Capstone Project Proposal

Pneumothorax Segmentation in Chest X-ray Images

A pneumothorax is a collapsed lung. It occurs when air leaks into the space between the lung and the chest wall. This air pushes on the outside of the lung and makes it either partially or completely collapse – Fig.1 [1].

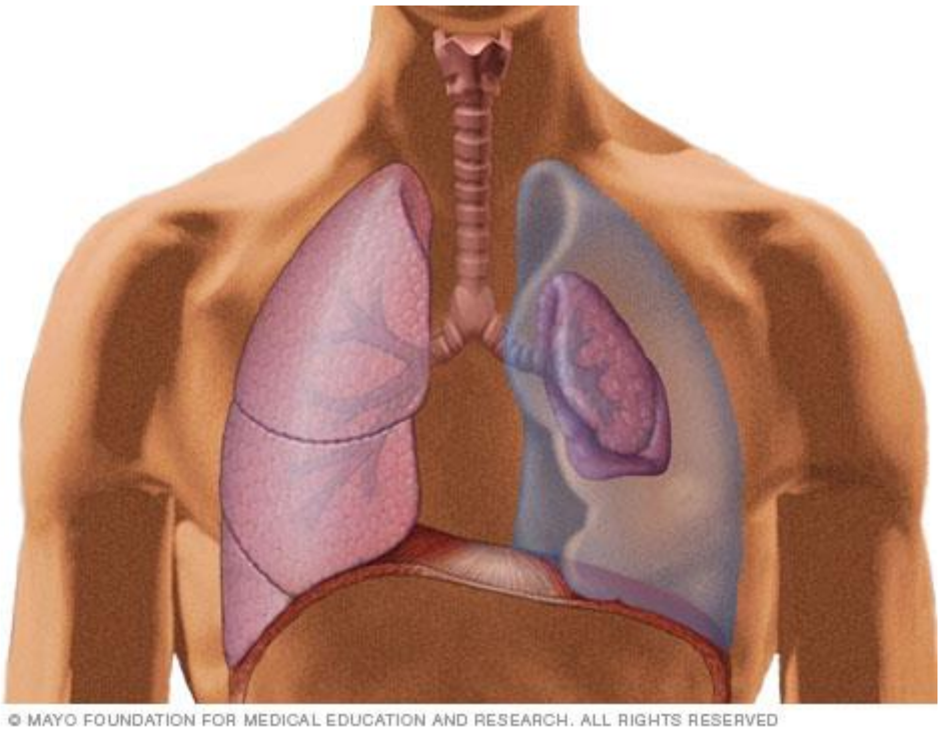

Fig.1: Pneumothorax: collapsed vs. normal lung. Adopted from [1]

The pneumothorax can be caused by a blunt or penetrating chest injury, certain medical procedures, damage from underlying lung disease, or most horrifying - it may occur for no obvious reason at all. The main symptoms usually include sudden chest pain and shortness of breath. On some occasions, a collapsed lung can be a life-threatening event. These symptoms can be caused by a variety of health problems, and pneumothorax is usually diagnosed by a radiologist on a chest x-ray, and can sometimes be very difficult to confirm.

Medical imaging is a rapidly evolving field due to large breakthroughs in deep learning techniques, which emerged in the last ten years. An accurate AI algorithm to detect pneumothorax would be useful in a lot of clinical scenarios. AI could be used to triage chest radiographs for priority interpretation, or to provide a more confident diagnosis for non-radiologists. Treatment for a pneumothorax usually involves inserting a needle or chest tube between the ribs to remove the excess air. However, a small pneumothorax may heal on its own.

The Society for Imaging Informatics in Medicine, American College of Radiology (SIIM-ACR) and Kaggle organized a competition challenge in the 2nd half of 2019 to develop a model to classify (and if present, segment) pneumothorax from a set of chest radiographic images in order to aid in the early recognition of pneumothoraxes and save lives [2].

The original data set for this challenge was pretty large and was in the DICOM format, which is standard in medical imaging. Although the raw DICOM data is no longer available, the images converted to PNG format are available on Kaggle in different resolution [3]. The training set consists of images and corresponding binary masks, where pneumothorax is encoded. About 22 percent of the images contain pneumothorax.

The goal of this challenge is to train a model using the training data and to make predictions on the test set. The predictions can be scored by uploading them to the Kaggle website. Late submissions are still available for this competition, even though it has already finished. My goal is to develop a model, which will allow to place in the top 10% of the leaderboard (in the medals), with a stretch goal to get in the top 3% on the Kaggle leaderboard [4]. This will ensure that the developed model is on par with some of the best approaches developed during the competition.

The competition is evaluated based on the mean Dice coefficient. The Dice coefficient is used to compare the pixel-wise agreement between a predicted segmentation and its corresponding ground truth using the following formula:

$$\frac{2 * |X \cap Y|}{|X| + |Y|},$$

where X is the predicted set of pixels and Y is the ground truth. The Dice coefficient is defined to be 1 when both X and Y are empty. The leaderboard score is the mean of the Dice coefficients for each image in the test set.

I am planning to use a Unet encoder-decoder architecture to build a model [5]. The architecture consists of a contracting path of the encoder to capture context and a symmetric expanding path of the decoder that enables precise localization – Fig.2. In order to further improve the localization high resolution features from the contracting path are combined with the upsampled path via cross-connections.

Pre-trained Unet models with different encoders are publicly available, e.g. in [6]. The plan is to explore different pre-trained (on ImageNet) encoders including, for example, ResNet34 and Efficient Net using both Pytotch and Keras frameworks. The final model is likely to be an ensemble of different models for improved accuracy.

Some of the other approaches that will be explored to improve model accuracy include:

- Use input image augmentations, e.g. Albumentations library [7], as well as test time augmentations (TTA), like horizontal flip.
- Explore different loss functions like focal loss and dice loss.

- Consider different learning rate schedules like cosine annealing, etc. and batch sizes for faster conversion.
- Split data into different folds, e.g. 5 folds, to improve model accuracy and to avoid overfitting.
- Since Unet is a fully-convolutional model, one can first train the model on smaller images and then progressively increase image size for faster conversion.

I have two 1080Ti Nvidia GPUs available on my home computer in order to successfully train the models. As a personal goal I'd like to write a clean python code using module package and object-oriented programming.

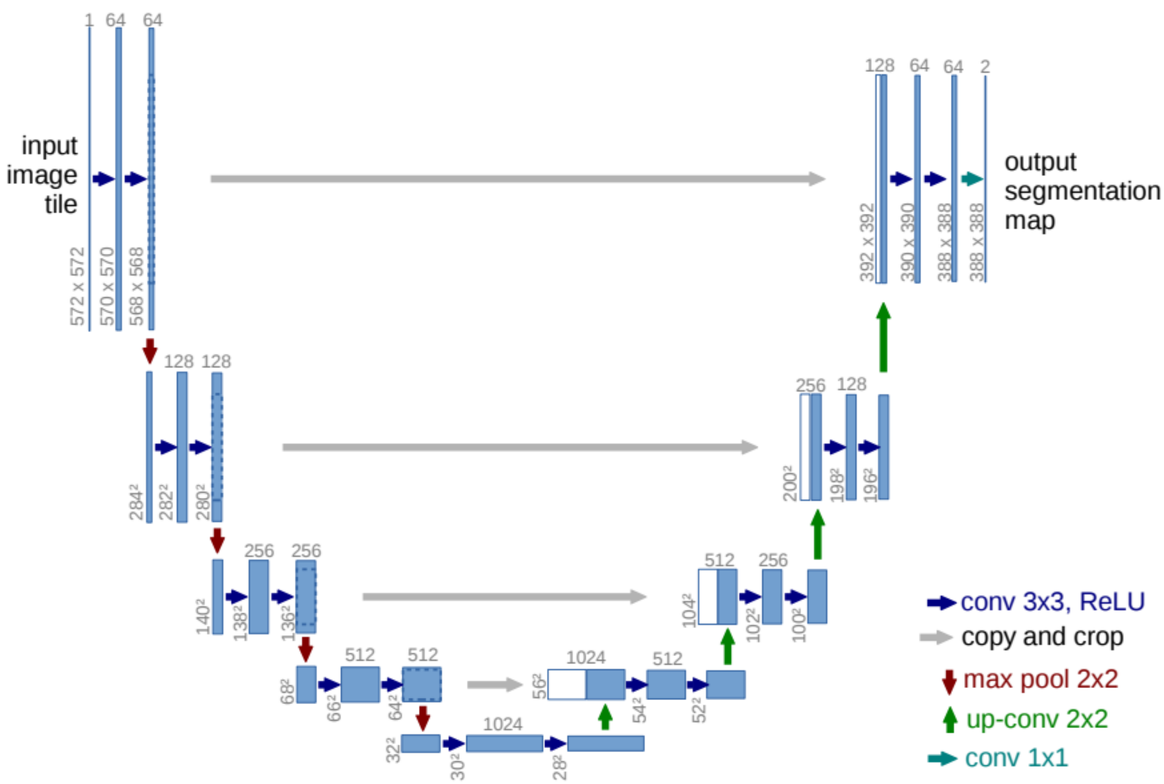

Fig. 2: U-net architecture (example for 32x32 pixels in the lowest resolution). Adopted from [5].

References:

[1] Pneumothorax – Mayo Clinic: https://www.mayoclinic.org/diseases-conditions/pneumothorax/symptoms-causes/syc-20350367

[2] www.kaggle.com/c/siim-acr-pneumothorax-segmentation

[3] https://www.kaggle.com/iafoss/siimacr-pneumothorax-segmentation-data-128

https://www.kaggle.com/iafoss/siimacr-pneumothorax-segmentation-data-256

https://www.kaggle.com/iafoss/siimacr-pneumothorax-segmentation-data-512

https://www.kaggle.com/iafoss/siimacr-pneumothorax-segmentation-data-1024

[4] https://www.kaggle.com/c/siim-acr-pneumothorax-segmentation/leaderboard

[5] Olaf Ronneberger, Philipp Fischer, Thomas Brox, U-Net: Convolutional Networks for Biomedical Image Segmentation, https://arxiv.org/abs/1505.04597

[6] Pavel Yakubovskiy, Segmentation Models for Keras and Pytorch, Github: https://github.com/qubvel/segmentation_models and https://github.com/qubvel/segmentation_models.pytorch

[7] Albumentations: Fast and Flexible Image Augmentations: https://github.com/albumentations-team/albumentations