Notes 13-03

# Registration

## Metrics (Raquel)

Two methods:

1. ITK-library: however, doesn’t contain the normalized MI / Ascorrelation

Shows negative values

1. Sklearn-library: normalized MI.

Very low similarity score after the affine transformation.

* Look at similarity of all images after affine
* Look at similarity of all images before affine, to save even more time
* Look at similarity of all images after B-spline
* Select randomly X images to register

Leave-one-out cross validation is only during training

## Cross-validation (Funmilayo)

Parameter file can be changed automatically while running

Combining the codes is more difficult

* Look at the code together in the break

## Report (Lotte)

Fixed the Overleaf template

Added all sources, Affine transformation has been rewritten.

B-spline part structure is improved, however there is still some information missing.

* Important formula’s will be included in the report.
* Look at the B-spline again to add information

## Next steps registration

Finish code of cross validation in the break

Funmilayo runs the code for all patients, save images to make sure fusion of the images can be done without running the complete cross validation.

# Machine learning part

## Options for the project

1. Look more at the conditional GAN
   1. Advantages: insert ground truth label and get an image based on that label
   2. Disadvantage: there is a lot of code to add, however there is not much information about it yet
2. Take the already implemented approach from the practical’s
   1. Change layers
   2. Change activations
   3. Change optimizers
   4. Insert another convolution layer
   5. Noise variation
   6. Advantage: base is already implemented
   7. Disadvantage: look into the variational auto encoder to get it to run

Add the GAN on top of the VAE

Generative model is more important

## Code of VAE

* Code combination

## Cian

1. Proposal

* Include segmentations in generator (see lecture slides) spade
* Use registration approach to segment the generated data. It is however not perfect 🡪 PSEUDO labels, not perfect but helpful for training
* Now generate reconstructions of that training data, how to generate more different images out of the generator (sample randomly from latent space and put that into the generator)

1. How do we get that code to work?
2. Are we allowed to use another structure for the project code?

Line 22: out\_ch or in\_ch

Line 38: which order? out=self.relu(self.bn1(self.conv1(x))) or first relu and then bn?

Line 72: how do we get the height and width?

Input size / number of max pooling layers

Line 97: how does torch.chunk work? How does it split into a mu and logvar?

Line 137 (generator) doe we have to specify self.proj\_o? It is not in the github code anymore

Line 157 (generator) how to call / use the self.upconvs?

Line 86 (train\_vae): scheduler, gamma values? 0.9

# Next meeting tasks

* Funmilayo & Lotte: look at the code of registration (transformix)
* Raquel: update part of B-spline in the report + U-Net (with Christos)
* Milan: Look into SPADE (lecture)
* Christos: look at different implementations of U-net
* Noortje: look at VAE again

Next meeting Thursday at 14:00

## Next next meeting

Look at how to generate different types of images with the VAE