In this laboratory, you are going to work with GANs, and in particular two types of architectures: pix2pix (image-to-image translation) and CycleGANs. For this final lab of the course, you are not requested to implement any architecture from scratch and you will provided with codes that are already ready to run. However, these architectures were originally implemented for other tasks so, as it usually happens in research, your main job consists of employing them for your specific field of application and fine-tuning the parameters to achieve a good performance. Moreover, we strongly recommend you to go through all the provided code, in order to understand how these more advanced architectures were implemented in Keras.

Task 1: pix2pix – from segmentation masks to X-ray images

For this task, you are going to work with a Keras implementation of pix2pix that was obtained from the following GitHub repository (and slightly modified in order to run it on our server): https://github.com/tdeboissiere/DeepLearningImplementations/tree/master/pix2pix

Please have a look at the repository and the explanatory figure which shows how, from simple sketches, realistic images of buildings were reproduced. You are going to work on a similar task, but this time you want to translate from chest segmentation masks to chest X-Ray images. In particular, the segmentation masks highlight five different body parts: left and right lung, left and right clavicle and heart.

Your training and validation data have already been pre-processed in order to be used as input to the network. You can find the .jpeg images in the folder */Lab1/Lab6/data_pix2pix/chest_xray/***.**

You can then run the training (*./pix2pix/main_pix2pix.py*). You can get an idea of the performance of your network by looking at the images that are being saved during training in the folder *./pix2pix/pix2pix/logging_dir_pix2pix/figures/*. You can try to improve the performance of the network by:

- Changing the number of epochs (line 23)
- Changing the weights of the L1 loss and binary cross-entropy loss (line 73)

Task 2: CycleGAN – from CT to MRI, and vice versa

You are now going to employ CycleGANs to translate from head CT images to MRI, and vice versa. The code was obtained (and slightly modified) from the following GitHub repository: https://github.com/simontomaskarlsson/CycleGAN-Keras

You training and validation data are saved in the folder */Lab1/Lab6/data_cyclegan/mrict/* (which is also already set as default *data_folder* in the function *load_data*). TrainA and TestA refer to MRI data, TrainB and TestB to CT data.

You can start training you CycleGAN by running *./cyclegans/main_cyclegans.py*. While training, images are saved after every epoch in the folder *./cyclegans/images/*. A new folder is created every time you run your code, and inside that folder different sub-folders are created to analyze separately the translations from A to B and from B to A. More information can be found in the GitHub folder linked above.

By looking at the first lines of the *main* code, you will notice that different parameters can be changed (e.g. loss weights and number of epochs, as in the task above). You can try to improve the performance of the network by changing these parameters.

Good luck!