OAR Segmentation

# TO DO

* ~~Transforms not added~~
  + ~~I didn’t add data augmentation to the dataset yet. I added some code, but didn’t use it yet, because we need to add something for fixing the labels after transforming the volume. In the single class version they were simply thresholded, but here the values are class indici, so I guess we need to round them to their nearest existing class index.~~  **DONE**
* ~~I didn’t explicitly deal with the problem of overlapping organs (multiple organs being annotated for the same pixel). When making the label mask volume, it now just goes through the annotations for that volume, and overwrites the class index for pixels that occur multiple times. So the last annotation wins. It would ofcourse be better to have a sensible ordering of which classes should take precedence. But didn’t have time to finish that. Possible solution: save annotations as list or ordereddict in json file. While generating annotation list, append bowel\_bag first. DONE~~
* ~~Evaluation~~
  + ~~I removed the Dice calculation. Not sure what the standard is for evaluating on multiclass segmentation, but maybe implement a multiclass dice score?~~ **DONE**
* Center crop in\_plane
  + ~~Remove unneccessary black pixels on border~~ **~~DONE~~**
  + ~~Possible upsample crop to original size (not sure?)~~
  + generalize CropDepthwise to all dimension or implement separate class??
* Class weights
* ~~For 16bit training, use memory by increasing depthwise crop size (e.g. 16->32)~~
  + ~~Or batchsize DONE~~
* ~~Do experiment with with subset of classes and non-missing annotations~~
* ~~Drop background dimension in soft dice loss DONE~~
* ~~Do onehot encoding in soft dice loss DONE~~
* ~~Run dice loss experiment for binary and subset without class weights DONE~~
* Run focal loss experiments with different gamma’s
* Implement basic missing annotations solution (spatial masking)
* 2D UNET
  + Possibly with some depth as channels
* ~~For subjects like below: "/export/scratch3/bvdp/segmentation/data/AMC\_dataset\_clean\_train/3955374440\_1949203334/20140630"~~

~~probably in prone position, annotations are in negative pixel. Some correction related to axis reversal needs to be done. DONE~~

* ~~interpolation of annotation along depth dimension or maybe bug fixing for merged series DONE~~
* ~~iterate over all depths of the path to look for annotations.json DONE~~
* save annotations and matching dicom meta data in one file
* Check Dice loss training
* Possible experiments:- downsampled images (128\*128) so that entire CT can be passed

# Ideas

1. One has to start with Unet (obviously 3D). Let’s see what it does and confirm the effect of the underlying problem.

2. Set up “Backprogate gradients for only confident annotations” along with “curriculum learning” and “self-supervised learning”.

3. Unet with auxilliary losses in the upsampling path

4. Unet with auxilliary predictions fed back to the upsampling path

5. Think of something for active contours in neural networks, conditional random fields, refinement network, boundary loss term

# Dataset size for various combinations of classes

# 

# Data Preparation Pipeline

## For each date folder:

1. Read RTSTRUCT file from folder “1”

uid = rtstruct.ROIContourSequence[label\_idx].ContourSequence[contour\_idx].ContourImageSequence[0].ReferencedSOPInstanceUID

label name = rtstruct.StructureSetROISequence[label\_idx].ROIName

contour = rtstruct.ROIContourSequence[label\_idx].ContourSequence[contour\_idx].ContourData

Note: first two axes are in-plane (rl, ap), third is along slices (cc)

save list of dicts {uid, label\_name, coords}

2. Read dicoms from all studies and save list of metadata dicts. Uid = dicom names

3. Match uids in rtstruct and dicom metadata, wherever matching uid list > 1 consider it matching.

4. sort dicoms and dicom uids according to SliceLocation. Wherever dicom uid had annotation in rtstruct assign coords

5. If “PatientPosition” = “HFP”: in-plane represents (lr, pa). So do annotations. So invert all?

6. Preprocessing on dicoms: rescale intensity using RescaleIntercept and RescaleSlope. Apply WindowWidth and WindowCenter.

## Skip dataset in cases

* no pixel\_array can be extracted
* datatype not uint16
* max(array) == min(array)
* if any of the field from ["orientation", "origin", "SliceLocation", "PixelSpacing",

"SliceThickness", "Modality", "RescaleIntercept", "RescaleSlope", "PatientPosition",

"WindowWidth", "WindowCenter"] cannot be extracted

## RTSTRUCT criterion

* include = ['rectum', 'hip', 'bowel', 'bladder', 'sigmoid', 'spinal', 'anal\_canal', 'anal canal', 'blaas']
* exclude = ['ctv','ptv','gtv', 'hippo']
* exclude if geometry a point
* exclude if number in label\_{number} not equal to 100