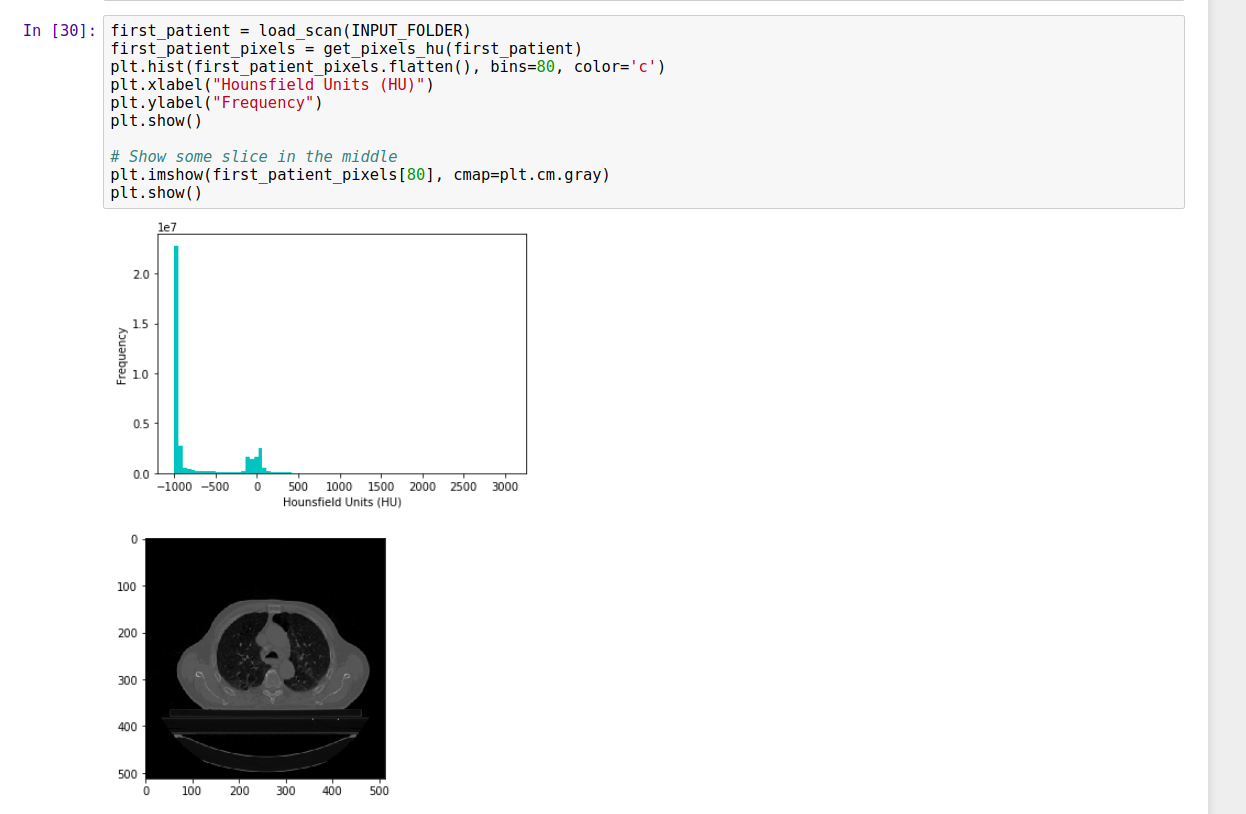
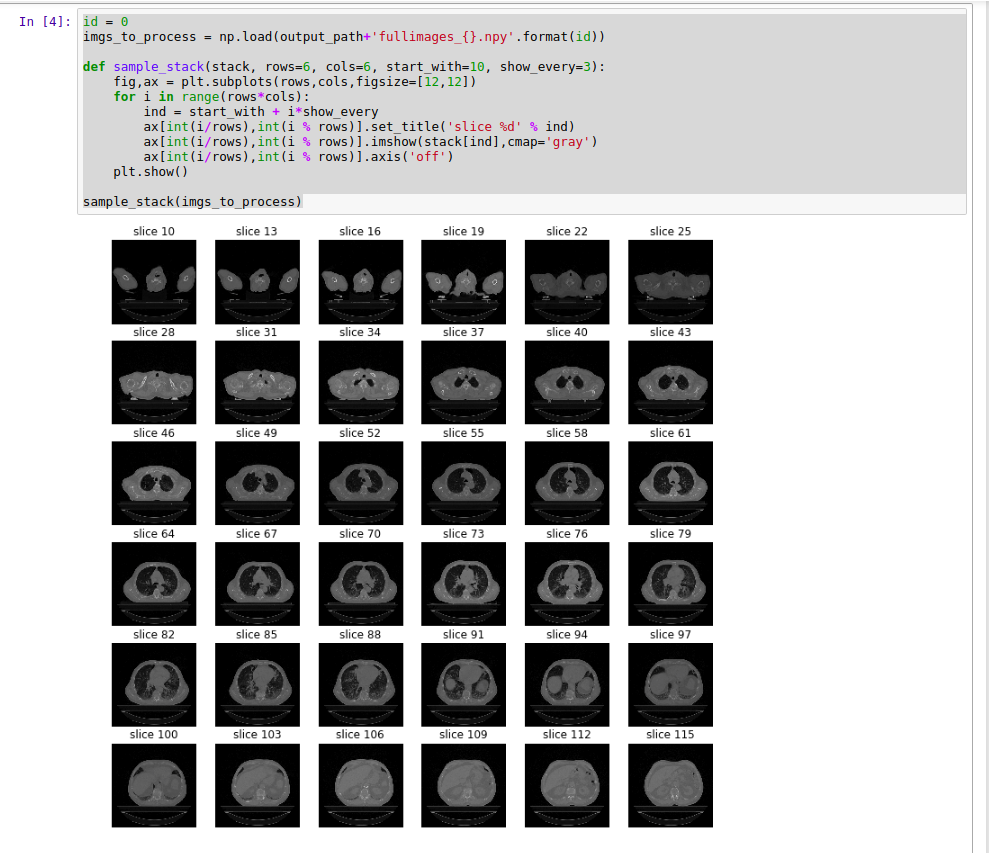
**1) Loading the DICOM files**, - Luckily the Data is in DCM format - which is a defacto standard for medical images. Hence I used some standard

**2)Converting the pixel values to *Hounsfield Units (HU)***, and what tissue these unit values correspond.

Values were around (-700 to -500 ) which corresponds to lungs. (correct me if I am wrong)



After this tried to slice the images an and display in a matrix of 12\*12 - I prefer to take one in every 5 or 4 images



what we were seeing as HU=-2000 are the voxels outside of the bore of the CT. "Air," in comparison, appears grey because it has a much higher value. As a result, the lungs and soft tissue have somewhat reduced contrast resolution as well.

This seems to be a issue, I am trying to address in the normalization part. ( **but have no idea how to do it**)

**3) Resampling** to an isomorphic resolution to remove variance in scanner resolution.

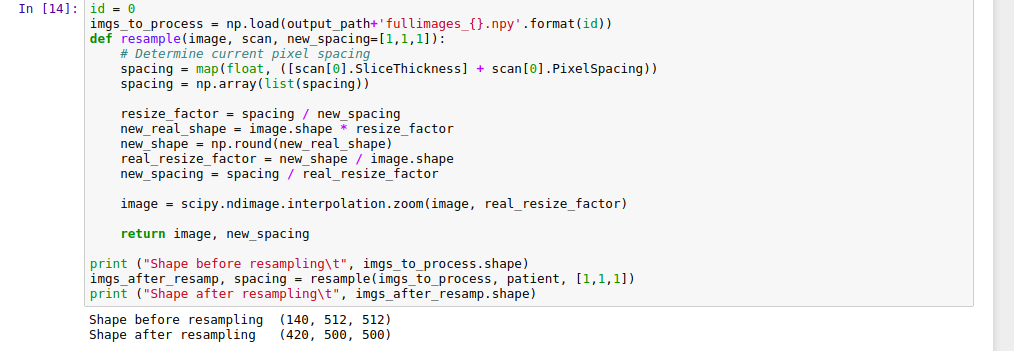
This step I need some clarifications.

I tried to reduce the image dimensions - initially, (Before Resampling it was 134,512,512)

Due to lack of powerful GPU -

I wanted the image size to be reduced to around (300,300,300) So I gave spacing to be (1,1,1) ( Iso-Morphic resolution).

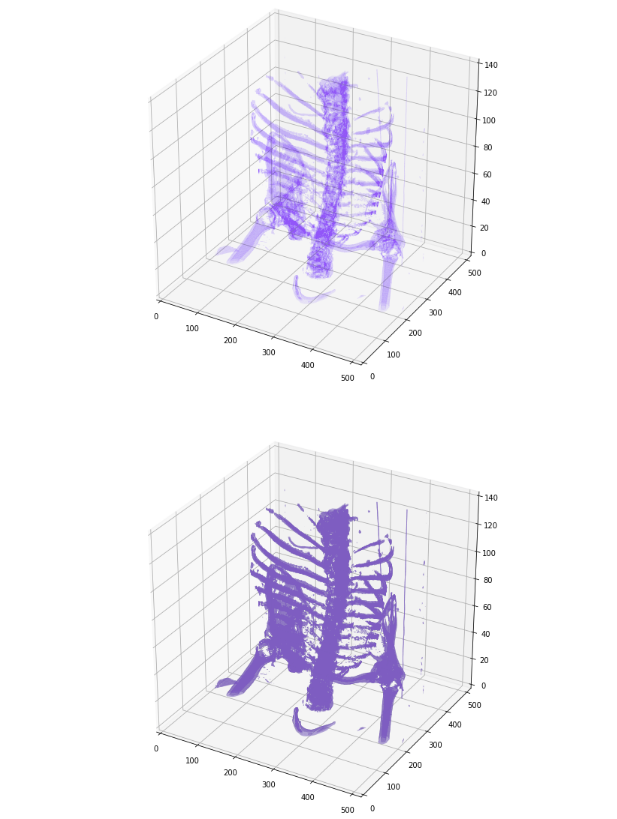
Also (1,1,1) will make all the images in the training dataset to be homogeneous so than in later stages to apply U-NET/conv net we need not worry about zoon/slice thickness.



I also tried some other values in hit and trail - your advice is needed to determine the final values for resolution.

**3D plotting**, I tried to do some 3D plotting - Using MarchingCubes API - I tried to plot the DCM images to 3d-Plot - It takes some time but able to plot it.

I was able to see the lung in the plots

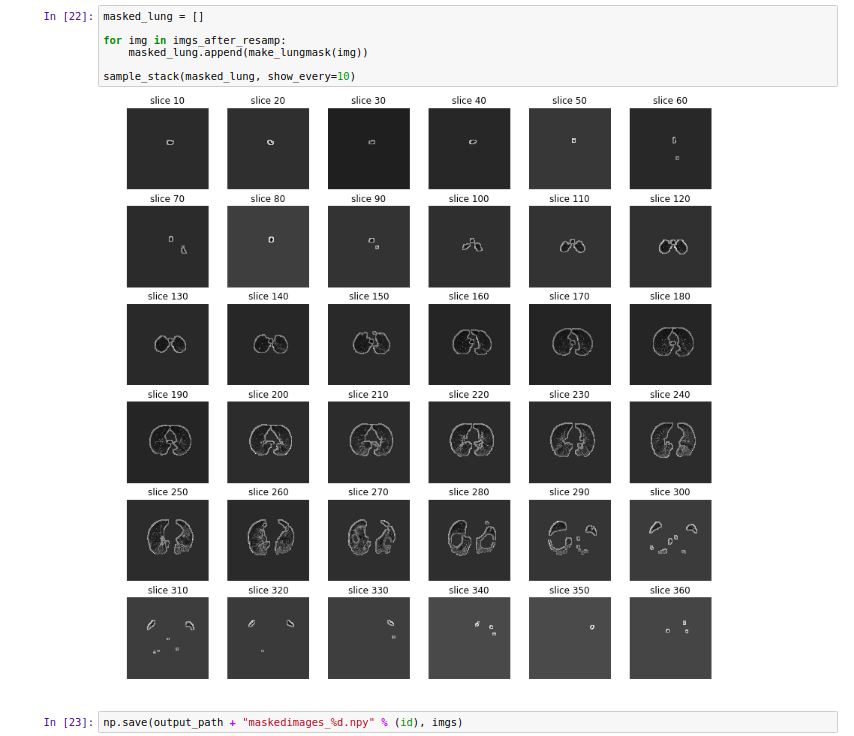


Standardize the pixel value and subtract the mean and dividing by the standard deviation. Used 2 Kmeans Clusters comparing centered on lung/air or some other tissue.

Using bounding boxes for each image label to identify lungs and which identify everything else.  
  
Finally, create masks for lung fields and apply mask onto the original image.



Then applied the same for all the images and saved the numpy ouput.



This is exactly what I have done till now, now I am trying to normalize the images and zero centres the scanner.  
  
I saw one 3d-uNET implementation which can be reused to our scenario.

My idea is to incorporate various loss functions into this model and see how the segmentation works ( uncommon loss functions like one used in NLP processing) - I am still learning this part. Now I have an almost clear understanding of the dataset and pre-processing is done

And trying to modify the architecture of the U-NET.