Dataset

The Dataset can be loaded from:

https://drive.google.com/drive/folders/0B1fxAwWnf4CORUdoQnliN1dtSHM

it was released for the MICCAI 2015 segmenation challenge:

Raudaschl, P. F. et. al. Evaluation of segmentationmethods on head and neck CT: Auto-segmentation challenge 2015.

- 49 CTs from the head and neck area
- 9 segmented structures (in individual files): BrainStem, Chiasm, OpticNerve_L, OpticNerve_R, Parotid_L, Parotid_R, Mandible, Submandibular_L, Submandibular_R
- all in nrrd format
- Some structures are missing (generate statistics!!) -> to get as many combined structures as possible the Submandibular glands are removed

Preprocessing:

- collectStatistics.py: get dimension and spacing statistcs and bring all images to the same spacing (mean spacing is used (1.1, 1.1, 3)) -> there is one outlier with a really bad resolution in z-direction
- combineDatasetLabelFiles.py: combine all structures into one label file
- collectChallengeData.py: separate the datasets into Train, Test and Onsite data as it is done for the challenge in order to get comparable results ->
  - Train: 25 datasets
  - Test: 9 datasets (one is removed as the Chiasm is lost after the rescaling 0522c0576)
  - Onsite: 5 datasets
  - Other 9 datasets are ignored as some organs are missing (mainly the Mandible)

Other used scripts:

- rescale.py: upscales all datasets to the original size and spacing
- calc_foreground_label_otsu.py: used to get the foreground mask for the first stage of the training
- create_maskfrom_labels.py: creates a uniform frequency map from a label file (used for the weighted sampling process while training, and the foreground_selective sampling while infering)

Bug!!!! SCHEISE Foreground masks sind alle falsch!!!!!

Ganze eval neu!!!

calc_foreground_label_otsu.py did not work for all volumes!!!!

#this line caused problems if the order was the other way around!! why??

```
connected_np[connected_np != largest_component] = 0

connected_np[connected_np == largest_component] = 1


uniq, count = np.unique(connected_np.flatten(), return_counts=True)

print "uniq count after thresholding: ", uniq[:4], count[:4]


connected_itk = sitk.GetImageFromArray(connected_np)

connected_itk.SetSpacing(image_itk.GetSpacing())

connected_itk.SetOrigin(image_itk.GetOrigin())

connected_itk.SetDirection(image_itk.GetDirection())


#connected_itk = sitk.BinaryThreshold(connected_itk, largest_component - 0.1, largest_component + 0.1)
<- this line!!!!
```