**UDCT**

You can find here the Cycle-GAN network (Tensorflow, Python 2.7+) with our histogram loss. Additionally, we provide the scripts we used to generate the synthetic datasets.

Our results can be found at <https://www.biorxiv.org/content/biorxiv/early/2019/03/01/563734.full.pdf>

**How to use**

1. Clone or download the repository
2. Create a synthetic dataset similar to your real dataset or use example dataset in ./Data/Example
3. Execute:

python create\_h5\_dataset.py <directory\_of\_raw\_images> <directory\_of\_syn\_images> <filename\_of\_hdf5\_file>

Example:

python create\_h5\_dataset.py ./Data/Example/Genuine/ \  
./Data/Example/Synthetic/ ./Data/Example/example\_dataset.h5

1. Create the directory 'Models' in the root directory
2. Execute:

python main.py --dataset=./Data/..../dataset.h5 --name=name\_of\_model

Example:

python main.py --dataset=./Data/Example/example\_dataset.h5 --name=example\_model

1. This will create a network that is saved in ./Models/ along with a parameter textfile. Furthermore, the average loss terms for each epoch are saved in this directory.
2. To generate the results after training, use:

python main.py --dataset=./Data/Example/example\_dataset.h5 --name=example\_model --mode=gen\_B

The generated synthetic images can be found in ./Models/<name\_of\_model>\_gen\_B.h5

**Parameters**

All parameters are of the shape: --<parameter\_name>=<value>  
Below is the list of all possible parameters that can be set. The standard value used if the parameter is not defined is given in brackets

**name ('unnamed')**  
Name of the model. This value should be unique to not load/overwrite old models. Its value must be changed to ensure functionality!

**dataset ('pathtodata.h5')**  
Describes which h5 files is used. Its value must be changed to ensure functionality!

**architecture ('Res6')**  
The network architecture for the generators. Currently, you can choose between 'Res6' and 'Res9', which corresponds to 6 and 9 residual layers, respectively.

**deconv ('transpose')**  
Upsampling method used in the generators. You can either choose transpose CNNs ('transpose') or image resizing ('resize').

**PatchGAN ('Patch70')**  
Different PatchGAN Architectures: 'Patch34', 'Patch70', or 'Patch142'. A mixture of these is possible: 'MultiPatch' (experimental).

**mode ('training')**  
Decides what should be done with the network. You can either train it ('training'), or create generated images: 'gen\_A' for raw images from synthetic images and 'gen\_B' for synthetic images from raw images.

**dataset ('pathtodata.h5')**  
Describes which h5 files is used. Its value must be changed to ensure functionality!

**lambda\_c (10.)**  
The loss multiplier of the cycle consistency term used while training the generators.

**lambda\_h (1.)**  
The loss multiplier of the histogram discriminators. If the histogram should not be used, set this term to 0.

**dis\_noise (0.1)**  
To make the network more stable, we added noise to the input of the discriminators, which slowly decays over time. This value describes how high the std of the gaussian noise is, which is added to the inputs.

**syn\_noise (0.)**  
It is possible to add gaussian noise to the synthetic dataset. Default: not used.

**real\_noise (0.)**  
It is possible to add gaussian noise to the real dataset. Default: not used.

**epoch (200)**  
Number of training epochs.

**batch\_size (4)**  
Batch size during training.

**buffer\_size (50)**  
Size of the buffer (history) saved to train the discriminators. This makes the network more stable.

**save (1)**  
If value is not 0, the network progress is saved at the end of each epoch.

**gpu (0)**  
If multiple GPUs exist, this parameter choses which GPU should be used. Only one GPU can currently be used.

**verbose (0)**  
If value is not 0, the network is more verbose.