

Pneumonia detection by Transfer learning and CNN

Team Member

1 Ruixuan Wang

2 Wengying Wen

3 JiongHui Chen

4 Jinjin Ma

N#

N19997322

N16116736

N19050877

N15491779

Background

- The risk of pneumonia is immense for many, especially in developing nations where people rely on polluting forms of energy.
- The problem can be further aggravated due to the dearth of medical resources and personnel.
- One of the most urgent needs in combatting pneumonia is a cost-effective manner of diagnosing X-ray images.

Abstract

- This project proposes a machine learning model using convolutional neural network to classify and detect the presence of pneumonia from a collection of chest X-ray image samples.
- This model could help mitigate the reliability and interpretability challenges often faced when dealing with medical imagery.
- It is difficult to obtain a large amount of pneumonia dataset for this classification task.
- Therefore, we deployed Structure Correcting Adversarial Network to segment lung fields relied on a pre-trained model to improve the validation and classification accuracy of the CNN model and achieved remarkable validation accuracy.

GANs-SCAN

- In medical field, always lacking of labeled training data. In this context, generative modeling can still perform well. The adversarial training concept behind GANs theoretically eliminates the need to model explicit pixelwise objective functions by learning a rich similarity metric to tell real and fake data apart. This allows to optimize for concepts in images beyond the pixel-level, leading to more realistic results.
- Dai[1] et al. propose a GAN based solution (SCAN) to enhance global consistency of segmentation and extract contours of the heart and left/right lungs. It is a fully connected network with a VGG down-sampling path and residual blocks to use much fewer feature maps in the generator. And we use this as our preprocessing step.

Preprocess

• Data Augmentation

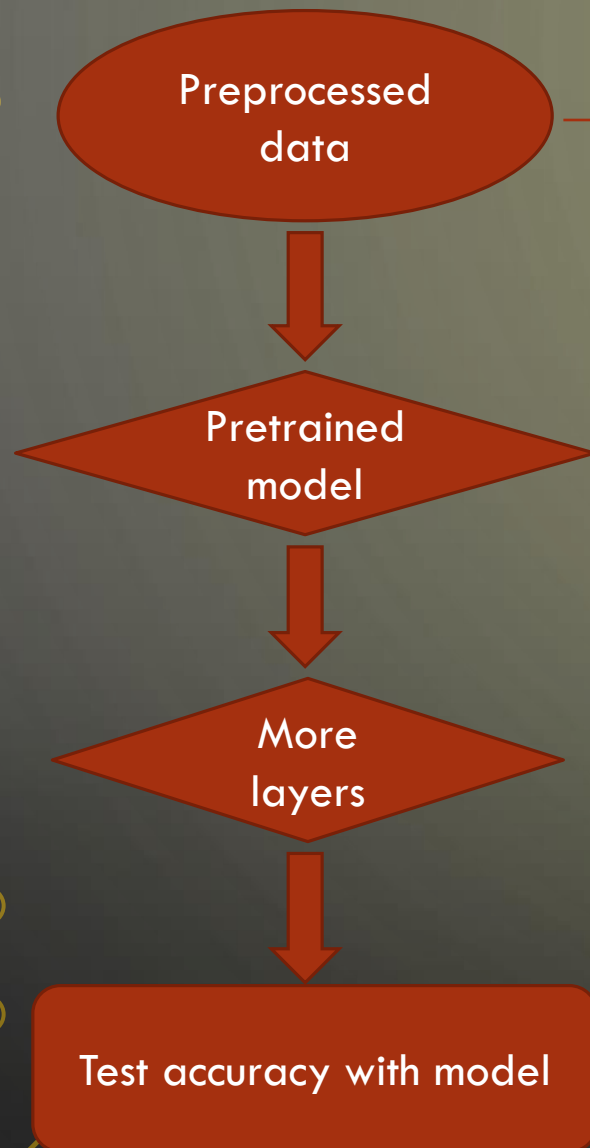
- Due to lack of data
- Increase sample size
- We can expand our dataset three times by rotation.

• Graph process

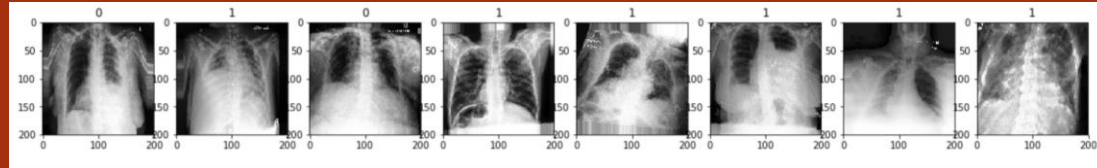
- Decrease unnecessary factors on graphs
- Increase emphasis on lung



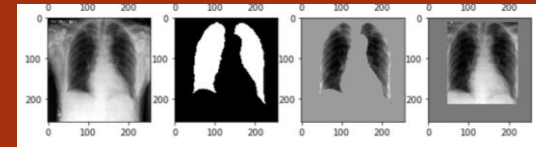
Construction



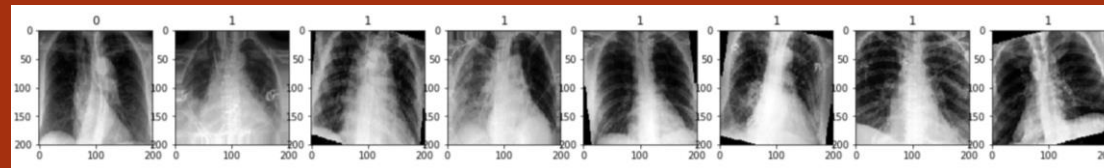
- Original Data



- Segmentation

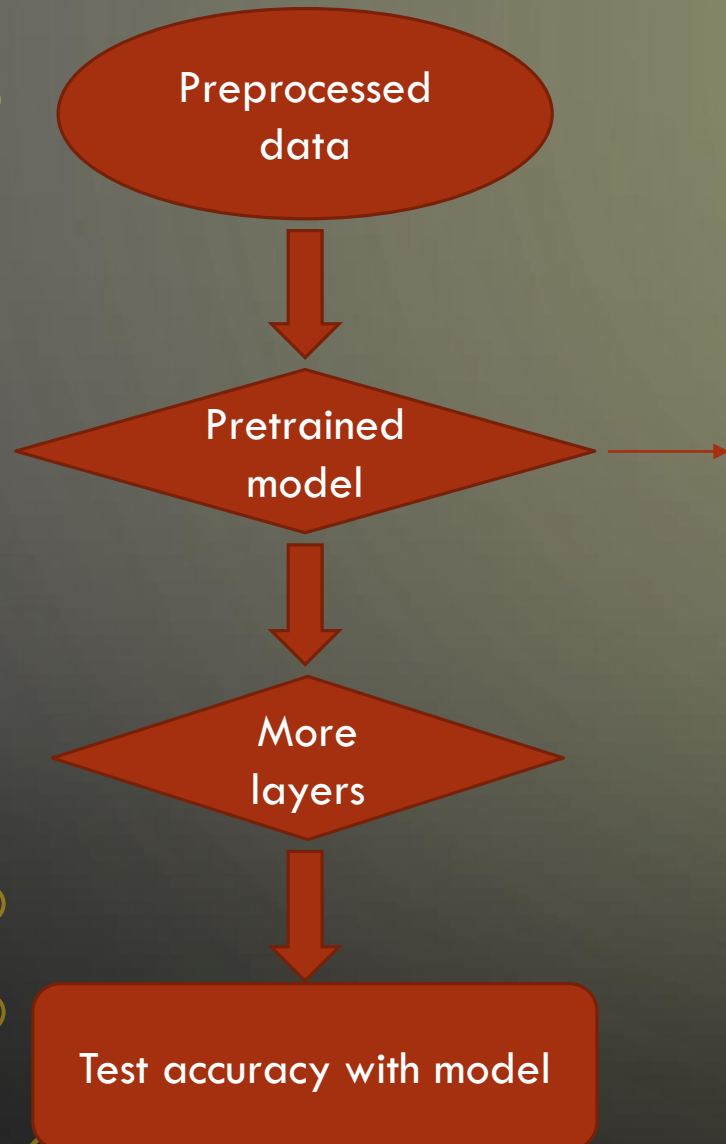


- Augmentation



- Data processed

Construction



```
1 model.summary()

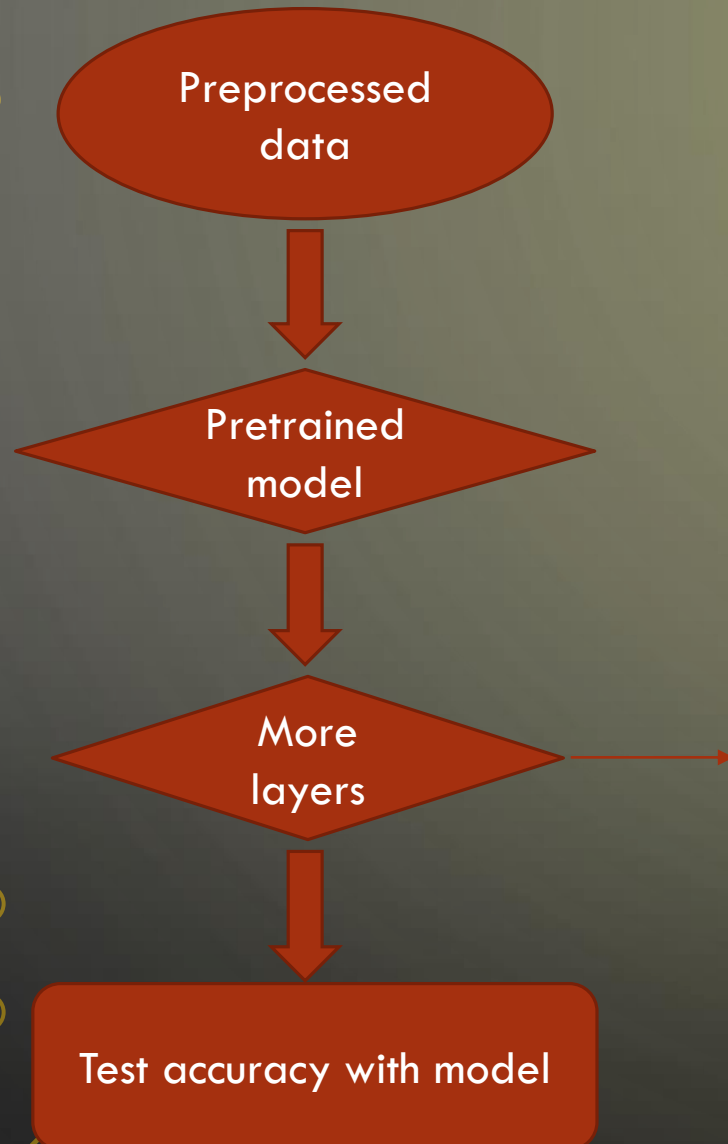
Layer (type)                 Output Shape                  Param #
-----
input_1 (InputLayer)         (None, 200, 200, 3)          0
block1_conv1 (Conv2D)         (None, 200, 200, 64)         1792
block1_conv2 (Conv2D)         (None, 200, 200, 64)         36928
block1_pool (MaxPooling2D)    (None, 100, 100, 64)         0
block2_conv1 (Conv2D)         (None, 100, 100, 128)        73856
block2_conv2 (Conv2D)         (None, 100, 100, 128)        147584
block2_pool (MaxPooling2D)    (None, 50, 50, 128)          0
block3_conv1 (Conv2D)         (None, 50, 50, 256)          295168
block3_conv2 (Conv2D)         (None, 50, 50, 256)          590880
block3_conv3 (Conv2D)         (None, 50, 50, 256)          590880
block3_conv4 (Conv2D)         (None, 50, 50, 256)          590880
block3_pool (MaxPooling2D)    (None, 25, 25, 256)          0
block4_conv1 (Conv2D)         (None, 25, 25, 512)          1180160
block4_conv2 (Conv2D)         (None, 25, 25, 512)          2359808
block4_conv3 (Conv2D)         (None, 25, 25, 512)          2359808
block4_conv4 (Conv2D)         (None, 25, 25, 512)          2359808
block4_pool (MaxPooling2D)    (None, 12, 12, 512)          0
block5_conv1 (Conv2D)         (None, 12, 12, 512)          2359808
block5_conv2 (Conv2D)         (None, 12, 12, 512)          2359808
block5_conv3 (Conv2D)         (None, 12, 12, 512)          2359808
block5_conv4 (Conv2D)         (None, 12, 12, 512)          2359808
block5_pool (MaxPooling2D)    (None, 6, 6, 512)            0
flatten_1 (Flatten)           (None, 18432)                 0
dense_1 (Dense)               (None, 64)                    1179712
dropout_1 (Dropout)           (None, 64)                    0
dense_2 (Dense)               (None, 1)                     65
Total params: 21,204,161
Trainable params: 1,179,777
Non-trainable params: 20,024,384
```

Model	T-1 ACC	T-5 ACC	Depth
VGG16	0.715	0.901	23
VGG19	0.727	0.910	26

Compared with VGG16:

- Higher accuracy
- More depth and parameters

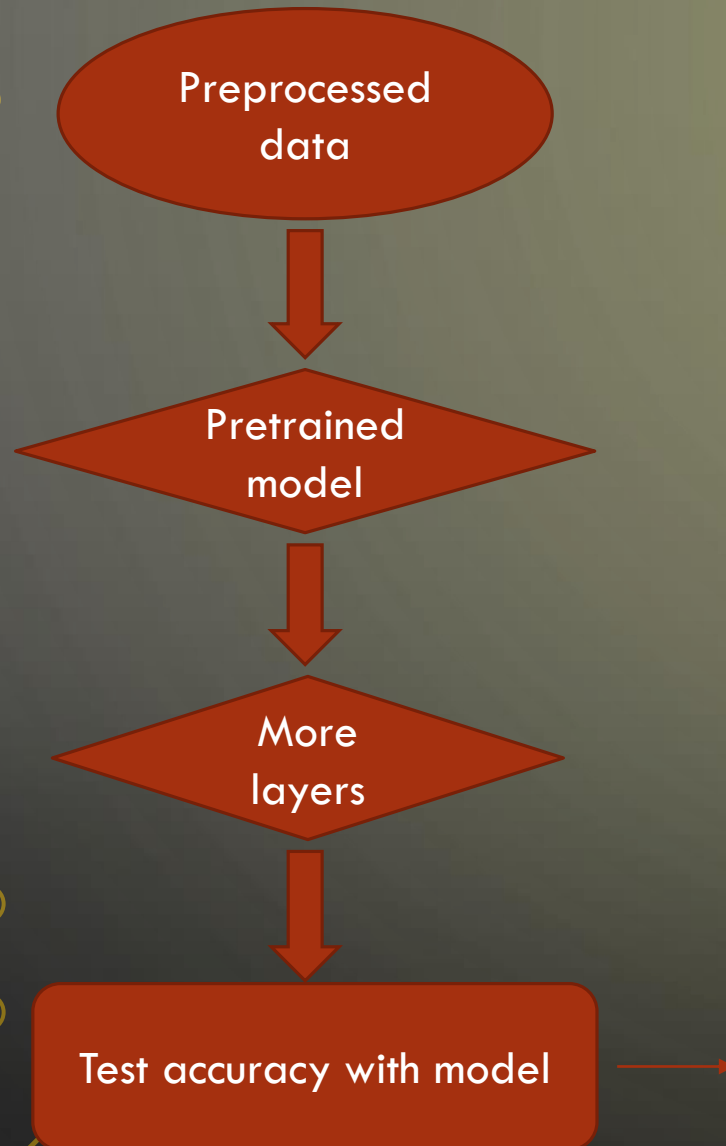
Construction



- Flatten layer:
Transfer the high dimension data calculated from convolution layer in to one dimension to fit the format in full connected layer.
- Dense layer:
Set 64 hidden units to simulate various features of the target, using activation function 'relu' and 'sigmoid'.
- Dropout layer:
In this case we drop 90% percent of coefficients to avoid overfitting.

```
for layer in model.layers:  
    layer.trainable = False  
model.add(Flatten())  
model.add(Dense(64,activation='relu'))  
model.add(Dropout(0.1))  
model.add(Dense(1,activation='sigmoid'))
```


Construction



- Totally, after data augmentation, We get
- 50031 images for training
- 12516 images for testing.
- Both in two classes
- For parameter selection, we can build a GridSearch function to get the best fitting parameter in this problem.
- We set three learning rates, three decays and three number of epochs for this grid searching.

```
opt = optimizers.RMSprop(lr = lrt, decay = dcy)
model.compile(optimizer=opt, loss='binary_crossentropy', metrics=['accuracy'])
hist = model.fit_generator(train_generator, steps_per_epoch = steps_per_epoch, epochs = npc,
                           validation_data = test_generator, validation_steps = validation_steps)
Matrix_acc[i,j,k] = np.mean(hist.history['acc'])
Matrix_valacc[i,j,k] = np.mean(hist.history['val_acc'])
```

Parameters

Epoch=5		Learning rate		
Decay	1e-3	77.71%	77.91%	77.73%
	1e-4	77.78%	79.03%	78.19%
	1e-5	75.63%	77.78%	78.68%

Epoch=10		Learning rate		
Decay	1e-3	77.37%	77.17%	77.43%
	1e-4	77.51%	76.08%	76.73%
	1e-5	78.71%	78.36%	76.36%

Epoch=15		Learning rate		
Decay	1e-3	77.86%	77.45%	77.26%
	1e-4	78.08%	77.50%	77.81%
	1e-5	77.17%	77.28%	77.45%

Evaluation

- After grid searching, we use the group of parameter with the highest accuracy.
- Learning rate = 0.0001, decay = 0.0001 and Epochs = 5
- Then we use F1-score to evaluate this model.
- With an average F1-score of 0.8729 and average val-F1-score of 0.08773 , our model exceeds average radiologist performance on the F1 metric.

More thoughts

- The main challenge in medical imaging often classification or segmentation is presented as a binary task: normal versus abnormal and object versus background. The problem is often a gross simplification as both classes can be highly heterogeneous
- Another data-related challenge is class imbalance. In medical imaging, images for the abnormal class might be challenging to find. In medical image analysis useful information is not just contained within the images themselves. Physicians often leverage a wealth of data on patient history, age, demographics and others to arrive at better decisions.
- Finally, deep learning methods have often been described as 'black boxes'. Especially in medicine, where accountability is important and can have serious legal consequences, it is often not enough to have a good prediction system.

Reference

- Dai W, Dong N, Wang Z, et al. SCAN: Structure correcting adversarial network for organ segmentation in chest X-rays[M]//Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support. Springer, Cham, 2018: 263-273.
- Litjens G, Kooi T, Bejnordi B E, et al. A survey on deep learning in medical image analysis[J]. Medical image analysis, 2017, 42: 60-88.
- Stephen O, Sain M, Maduh U J, et al. An Efficient Deep Learning Approach to Pneumonia Classification in Healthcare[J]. Journal of Healthcare Engineering, 2019.