**Approach**

**2d unet:**

**nah**

**3d unet:**

**Dataset**

Extracted the ground truth(GT) from the xml file. Filled the areas from the GT-surface, but selected only the pixels which 2 or more doctors agreed on. Saved the image-chunks of 64 images and rescaled the in XY dir to half, because of memory constraints. Saved only the chunks with GT in them. Scaled the Hounsfield units(pixel values) lower than -1024 to -1024, and higher than 400 to 400. don’t scale to 0-255 slice-wise.

**Architecture**

Unet with some modifications to work in 3d, but not like Vnet. Used 3d convolutions of (3x3x3) per layer, and a max pooling(2x2x2) between each 3dConv. Also included BatchNormalization for each convolution block

**Dataset split**

Split the data into ~80% training set, 10% validation and 10% test. Choose the 80 first patient for test, 80-160 validation to keep track of what the algorithm has trained on.

**Augmentation**

Flip: flip about a random axis or none

Rotate: rotate a random angle from -20 to 20. About xy and xz

implemented others, like elastic deformation, but this was slow to compute for ndarrays

**Hyper parameters**

For the 3d network the optimizer that worked best was ‘adadelta’. ‘adam’ and some others were tested but failed to converge to a suitable optimum.

**Training**

The best network were trained for 54 epochs, with predefined settings for the optimizer. Due to the memory constraint, the batch size was set to 1. Spatial dropout was set to 0.1 for the decoder and the encoder.

No post processing were implemented, but a change of threshold for the prediction changes the scores for specificity and sensitivity.

Implemented a predictor with overlap, but scored lower than the one with no overlap. But intuition tells us that overlap should perform better.

The algorithm may converge faster with [0,1] scaling. But it needs to be scaled from [~ -3000,~ 3000] to [-1024,~300] first to adjust for noise.

**Future work:**

* Applying a transform to make the voxels equal in [mm] in all directions, and train with those values may result in a better network.
* Make the network predict faster
* Try a deeper network with fewer convolutions per layer
* include non-nodules and small nodules.
  + Could be done with level set with the annotated voxel as the seed point.
  + Could implemented as different classes: (nodule, non-nodule, background) or (small nodule, nodule, non-nodule, background).
* Use the fact that 4 radiologists have annotated tha dataset:
  + Instead of binary GT, do something smarter
* Test the approach above on liver nodule segmentation.

**Classification with VGG**

**dataset:**

The dataset is made of of tumorwise 5d-arrays with 2 channels: [batch, z, x, y, channel], where the first channel is a chunk of the ct slice, but stretched/reduced in z direction to make the voxels equal in all dimensions, and the second is the binary segmented tumor.

**Architecture:**

The architecture is equal to the first half of the Unet (the encoder half), and with 3 dense layers at the top with 100 neurons.

*used the same dataset split and augmentation as the unet.*

used the optimizer ‘adam’ and the ‘categorical\_crossentropy’ loss function.

Future work:

* use a dataset with true ground truth(biopsy results), not just what radiologists have guessed from a ct scan.

Some articles on the subject:

Lung tumor segmentation algorithm:

<https://reader.elsevier.com/reader/sd/29784244B14FD1B93141E54C169EE59C2170A46E4AC982C744CF07453FB07418258B928E77A6A1C0EF33A8004E2A2353>

1. remove noise from the image
2. thresholding
3. Binary image with tumor and non-tumor
4. remove non-tumor from image

u-net:

<https://arxiv.org/pdf/1505.04597.pdf>

Deep learning algorithm with only CNN’s. 28 layers

Liver tumor segmentation:

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4983804/>

integrating cuckoo, fuzzy c-means and random walkers.

state of the art lung tumor segmentation and classification techniques reviewed (2015):

<https://thesai.org/Downloads/IJARAI/Volume4No4/Paper_6-Lung_Cancer_Detection_on_CT_Scan_Images_A_Review_on_the_Analysis_Techniques.pdf>

three stages:

1. pre processing
2. segmentation
   1. 2d
   2. 3d
3. classification

Lung CT Image Segmentation Using Deep Neural Networks: (U-NET)

<https://ac.els-cdn.com/S1877050918301157/1-s2.0-S1877050918301157-main.pdf?_tid=e0abba84-4759-4715-9f60-b145b647aaf9&acdnat=1529417704_a36fe4625d96980573e59c37f903b368>

segmenting the lung from the rest of the image, using the u-net. In the concussion -> next we will use u-net to segment the nodule (tumor) from the ct images (march 2018).

Lung tumor segmentation with fully CNN’s:

<http://cs231n.stanford.edu/reports/2016/pdfs/302_Report.pdf>

focus on segmentation, not detection. all images have tumors in them. not the best paper. using bounding boxes, not full segmentation(perhaps).

Lung Nodule Segmentation Using 3D Convolutional Neural Networks:

<https://beta.vu.nl/nl/Images/werkstuk-bronmans_tcm235-875352.pdf>

low computational power and a small network with goood results. (feb 18)

“DATA: From the Lung Imaging Database Consortium (LIDC)”

**Generalised Dice overlap as a deep learning loss function for highly unbalanced segmentations:**

<https://arxiv.org/pdf/1707.03237.pdf>

test of different loss functions, on 2d and 3d cases

**Spartial dropout:**

<https://arxiv.org/pdf/1411.4280.pdf>

ch: 3.2

LIDC stuff:

<https://www.sciencedirect.com/science/article/pii/S1361841515000316>

3d Conv-nets

**3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation**

<https://lmb.informatik.uni-freiburg.de/Publications/2016/CABR16/cicek16miccai.pdf>

**3D Convolutional Neural Network for Brain Tumor Segmentation**

<http://cs231n.stanford.edu/reports/2017/pdfs/526.pdf>

**v-net**

https://arxiv.org/pdf/1606.04797.pdf

no max-pooling, instead they use 2x2x2 convolution with a stride of 2 -> faster convergence

**classification**

<https://arxiv.org/pdf/1605.08350.pdf>

<http://www.cancernetwork.com/lung-cancer/radiologic-appearance-lung-cancer>