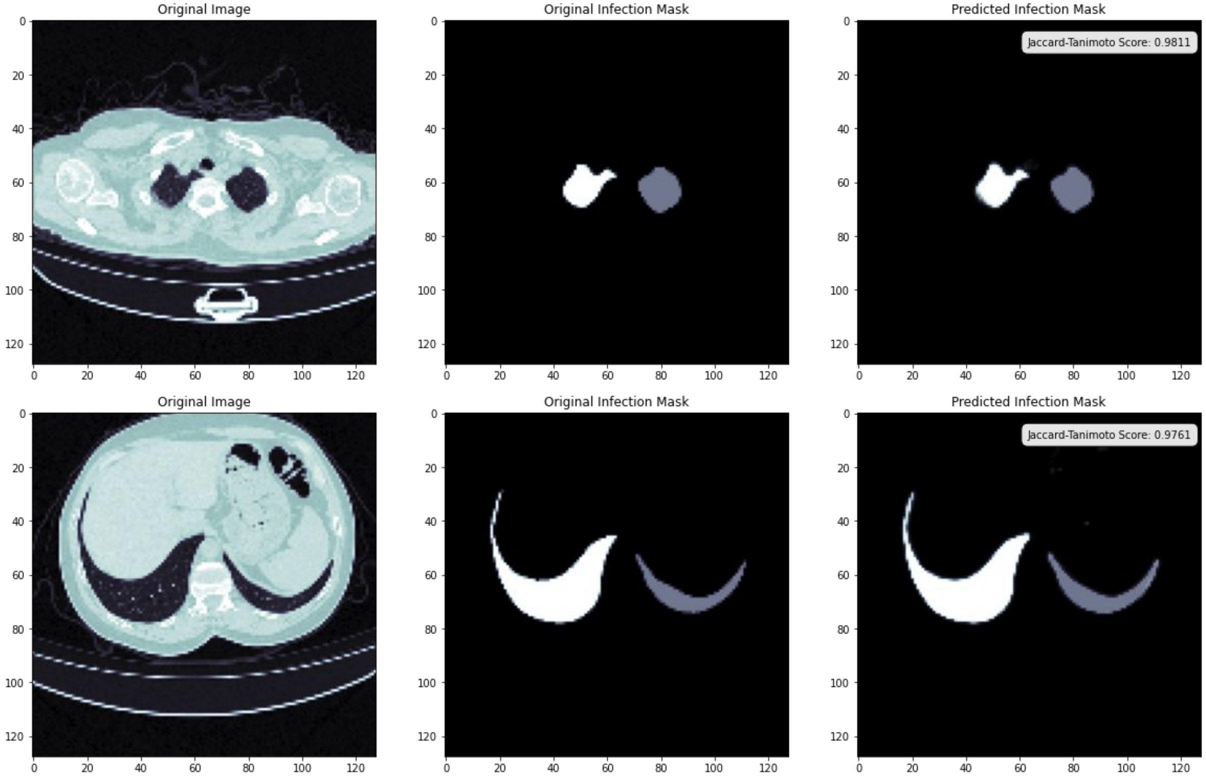
Loss: 0.0376

Jaccard Score: 0.987

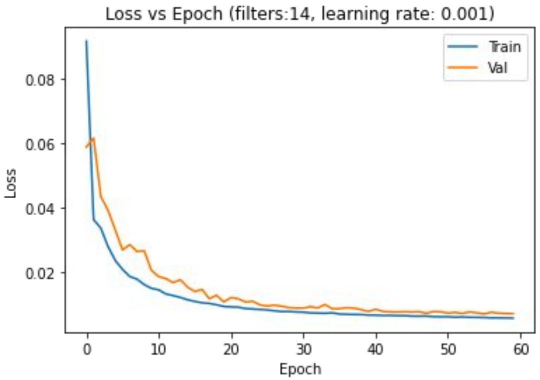
Lung Predictions...

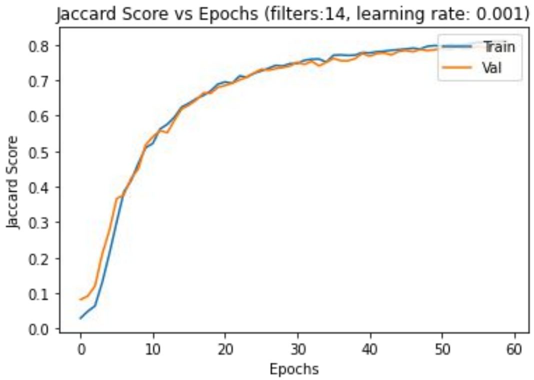


Lung Predictions...



Infection Dataset (dropout: 0.3)

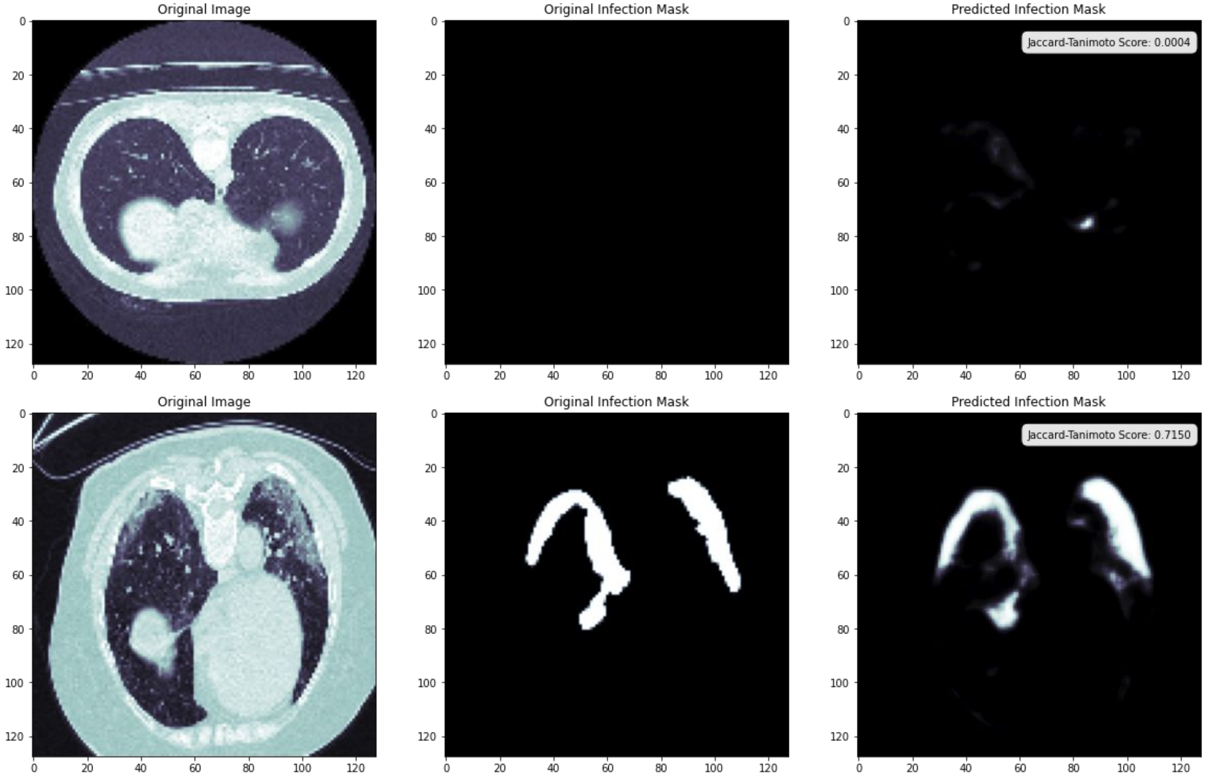




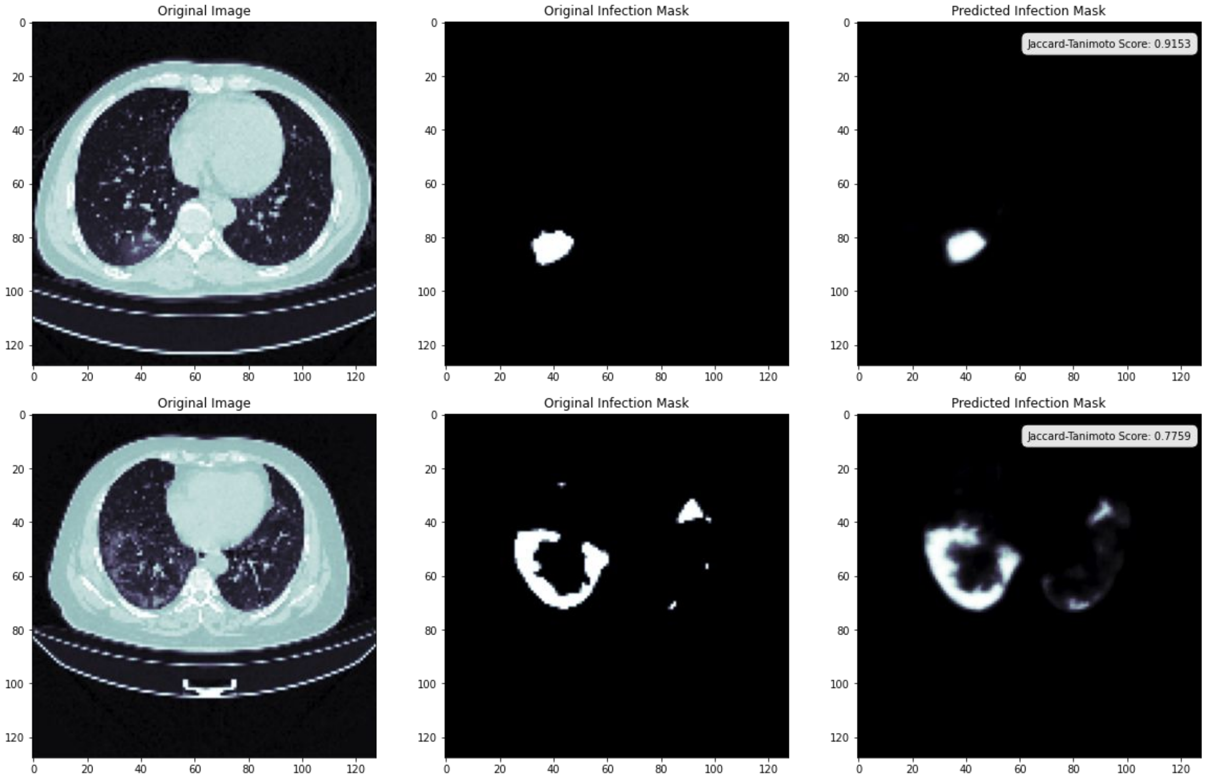
Loss: 0.006

Jaccard Score: 0.809

Infection Predictions...



Infection Predictions...



Comparing Results

5-Fold Cross-Validation from the Proposed Project

5-Fold Cross-Validation				
Segmentation	**Lung**		**Infection**	
Dataset	**Train**	**Validation**	**Train**	**Validation**
Jaccard-Tanimoto	0.9812±0.0009	0.973±0.008	0.719±0.004	0.74±0.02

Results from Other Groups

Table 1: Achieved results showing the median Dice similarity coefficient (DSC), the sensitivity (Sens) and specificity (Spec) on Lung and COVID-19 infection segmentation for each CV fold and the global average (AVG).

	Lungs			**COVID-19**		
Fold	**DSC**	**Sens.**	**Spec.**	**DSC**	**Sens.**	**Spec.**
1	0.907	0.913	0.995	0.556	0.447	0.999
2	0.977	0.979	0.998	0.801	0.875	0.999
3	0.952	0.945	0.999	0.829	0.796	0.999
4	0.979	0.975	0.999	0.853	0.836	0.999
5	0.967	0.967	0.999	0.765	0.697	0.999
AVG	0.956	0.956	0.998	0.761	0.730	0.999

https://arxiv.org/pdf/2007.04774.pdf

Experiment	Dice score	Structure measure	MAE
UNet	0.46	0.77	1.01
SUNet	**0.71**	**0.83**	**0.74**
UNet++	0.73	0.84	0.72
SUNet++	**0.75**	**0.84**	**0.67**
InfNet	0.75	0.85	0.71
SInfNet	**0.77**	**0.85**	**0.63**

Table 1: Segmentation results comparison with baselines. The best performance is obtained using the Symbolic InfNet architecture (SInfNet) with a Dice score of 0.77. The symbolic versions of the architecture show significant improvement in performance over their baseline counterparts.

https://arxiv.org/pdf/2008.09866.pdf

TABLE IV: Quantitative Results of 5-fold cross validation on COVID-19-CT-Seg dataset for Task 1: Learning with limited annotations. For each fold, average DSC and NSD values are reported. The last row shows the average results of 80 (= 5 folds ×16 testing cases per fold) testing cases.

Subtask	Lung				Infection		Lung and Infection Union Segmentation					
	Left Lung		Right Lung				Left Lung		Right Lung		Infection	
	DSC (%)	NSD (%)	DSC (%)	NSD (%)	DSC (%)	NSD (%)	DSC (%)	NSD (%)	DSC (%)	NSD (%)	DSC (%)	NSD (%)
Fold-0	84.9 ± 8.2	68.7 ± 13.3	85.2 ± 13.0	70.6 ± 15.8	68.1 ± 20.5	70.9 ± 21.3	50.5 ± 30.4	36.9 ± 19.6	64.8 ± 18.9	47.1 ± 13.8	66.5 ± 23.4	68.7 ± 22.5
Fold-1	80.3 ± 14.5	61.8 ± 15.1	83.9 ± 9.6	68.3 ± 9.0	71.3 ± 20.5	71.8 ± 23.0	40.3 ± 18.7	27.5 ± 12.0	60.1 ± 11.1	41.7 ± 9.9	64.7 ± 21.8	60.6 ± 25.1
Fold-2	87.1 ± 12.1	74.3 ± 16.0	90.3 ± 8.2	78.5 ± 12.0	66.2 ± 21.7	71.7 ± 24.2	80.3 ± 18.8	66.8 ± 18.8	85.2 ± 12.4	68.6 ± 15.1	60.7 ± 27.6	62.5 ± 28.9
Fold-3	88.4 ± 7.0	75.2 ± 8.8	89.9 ± 6.3	78.5 ± 8.0	68.1 ± 23.1	70.8 ± 27.1	79.7 ± 13.6	65.4 ± 14.4	84.0 ± 9.8	67.7 ± 13.0	62.0 ± 27.9	65.3 ± 28.9
Fold-4	88.3 ± 7.6	75.8 ± 11.0	90.2 ± 7.0	78.3 ± 10.2	62.7 ± 26.9	64.9 ± 28.2	72.4 ± 21.1	58.6 ± 20.8	80.9 ± 13.4	63.4 ± 15.9	51.4 ± 30.2	51.9 ± 31.0
Avg	**85.8 ± 10.5**	**71.2 ± 13.8**	**87.9 ± 9.3**	**74.8 ± 11.9**	**67.3 ± 22.3**	**70.0 ± 24.4**	64.6 ± 26.4	51.1 ± 23.4	75.0 ± 16.8	57.7 ± 17.4	61.0 ± 26.2	61.8 ± 27.4

https://arxiv.org/pdf/2004.12537v2.pdf