Segmentation of lung field in Chest X-Rays

CXR segmentation or Chest X-Rays segmentation is a vital step in understanding lungs field view and diagnosis by the use of deep learning. CXR has mainly grayscale data and thus can be thresholded easily for segmentation. However, thresholding is not an solid solution for segmentation when we require to diagnosis based on important parameters such as ***cardiothoracic ratio*** **(CTR).** Semantic segmentation is used here to segment lung fields.

In semantic segmentation, each pixel is being labelled according to the classes. I have used U-Net which is Convolutional layer based semantic segmentation architecture developed by Olaf Ronneberger et al. for Bio Medical Image Segmentation.

The U-Net Architecture

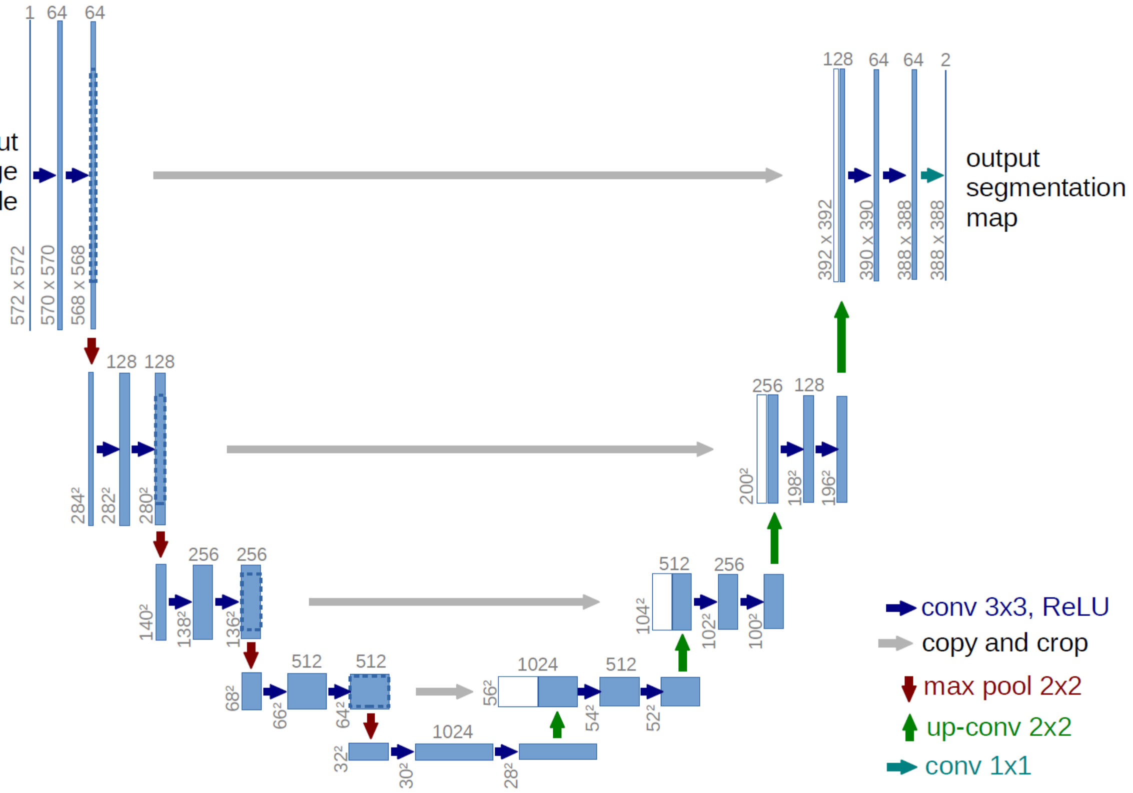


Fig 1. Sources: arXiv:1505.04597v1

The path on the left side are the contraction paths, also called Encoders, which are only a stack of convolutional and max-polling layers. The right path is the symmetric expanding path (or Decoders) which is used to enable precise localization using transposed convolutions. These transposed convolutions layers are also called deconvolutional layers and the process is called Up-sampling. Thus, it is an end-to-end fully convolutional networks (FCN) and does not contain any dense layers.

Training

I have used keras to train the model using the above architecture.

Input size of image: 256x256

Model is compiled with Adam with a learning rate of 0.0001 and used loss as binary_crossentropy with metrics set as accuracy.

Callbacks is used for these purposes:

a) Early stopping if the validation loss does not improve for 5 continues epochs.

b) Save the weights only if there is improvement in validation loss.

Used a batch size of 4.

Trained on local GPU with train accuracy of 0.9876 and validation accuracy of 0.9798.

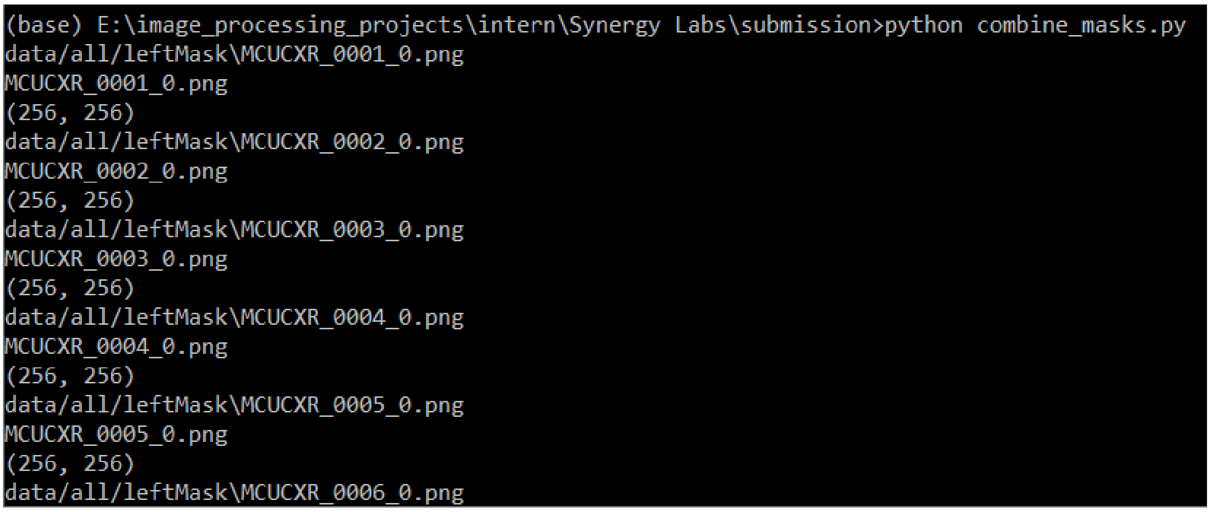
Description of Files

Dataset

Dataset used is **MontgomerySet** which included chest x-rays and its ground truth masks – left and right.

combine_masks.py

The first job is to combine left and right lung masks present in the respective folders in data/all. This script is used to add the left and right mask and write the combined mask in the folder data/all/combined_mask.



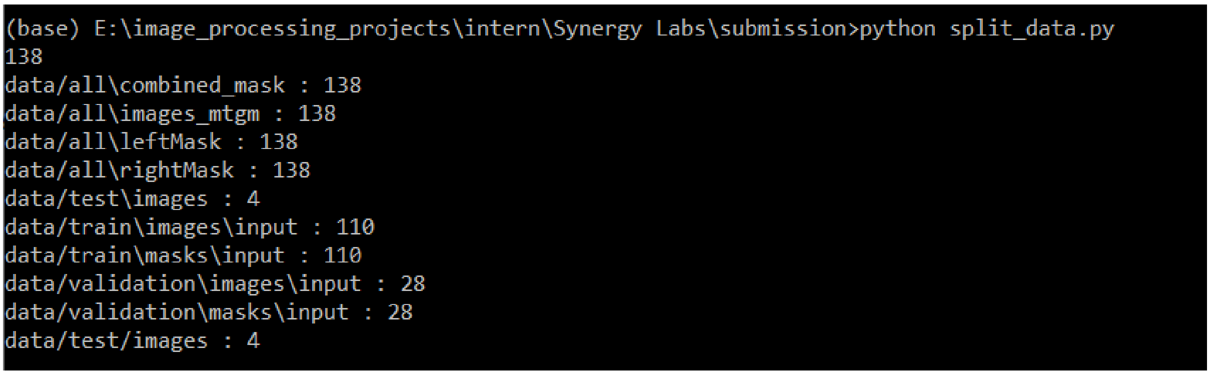
```
(base) E:\image_processing_projects\intern\Synergy Labs\submission>python combine_masks.py
data/all/leftMask\MCUCXR_0001_0.png
MCUCXR_0001_0.png
(256, 256)
data/all/leftMask\MCUCXR_0002_0.png
MCUCXR_0002_0.png
(256, 256)
data/all/leftMask\MCUCXR_0003_0.png
MCUCXR_0003_0.png
(256, 256)
data/all/leftMask\MCUCXR_0004_0.png
MCUCXR_0004_0.png
(256, 256)
data/all/leftMask\MCUCXR_0005_0.png
MCUCXR_0005_0.png
(256, 256)
data/all/leftMask\MCUCXR_0006_0.png
```

split_data.py

The purpose of this script is to split the x-ray images and its corresponding masks (combined) into

- Training set - created folder train/images and train/masks
- Validation set – created folder validation/images, validation/mask

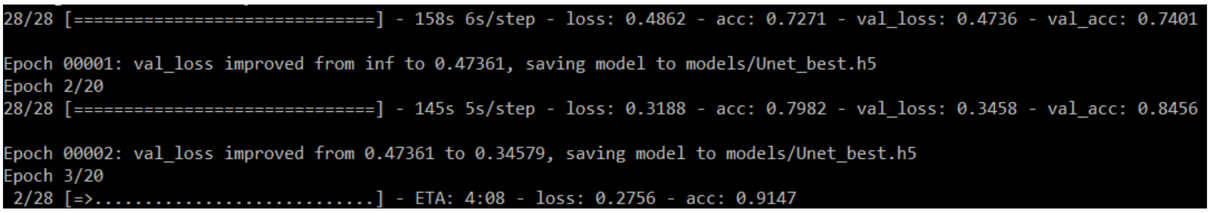
The folder test/images is created manually and random images are added



```
(base) E:\image_processing_projects\intern\Synergy Labs\submission>python split_data.py
138
data/all\combined_mask : 138
data/all\images_mtgm : 138
data/all\leftMask : 138
data/all\rightMask : 138
data/test\images : 4
data/train\images\input : 110
data/train\masks\input : 110
data/validation\images\input : 28
data/validation\masks\input : 28
data/test/images : 4
```

train.py

This is the main training and image generating script. Under the class Unet, functions defining the architecture, callbacks, data augmentation, model compilation, generating predictions and saving it, generation of masks are written.

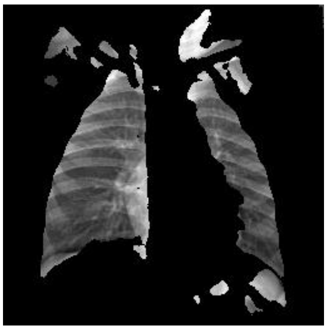


```
28/28 [==============================] - 158s 6s/step - loss: 0.4862 - acc: 0.7271 - val_loss: 0.4736 - val_acc: 0.7401

Epoch 00001: val_loss improved from inf to 0.47361, saving model to models/Unet_best.h5
Epoch 2/20
28/28 [==============================] - 145s 5s/step - loss: 0.3188 - acc: 0.7982 - val_loss: 0.3458 - val_acc: 0.8456

Epoch 00002: val_loss improved from 0.47361 to 0.34579, saving model to models/Unet_best.h5
Epoch 3/20
 2/28 [=>............................] - ETA: 4:08 - loss: 0.2756 - acc: 0.9147
```

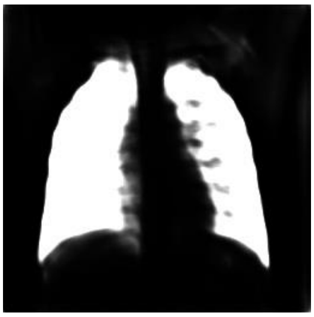
The predicted masks are saved in .npy format and the trained model in .h5 format.

There are three types of outputs that can be generated:

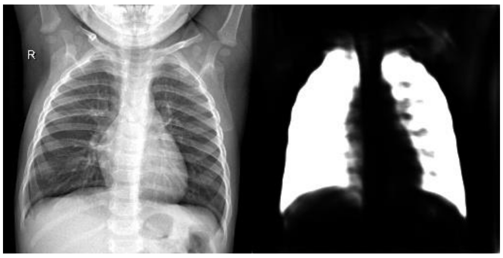
- **Cropped view** – the original images are cropped according to the mask predicted



- **Only masks** - outputs a binary image of mask



- **Side-by-side** – the original x-ray and the binary mask



Contour_output.py

This script draw the contours of the lung field on the original x-ray image by using its corresponding mask. The mask only output is thresholded and then contours are drawn. Since the contours of the lungs field are the two largest contour by area, the left and right, so the two largest contours are considered and drawn on the original image. The original image was resized to 256x256 which is same as the size of the generated mask.