TERM PROJECT

ED6001 – MEDICAL IMAGE ANALYSIS

Lungs Segmentation from CXR (Chest X-Ray) Scans

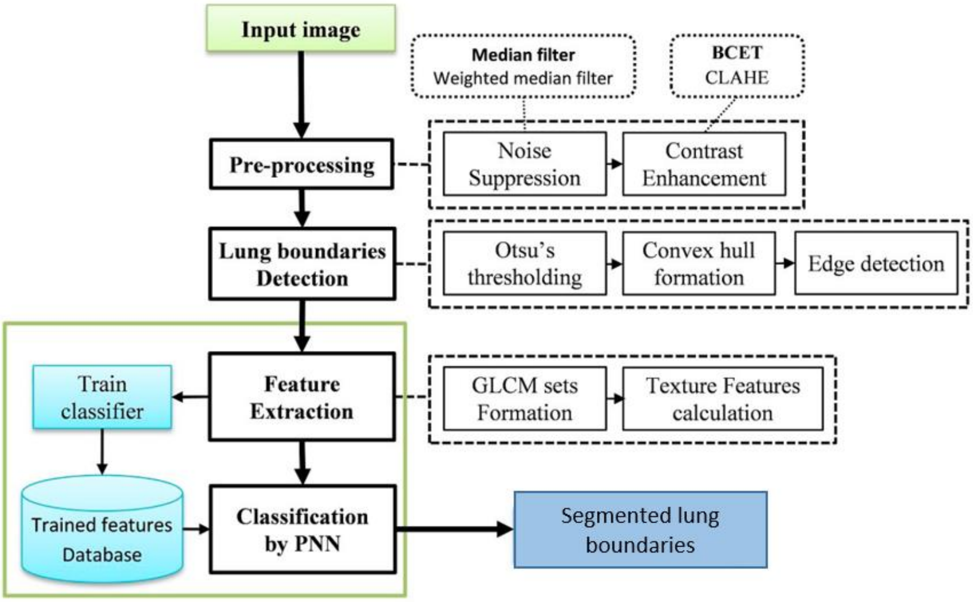
ED17B022

RAZEEM AHMAD

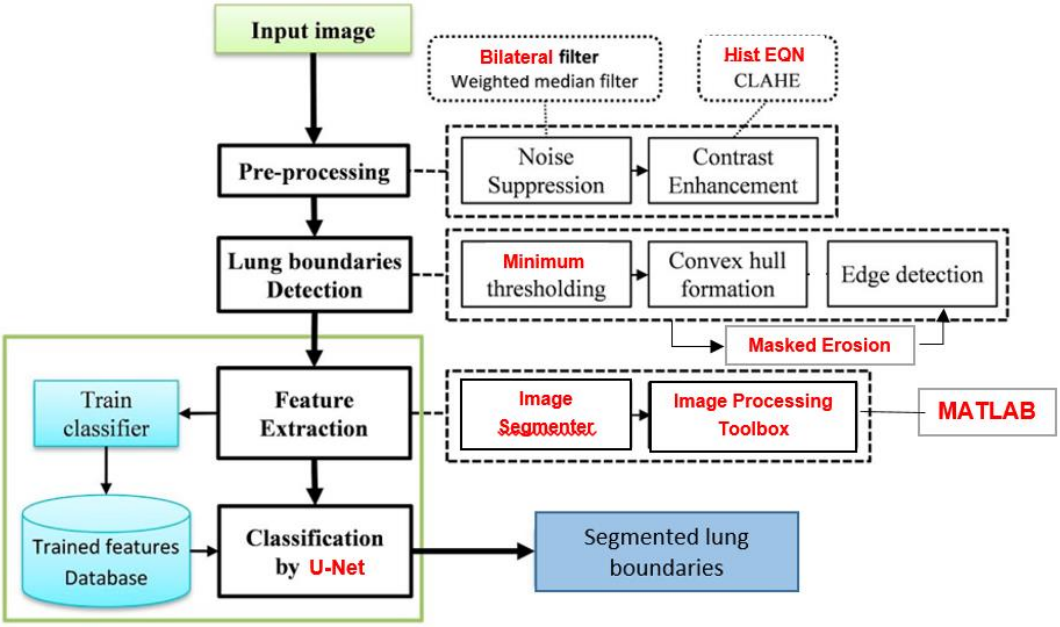
Introduction

Comparing algorithm followed with research paper

- Algorithmic scheme – Research Paper



- Algorithmic scheme – Proposed alternative methods

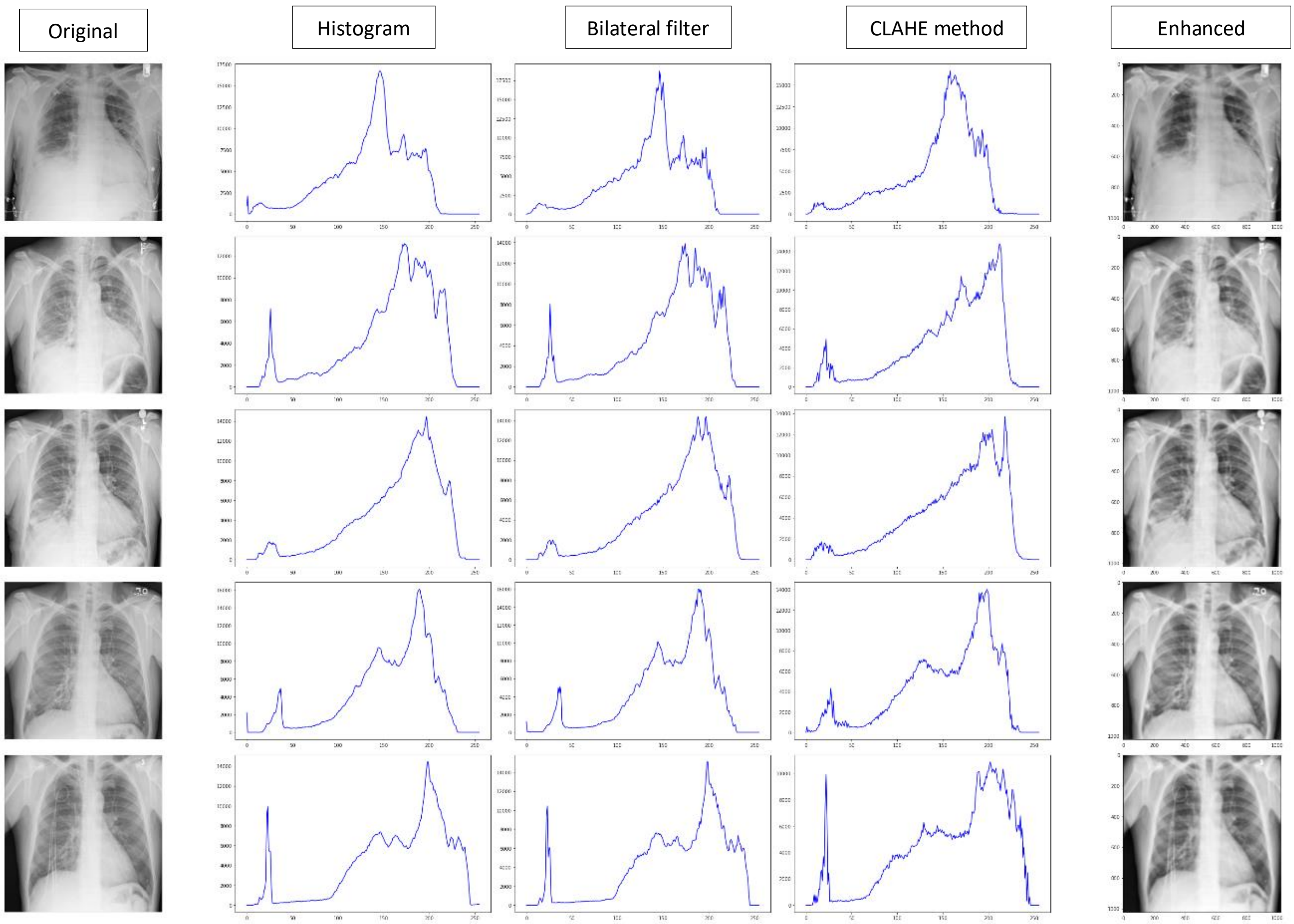


- **The research paper methods have been implemented along with (not separately) the proposed alternative methods so as to easily compare the outputs.**
- **GLCM and PNN could not be implemented on the current system due to lack of good computational power (GPU), even on Google Collab.**
- **Dataset used in the research paper as well as the proposed methods is the CXR scans provided by the NIH clinical center.** (~30k images, of which **random 762** scans, including those in assignment 4, have been used in this implementation, due to memory issues.)

Part 1: Preprocessing & Labelling

Noise Removal + Contrast Enhancement

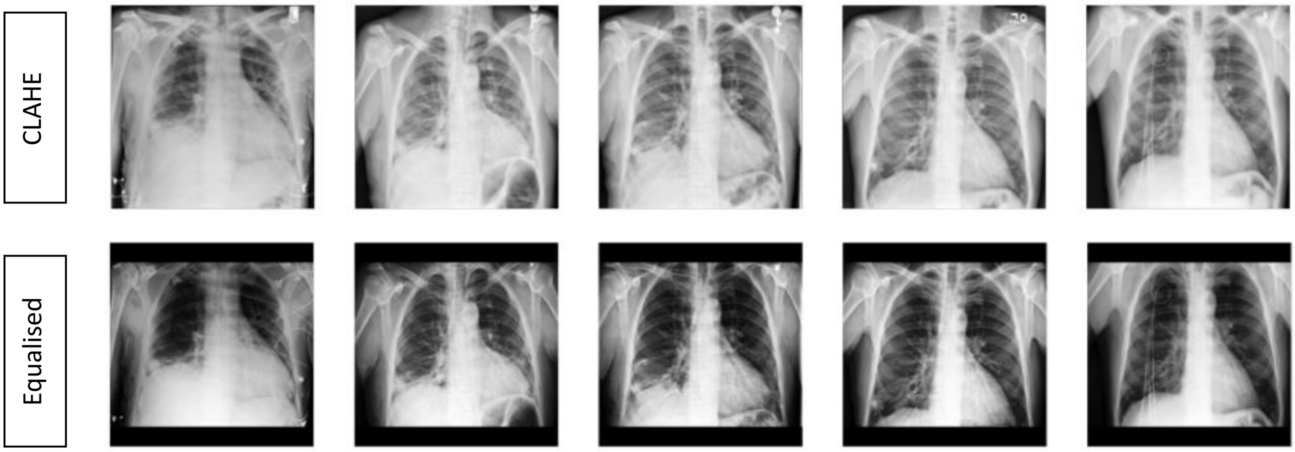
- Histograms of the images were observed to check for any noise (if, exists).
- Bilateral filter was used to denoise the images although most scans were fairly clean.
- To improve contrast of the lungs from the ribcage and the heart, CLAHE technique was used to enhance the image.



PSNR				
Image 1	Image 2	Image 3	Image 4	Image 5
30.763614	30.075971	29.869903	30.471523	30.443774

Observations:

- Most of the scans as observed from the profiles of their histograms were not noisy. But a majority of them had very low contrast.
- Good contrast is crucial to segment out the lungs from the scans as ground truth.
- As observed, even the ‘enhanced’ image using CLAHE did not have a stark difference between the original in terms of contrast.
- **Hence to improve contrast of the lungs from heart, ribcage and the rest of the body, direct histogram equalization is applied on the CLAHE enhanced scans.**



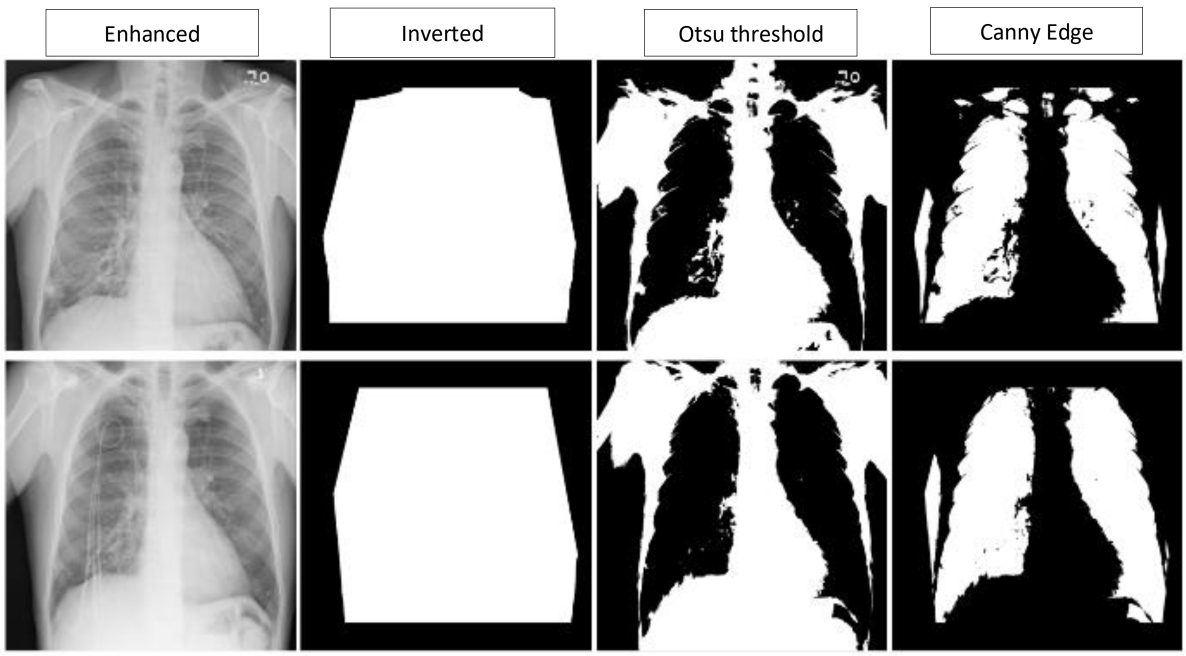
The top and bottom edges were trimmed, this would help us segment the ROI easily.
It can be observed that the linings of the lungs and heart have better contrast than the original

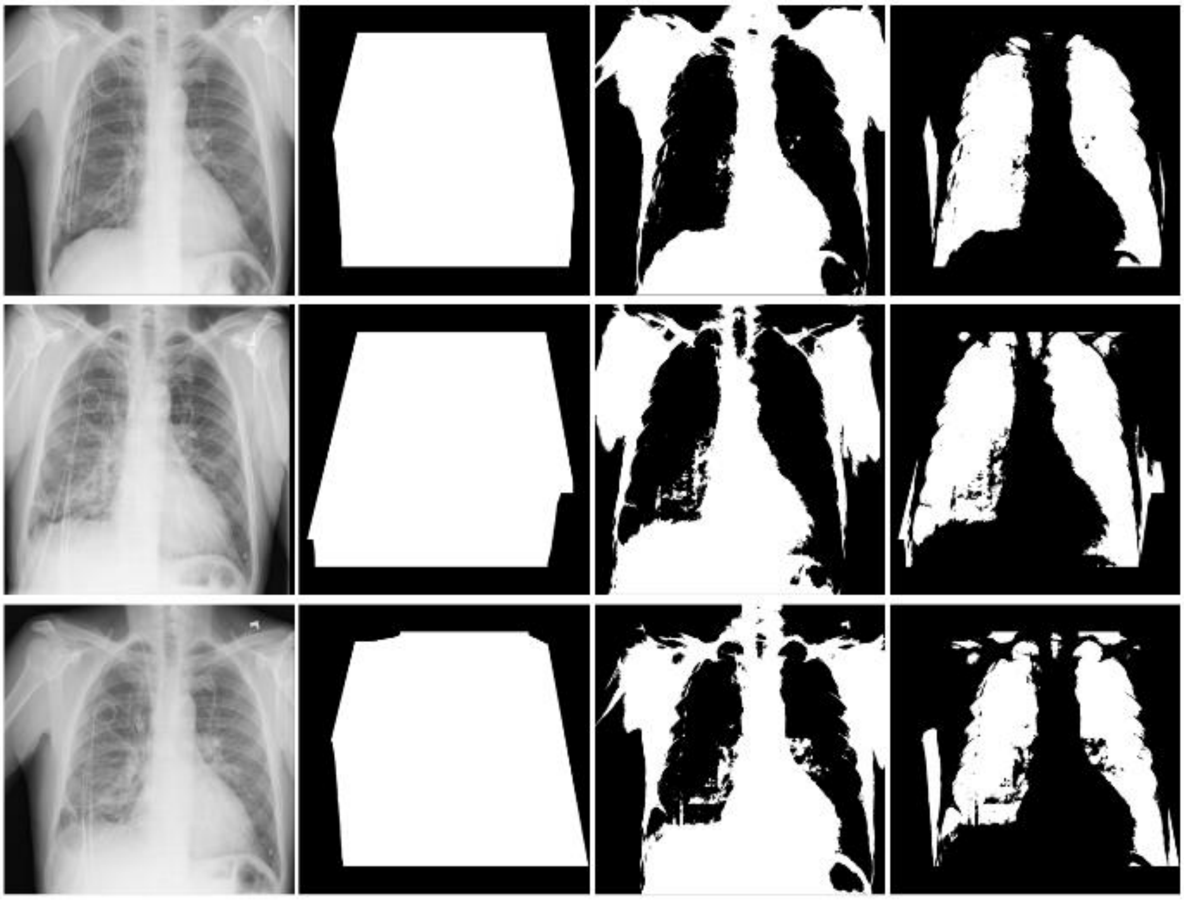
PSNR				
Image 1	Image 2	Image 3	Image 4	Image 5
37.909849	37.910059	37.873035	37.762828	37.860376

ROI retrieval and Edge Detection

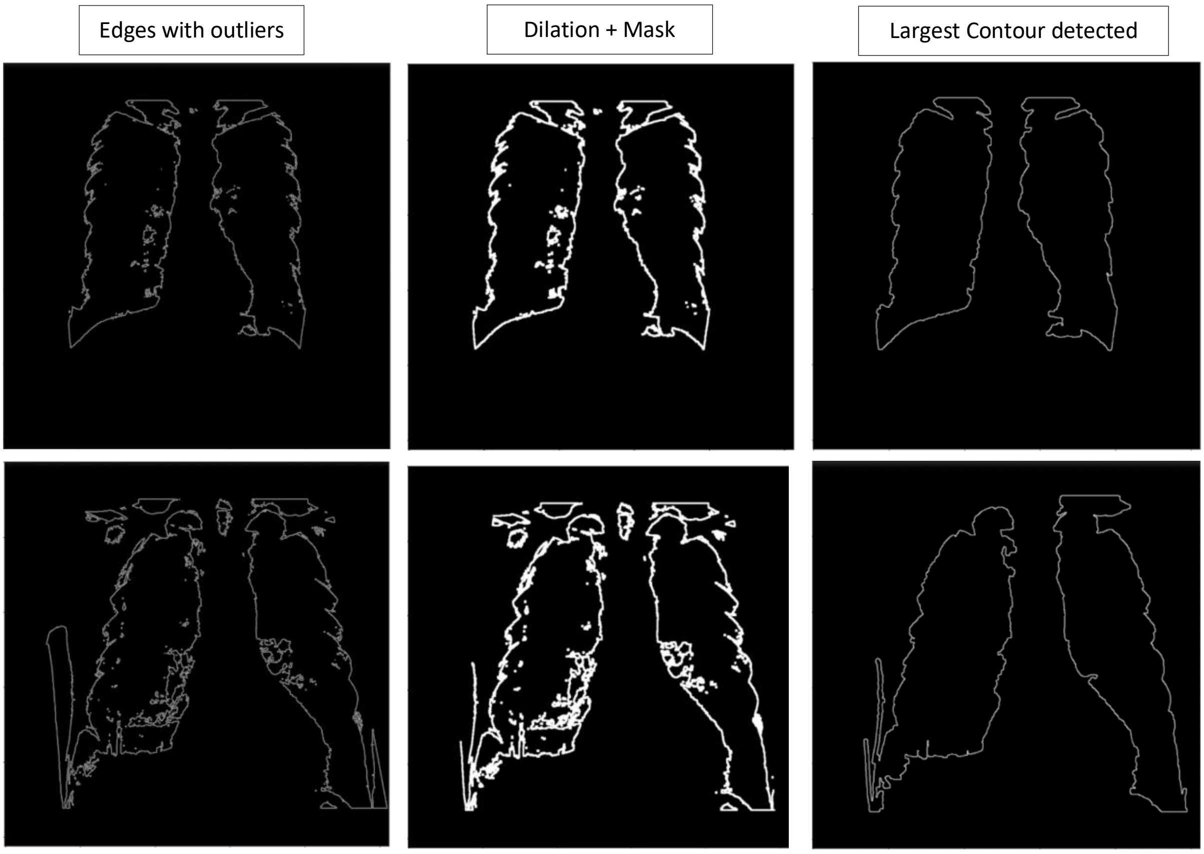
Algorithm:

- To retrieve ROI, i.e. the lungs boundary, the histogram equalised images are first **inverted**. This highlights the lungs from the rest of the body because in X-ray, the **organs are darker than bones**.
- Then, **Otsu thresholding** along with a spatial mask to get a binary image whose largest area are the lungs.
- Then **Canny Edge Detection** algorithm is used to get the contours/edges of the image.





- **It can be observed that there are some outliers in the edge detected images.**
- **These are removed, using spatial masks and contour detection methods.**



Observations:

- Most outliers have been successfully removed and an almost clean boundary of the lungs was retrieved
- Nevertheless, **some issues still persisted**:
  - The **lung region present behind the heart** needed to be retrieved as well.
  - Many masks still had very minimal outliers which included a few **skeletons of the ribcage**, while others had a bit of the **boundaries of the body** incorporated as well.

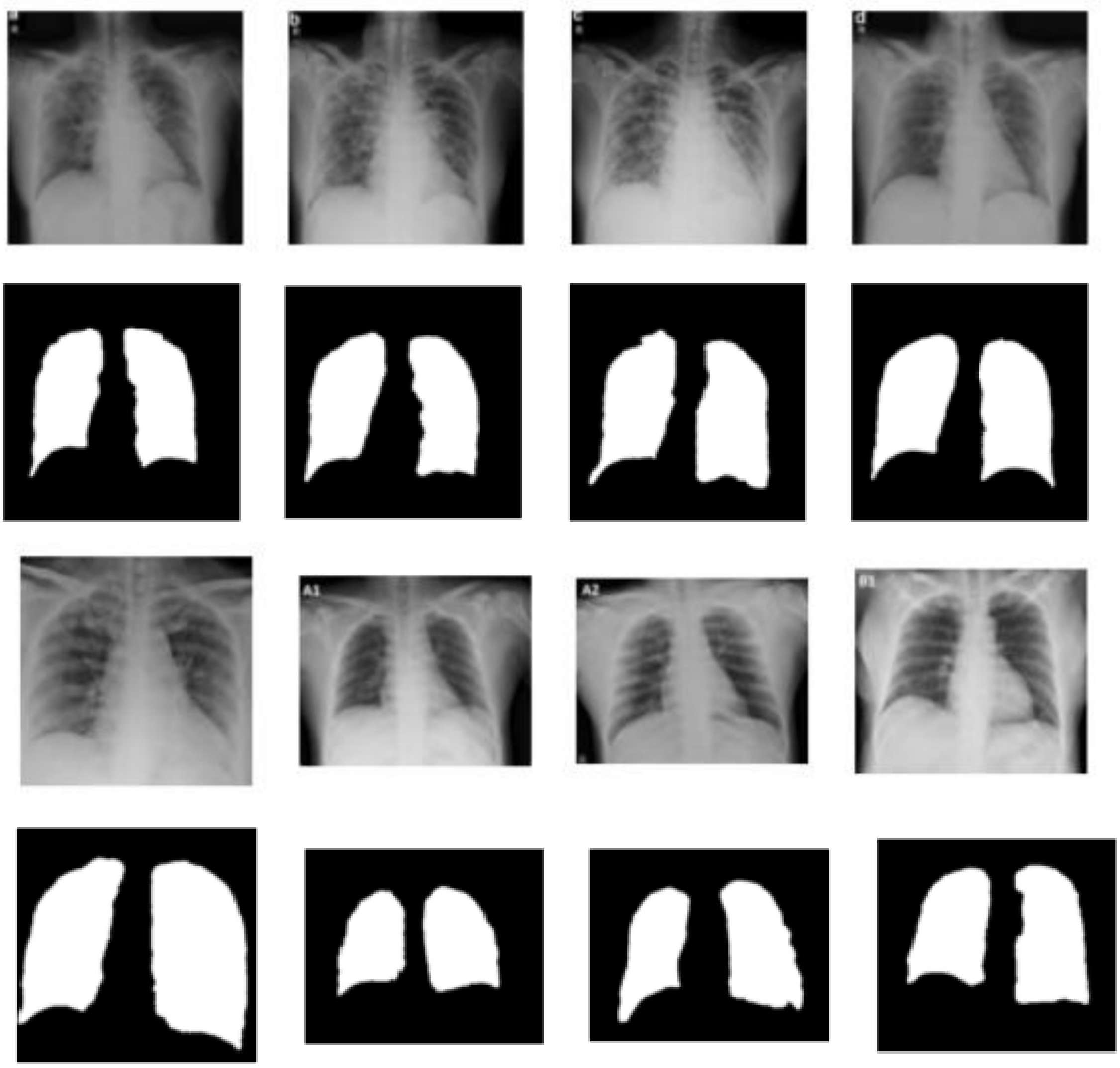
To take care of the above two issues the following steps were taken:

- To incorporate the lung region behind the mask, a shape prior was used (based on assignment 4) and then active contours were used to expand just the ROI which had to be grown.
- Further selective erosion, **manually,** on each image was done to remove the outlying skeletons.

Note: This was a bit time consuming and was the active contour segmentation was done on MATLAB Image Segmenter.

Manual erosion was done using MATLAB Image Processing Toolbox on all 762 images.

FINAL LABELLED GROUND TRUTH MASKS

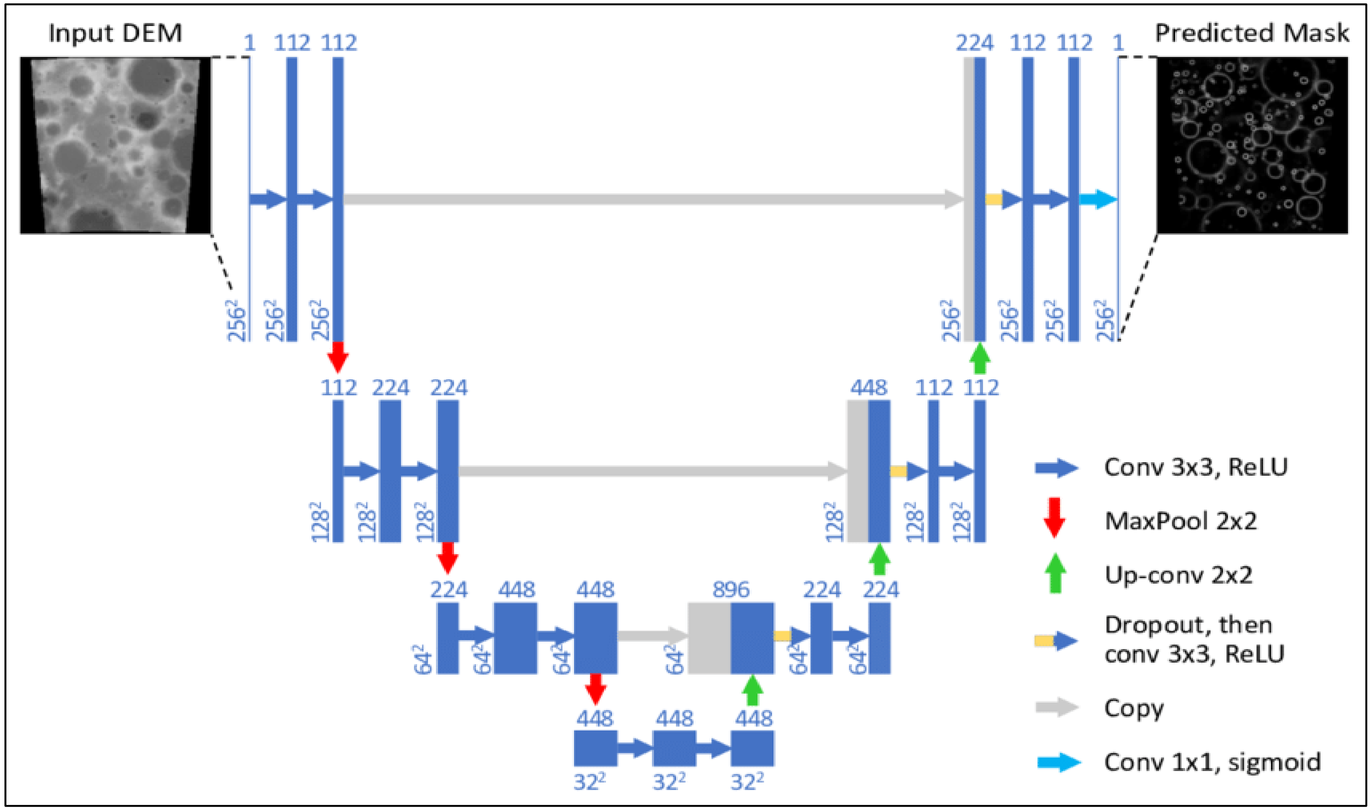


Part 2: Segmentation

Data Preparation for Training

- The data now is made up of 762 Chest X-rays with corresponding ground truth masks for each.
- As can be observed above, not all X-rays have the same dimensions, hence we resize all images to [256 x 256] pixels so as to make it suitable for training.
- Also, both ground truth and the image are converted to grayscale. This does not affect the predictions later as no loss of information due to this conversion.

U-Net Architecture



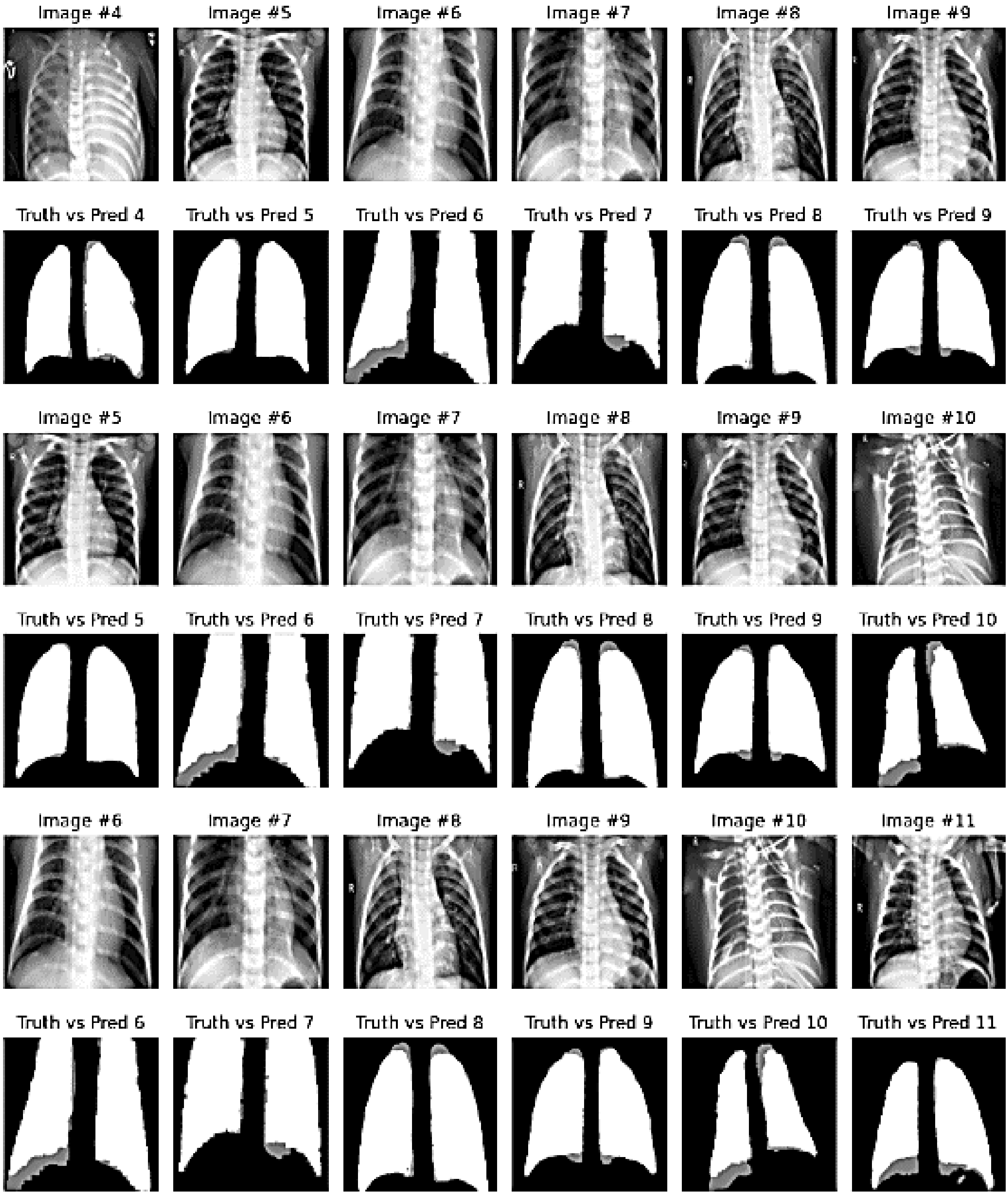
Source : U-Net: Convolutional Networks for Biomedical Image Segmentation (2015). Olaf Ronneberger, Philipp Fischer, Thomas Brox

- The U-net architecture is synonymous with an **encoder-decoder** architecture.
- Essentially, it is a deep-learning framework based on **Fully Convolutional Networks**; it has :
  - A **contracting path** similar to an encoder, to capture context via a compact feature map.
  - A symmetric **expanding path** similar to a decoder, which allows precise localisation. This step is done to retain boundary information (spatial information) despite down sampling and max-pooling performed in the encoder stage.
- Advantages of Using U-Net

Computationally efficient	Trainable with a small data-set	Trained end-to-end	Preferable for biomedical applications

Training and Hyper-parameter Tuning

- **Train-Validation-Test split – 0.75 , 0.15, 0.1**
- **Epochs – 100**
- **Early stopping applied.**
- **No data augmentation.**
- Experimented with multiple **IoU metric algorithms** to penalize boundaries of segmentation appropriately**.**
- Implemented in **Keras built on top of Tensorflow.**



Part 3: Validation

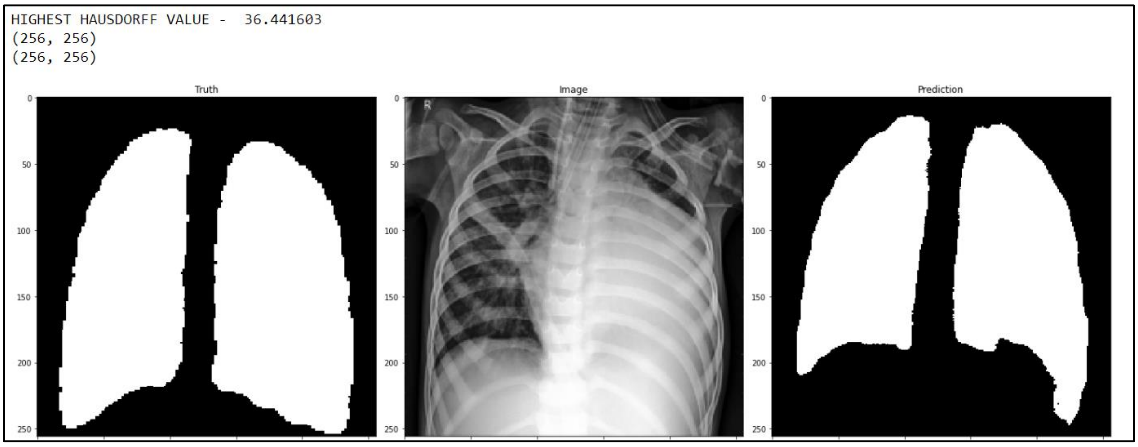
U-NET MODEL

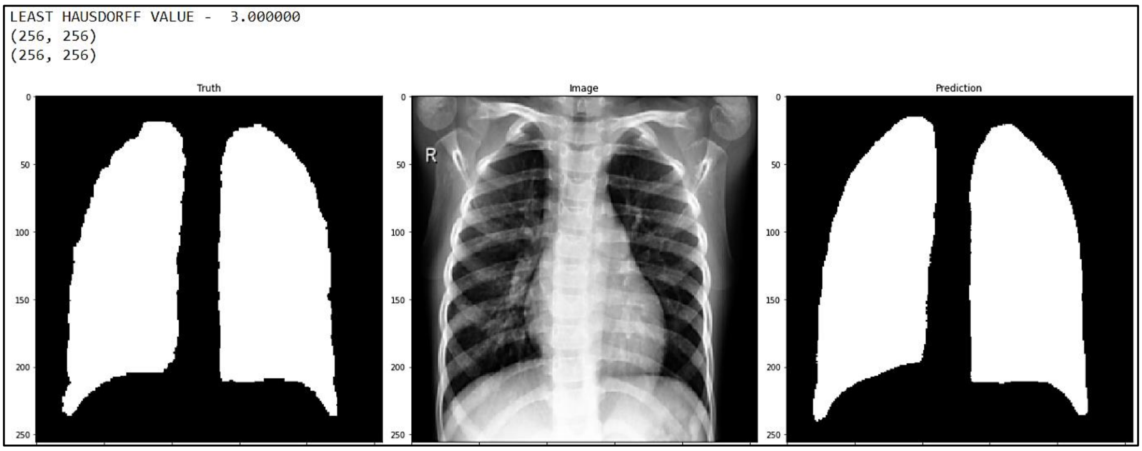
ESTIMATES	STATISTICAL DATA			
	max	med	min	**Avg ± SD**
DICE COEFFICIENT	0.977634	0.932780	0.836834	**0.926389 ± 0.034103**
JACCARD SCORE	0.956247	0.874029	0.719445	**0.864704 ± 0.057677**
HAUSDORFF DISTANCE	36.441603	16.000000	3.000000	**15.679237 ± 8.013009**
SENSITIVITY	0.974680	0.811603	0.717192	**0.886047 ± 0.067014**
ACCURACY	0.890436	0.711603	0.564554	**0.710910 ± 0.064370**

Probabilistic NN (Research paper)

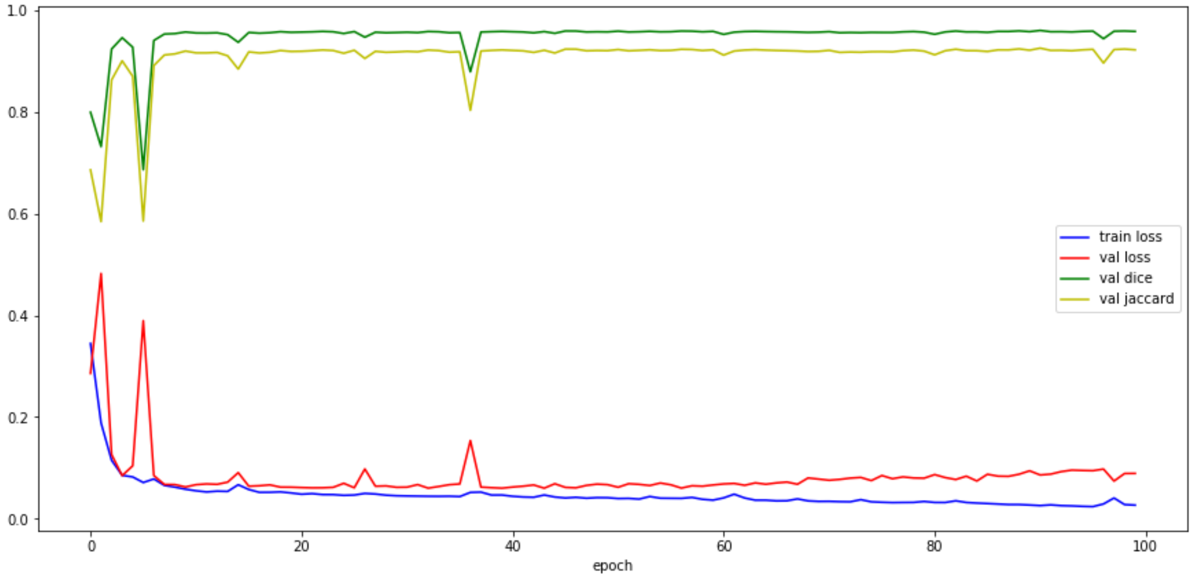
Estimates	Statistical data			
	min	avg ± SD	med	max
Jaccard similarity	0.882	0.915 ± 0.016	0.917	0.945
Sensitivity	0.940	0.964 ± 0.012	0.967	0.988
Accuracy	0.950	0.965 ± 0.007	0.964	0.977
Dice similarity coefficient	0.937	0.955 ± 0.015	0.957	0.971

Source : Research Paper





Observations & Conclusion:



- The metrics show the comparison between the ground truth and segmented lungs.
- The U-Net model although it was trained on minimal number of images, with no data augmentation still achieved acceptable results.
- Comparing with PNN used in the research paper, the statistical data clearly shows it was better. This is mainly because, it was trained with a larger dataset and better computational power (better GPU)
- Hausdorff Distance – Lower the Hausdorfff value, better the prediction. The highest and lowest predictions have been shown.
- The main reason for a higher HD seems to be because the model could not get the lower region features of the lungs, mainly as because the intensity level is similar to bones.
- Dice coefficient – As the values get closer to 1, the overlap gets better. Since our values are very much close to 1, our model has been able to detect the lungs well, but not segment it perfectly as shown by Hausdorff Distance.
- Our U-Net model can be made better with data augmentation enabled, training for longer epochs and better IoU metric approach, but this requires a stronger GPU, at least NVIDIA CUDA.

References:

- Aleksandr Zotina, Yousif Hamadb, Konstantin Simonovc, Mikhail Kurakob, " Lung boundary detection for chest X-ray images classification based on GLCM and probabilistic neural networks " 2019 23rd International Conference on Knowledge-Based and Intelligent Information & Engineering Systems.
- https://github.com/MEDAL-IITB/Lung-Segmentation/tree/master/VGG_UNet
- https://github.com/gaurav104/LungSegmentation/blob/master/build_model.py
- https://www.kaggle.com/keegil/keras-u-net-starter-lb-0-277

README

- All codes are written in Jupyter Notebook (.ipynb).
- https://drive.google.com/drive/folders/1G_Sfxxq5sdsr_woMI3v_pZthq5Ahp4K1?usp=sharing
- All data (scans + masks) can be found in above drive folder.
- The notebook files with output of each code block can also be found in the above drive folder.
- If any issues, I'm always ready to run the code in my laptop.