A great and important challange: the synthesis of the medical images

Andrea Bellia 1586420
Alessia Carotenuto 1764400
Veronica Romano 1580844

15th September, 2020

Abstract

Our article focuses on the synthesis of medical images in order to avoid certain risks and limitations of medical imaging as it estimates an imaging mode without performing a real scan. Therefore, this article is based on a generative contradictory approach to address this problem and in particular, we reimplement the following: a Fully Convolutional Network (FCN) to generate a target image to which a source image is given; we use the antagonistic learning strategy to better model the FCN; the loss function is based on image-gradient-difference to avoid generating blurred target images; finally, we apply the Auto-Context Model (ACM) to implement a deep convolutional context-aware contradictory network. The results obtained show that the above method is accurate and robust to this task. In particular, we evaluate the method on "BraTS'20" dataset, to address the task of generating T2 from T1CE (T1 Contrast Enhanced) [4] [5] [6].

1 Introduction to medical imaging and work purposes

Medical imaging today is very important in the medical field for the diagnosis and treatment of various diseases. Normally not only one imaging mode is used but decisions are made on the basis of different modalities in order to have a complete view of the problem. In our specific case we have focused on images from Magnetic Resonance Imaging (MRI). MRI or simply MR is an imaging technique mainly used for medical diagnostic purposes. The main components of this procedure are a magnetic field, radio frequency pulses and means from which detailed images can be studied. MR is based on the physical principle of nuclear MRI, because the density signal in MR is given by the atomic nucleus of the examined element, while, in the most common radiological imaging techniques, the radiographic density is determined by the characteristics of the atoms affected by X-rays. MRI is often preferred to other techniques such as Computer Tomography (CT) because it is safer considering it does not include any radiation, but at the same time it is also more expensive. MRI images normally have dimensions from 256×256 pixels (cardiac images) to 1024×1024 pixels (high-resolution brain images) for a depth of 16 bits/pixel. This results in a rather low intrinsic spatial resolution (details of 1 mm are practically at the limit of visibility), but the importance of this examination lies in being able to discriminate, for example, between liver and spleen tissue (which have the same transparency as X-rays), or healthy tissue from lesions. The scan times are much longer than other

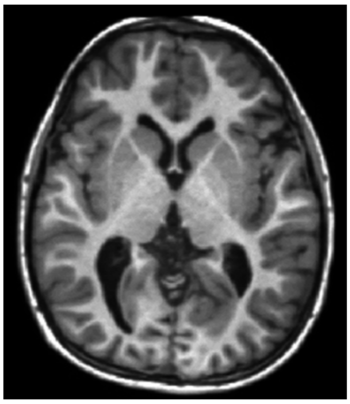

Figure 1: Example of T1 image of a specific brain section.

radiological techniques, and the temporal resolution is generally quite low (a few images per second for lower spatial resolutions). A fundamental feature of the resonance is the possibility to vary the type of contrast of the image simply by modifying the excitation sequence that the machine performs. For example, it is possible to highlight or suppress the signal due to blood, or obtain information of a functional nature instead of simply morphological. Moreover, MRI is a multiplane imaging technique, as it is possible to acquire images on axial, coronal or sagittal planes and multiparametric, as the reference parameters that can be used are both proton density and relaxation times T1 and T2. Unlike other imaging techniques, which allow the collection of information on a single physical quantity, MRI produces images that reflect different physical properties, depending on the type of sequence used. Images of different physical quantities are said to have different contrasts. At the end of the radiofrequency excitation, each tissue returns to its state of equilibrium thanks to its own processes of longitudinal relaxation (T1 or spin-spin, i.e. recovery of magnetization in the same direction as the static magnetic field) and transverse relaxation (T2 spin-spin, transverse to the static magnetic field). To create a T1-weighted image, the magnetization must recover before the signal is measured by changing the repetition time (TR). This image weighting is useful, for example, to evaluate the cerebral cortex, identify adipose tissue, characterize focal liver lesions and generally to obtain morphological information, as well as for post-contrast imaging. To create a T2-weighted image, magnetization can decay before measuring the MRI signal by changing the echo time (TE). This image weighting is useful for detecting oedema and inflammation, revealing white matter lesions and evaluating the anatomy of organs such as the prostate and uterus. In short, the weighing sequence in T1 or simply T1 is a measurement of longitudinal relaxation using short TR and TE, while weighting T2 or simply T2 is a measurement of transverse relaxation using long TR and TE times. In addition, T1 has a low signal from water, as in the case of edema, tumours, ischemia, inflammation, infection, chronic or acute bleeding, while T2 has a high signal from water-rich tissues. T1 has a high signal from fat and paramagnetic substances such as gadolinium (used as contrast medium in MRI) while T2 has a low signal. In Figure 1 and Figure 2 both of T1 and T2 images of the same brain section are shown. The evident difference can be noticed. Starting from the project illustrated by Dong Nie et al. "Medical Image Synthesis with Deep Convolutional Adversarial Networks" [7], which proposes a neural network for the transformation of MRI to CT images, we worked on the transformation of images from T1 to T2. Today deep learning has become fundamental and very popular in computer vision and medical image analysis. Various methods of analysis using Convolutional Neural Networks (CNN) have been developed, but it has been found that these tend to ne-

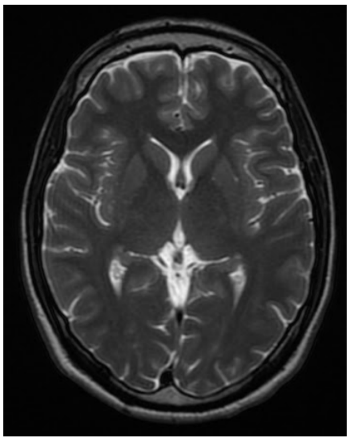

Figure 2: Example of T2 image of a specific brain section.

glect neighborhood information in the predicted target images, especially when the input size is small. To overcome this, FCNs, which can preserve structural information, have been used for image synthesis. Usually the distance L2 between the predicted target image and ground truth is used as a loss function to train CNN and FCNs. However, the resulting images are blurred. Minimizing L2 is equivalent to maximizing the PSNR (peak signal-to-noise rate) but it has been estimated (in 36->vediamo se vogliamo mettere il riferimento che avevano messo loro) that this does not improve the result. An alternative approach is proposed in [7]. The adversarial strategy is used to train at 3D FCN, which acts as a generator of realistic target images. In addition, simultaneously a discriminator network is also trained, which solicits the output of the generator to make it similar to the target image. To limit the image blur generated by FCN it has also been proposed to use an additional term in the loss function based on image gradient difference. A residual learning term has been adopted to make learning easier. Dong Nie et al. were inspired by Auto-Context-Model to concatenate several trained networks. In this way the whole framework includes long-range visibility to image information and become context-aware. They evaluated the method on three different real datasets and two tasks. Two datasets containing one brain image and the other pelvic image were used for the first task which consists of MRI-to-CT estimation. While the third dataset containing brain images was used for the second task which consists of 3T-to-7T image synthesis. Starting from their work, we reorganized the provided codes and adapted them to run another task that consists of T1-to-T2 image synthesis. The dataset we used is different from the original because of the difficulty in finding a dataset containing both MRI and CT images. However, the work of Dong nie et al. is not only adaptable to the MRI-to-CT or 3T-to-7T task, but also to many other possible transformations of images from MR. We were able to get the "BraTS'20" dataset, which contains both T1 weighted and T2 weighted brain images. In the following paragraphs we will explain in detail the dataset used.

2 Dataset choice and its principles

The first choice was the dataset and, in particular, the authors of our article proposed 5 datasets. Our choice was that of the "BraTS'20" dataset as it is easily accessible. In particular, the name of this dataset stands for "Multimodal Brain Tumor Segmentation Challenge 2020" and its purpose is the following: it focuses on advanced methods for the segmentation of brain tumors, particularly on a specific type of tumor - gliomas - in MRI scans. Another objective of "BraTS'20" is to predict the overall survival of the patient, through analysis of radiomic features and machine learning algorithms. In particular, our work focuses on the following task: predicting some MRI imaging sequences and specifically from T1CE to T2. In particular, the scans are available as NIfTI files (.nii.gz) and have been acquired with different clinical protocols and by various institutions. In addition, all images have been manually segmented, from one to 4 classifiers, following the same protocol annotation and later approved by qualified neuro-radiologists. Moreover, it is a important consideration that the program can easily be generalized for MR-to-CT prediction, 3T-to-7T prediction, etc... but we focused only on the above mentioned task because the datasets for the prediction of other tasks are not easily accessible and in particular this problem was common to other teams of researchers. As a result it was not possible to generate data for these two tasks and so the only goal we set ourselves was to compare our results with those proposed by [7]. We will show our results in the section "Experiments and results: our work VS the original work" and then we will compare our results with those of [7].

3 Network architecture and technical choices

For the elaboration of this work a deep convolutional adversarial network has been proposed to train a FCN as generator and a CNN as discriminator. Through a 3D FCN the target image is estimated from the target image. In particular, a 3D model has been adopted to solve the problem of discontinuity of the 2D model, which often happens with a two-dimensional CNN. The discriminator makes the generator output similar to the ground-truth target image. The generator incorporates the image gradient difference in the loss function in order to generate a sharp target image. In addition, the long-term residual unit has been explored to train the network and ACM has been used to refine the generator output iteratively. In the test phase, the input image is partitioned into patches and the target from the generator is estimated for each of them. Finally, all the target patches generated are merged into a single image. In the following sections we will describe in detail the Generative Adversarial Network (GAN) used for the purpose of the work.

3.1 Supervised deep convolutional adversarial network

As mentioned above, a supervised deep convolutional adversarial network inspired by a GAN was used, as illustrated in Figure 3.

3.1.1 Fully Convolutional Network for medical image synthesis

FCNs are widely used both for segmentation and for image reconstruction in computer vision and in the medical field, since the spatial information of the image is preserved and is faster than a CNN in the test phase. In [7] an FCN is adopted to implement the image generator. In particular, a 3D FCN, shown in image 3, has been proposed. Only convolution operations without pooling were used, which could lead to loss of resolution. An Euclidean loss was used to train the model:

$$L_G(X, Y) = ||Y - G(X)||_2^2 \tag{1}$$

where Y is the ground-truth target image and G(X) is the target image generated from the source image X by the G generator.

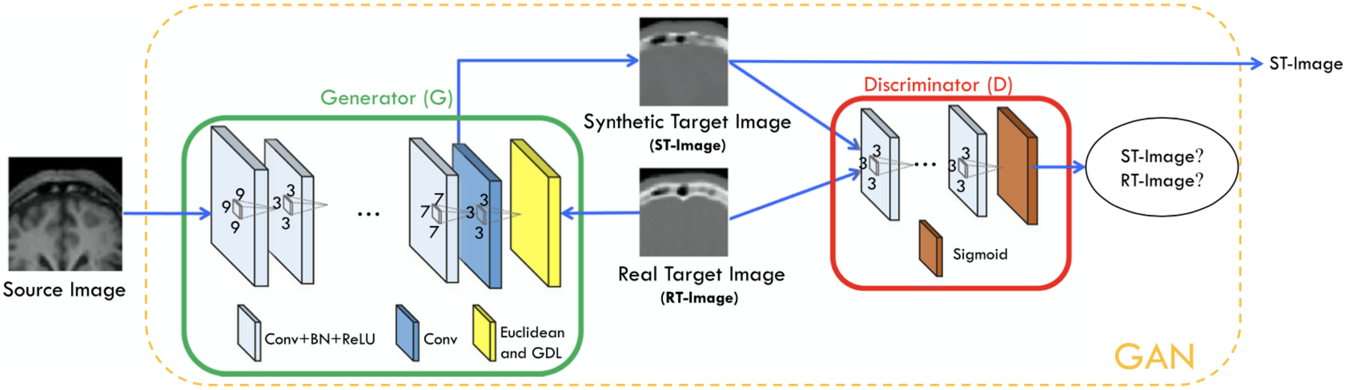

Figure 3: Architecture used in the deep convolutional adversarial setting for estimation of the synthetic target image.

3.1.2 Adversarial learning

To make the target image generated better, it was proposed to use adversarial learning to increase FCN performance. The network includes a generator to estimate the target image and a discriminator to distinguish the real target image from the generated one. The discriminator D is a CNN that estimates the probability of the input image will be extracted from a real image distribution. D can classify an input image as "real" or "synthetic". Both networks are trained simultaneously: D tries to correctly discriminate between synthetic and real data, while G tries to produce realistic images that confuse D. The loss function for G and D is:

$$L_D(X, Y) = L_{BCE}(D(Y), 1) + L_{BCE}(D(G(X)), 0) \tag{2}$$

where X is the source input image, Y the corresponding target image, G(X) is the image estimated by the generator, and $D(\cdot)$ calculates the probability that the input is real. Furthermore, the cross-entropy binary (L_{BCE}) is defined by:

$$L_{BCE}(\hat{Y}, Y) = -\sum_i Y_i log(\hat{Y}_i) + (1 - Y_i) log(1 - Y_i) \tag{3}$$

where Y which belongs to $0, 1$ represents the label of the given input (0 for the generated image and 1 for the real image), and $\hat{Y}$ is the probability predicted in $[0, 1]$ that the discriminator assigns to the input of being drawn from the real image distribution. While the loss term used to train G is defined by:

$$L_{G-ADV} = \lambda_1 L_{ADV}(X) + \lambda_2 L_G(X, Y) + \lambda_3 L_{GDL}(Y, G(X)) \tag{4}$$

which includes the loss of G. Specifically, the binary cross-entropy between D's decisions and the correct label is minimised, while G minimises the binary cross-entropy between D's decisions and the wrong label for the generated images. The loss function used for G includes an adversarial term with the purpose of deceiving D:

$$L_{ADV}(X) = L_{BCE}(D(G(X)), 1) \tag{5}$$

The training of the two networks takes place alternately: D is updated by taking a mini-batch of the real target data and a mini-batch of the generated one; then, G is updated using another mini-batch of samples that include the sources and their corresponding ground-truth target images.

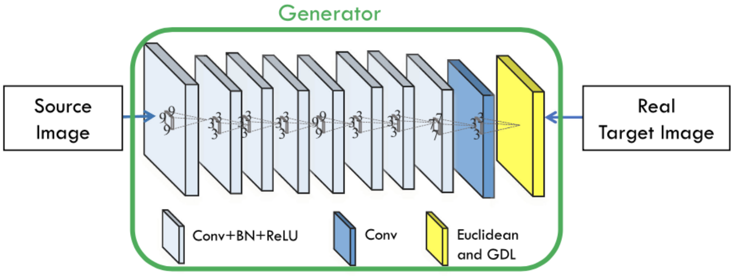

Figure 4: 3D FCN architecture for estimating a target image from a source image.

3.1.3 Image gradient difference loss

If we consider the formula (5) the system is already able to generate images extracted from the distribution of real images and using the L2 loss term the images produced are realistic. However, L2 loss can produce blurred images, so it has been proposed to add an image gradient difference loss ("GDL") term:

$$L_{GDL}(Y,\hat{Y}) = \left| |\nabla Y_x| - \left| \nabla \hat{Y}_x \right| \right|^2 + \left| |\nabla Y_y| - \left| \nabla \hat{Y}_y \right| \right|^2 + \left| |\nabla Y_z| - \left| \nabla \hat{Y}_z \right| \right|^2 \tag{6}$$

where Y is as in (1) and $\hat{Y}$ is the target estimated by the generator. This loss tries to minimize the difference of the magnitudes of the gradients between the ground-truth target image and the synthetic target image. In this way the synthetic target image keeps the regions with strong gradients, i.e. edges, to compensate for the L2 reconstruction term. By combining all the losses shown, the generator can minimize the loss function (4).

3.1.4 Architecture details

Figure 4 shows the architecture of generator G, built empirically, where the numbers indicate the size of the filters. The network input is the source image size $32 \times 32 \times 32$ and as output the network tries to return the corresponding target image size $16 \times 16 \times 16$. Using only small kernels $(3 \times 3 \times 3)$ would have too many layers and an overload of physical memory, so larger kernels have been used to decrease the depth of the network. The network has 9 layers that contain convolution, batch normalization (BN) and ReLU operations. The kernel sizes are shown in Figure 4, and the filter numbers are 32, 32, 32, 64, 64, 64, 32, 32, and 1 for the respective layers. Furthermore, in order to guarantee a sufficiently effective receptive field, dilated convolution has been adopted. For the first and last convolutional layer it is equal to one and two for all the other ones. The discriminator D is a typical CNN architecture includ- ing three stages of convolution, BN, ReLU and max pooling, followed by one convolutional layer and three fully connected layers where the first two use ReLU as activation functions and the last one uses sigmoid. The filter size is $3 \times 3 \times 3$, the numbers of the filters are 32, 64, 128 and 256 for the convolutional layers, and the numbers of the output nodes in the fully connected layers are 512, 128 and 1.

3.2 Auto-Context Model for refinement

Since our work is patches-based, the context information available is only that of the evaluated patch and this affects the modeling capacity of our network. To solve this problem, an ACM was used. Several classifiers are iteratively trained using the feature data of the original image

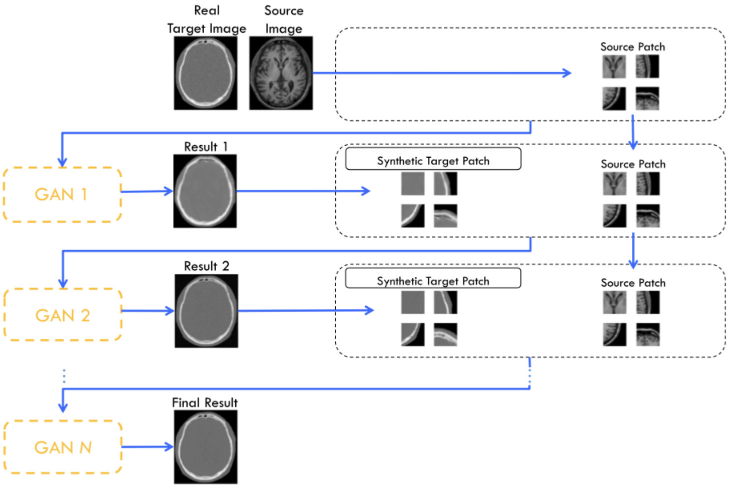

Figure 5: Proposed architecture for ACM with GAN.

and the probability map returned by the previous classifier; this provides additional context information. At testing time, the input will be processed for each classifier one after the other, concatenating the probability map to the initial input. The ACM was adopted to iteratively refine the generated results making the GAN context-aware. Several GANs are trained and take both the synthetic target patch and the source patch as input, as shown in Figure 5. These patches are then chained to become the input of the next GAN. The same size of the input patches was maintained during ACM-based refinement. Because context information is extracted from the entire previously estimated target image, it can encode information that is not available within the initial input image patch.

4 Experiments and results: our work VS the original work

As said before we use "BraTS'20" dataset to analyze the proposed method on the task of predicting MRI image sequences, from T1CE to T2. So, in this section we will provide the experiments and results of our task and then we will compare them to those of [7].

Before running the experiments, it is necessary to convert the dataset that you want to study, using one of the .py files present in the project, dedicated to the conversion of the files. These files allow to convert a multiple number of extensions ("nii", "hdr", "mha", etc.) of the datasets into files with "h5" extension that will be used later for the training phase. However, for the generation of these files it is necessary to obtain a training dataset containing both the input images of the program (which can be for example MRI or T1CE), and the output images (for example CT or T2) as both are used for the generation of h5. It should also be emphasized that for each pair of files, 2 “h5” files are generated, one traditional, while the other with a reverse along the 1st dimension. The network, with a batch size of 32, needs 381 iterations to process a single h5 file of our dataset. Not having excellent computers we decided to carry out 3810 iterations and therefore to use a limited number of h5 files for training (10 in total, 5 normal + 5 reversed) as each file required more than 20 minutes of processing. Once this phase is completed, the training h5s are used by the neural network to generate models with the .pt extension that can be used later for predictions. The training phase was possible thanks to the use of the PyTorch library (v. 1.6.0) [3] and thanks to the CUDA [1] development environment (v. 10.2), through which it was possible to perform parallel computing on the GPUs of our

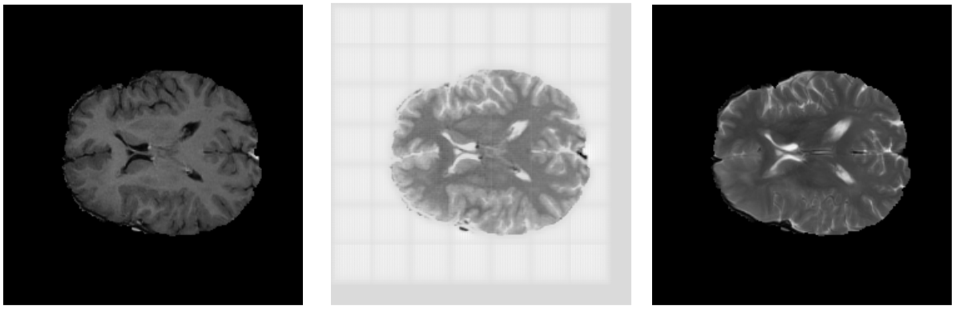

Figure 6: A single prediction that we have achieved. (left) T1CE, (center) T2 prediction and (right) T2 original.

NVIDIA video cards; the used version of Python is 3.7.8 [2]. As regards the training loss statistics we obtained the following results (showing the average value):

- Average running loss for generator: 0.0324;
- Loss term to train the generator network (G): 0.0304;
- Image gradient difference loss (GDL): 0.000075.

Once the training is finished, it is now possible to make predictions starting from new input files. We then selected 10 T1CE files and generated their predictions in T2. The results, even if generated with a model resulting from a limited number of iterations, were really good. In fact, comparing the prediction files with the original T2 files, the differences are really minimal. Some predictions of the model we have obtained are shown in Figure 6. The average value of the PSNR obtained is equal to 12.2663; this is obviously slightly lower than the values found by the authors of the paper precisely because of our limited training.

5 A final comparison between our work and the original one

At the end we can highlight some main differences between our work and that proposed by [7]. The first important difference is the dataset: in the original work three datasets were used and moreover our dataset is not included in those; in fact, thanks to the suggestion of one of the authors of the original article, we used the dataset "BraTS'20". Another important difference is the following: in the original article two tasks were performed and analysed, i.e. the estimation of CT images from its corresponding MRI data and the estimation of 7T MRI data from its corresponding 3T MRI data; while our task is also a medical image synthesis task, but is based on the estimation of T2 MRI data from its corresponding T1 MRI data. Actually, the network we set up could easily handle many other tasks for converting images, but we could not test them because of the difficulty in finding suitable datasets. Moreover, there is another difference from the point of view of network implementation: in the original work the residual learning is used for the generator only in the 3T-to-7T task while we do not use it. Let's see specifically what it is and why it could be important. CNNs with residual connections are useful for the image processing task because they help bypass non-linear transformations with identity mapping in the network. In particular, the residual connections can be defined as follows:

$$y = F(x, W_i) + x \tag{7}$$

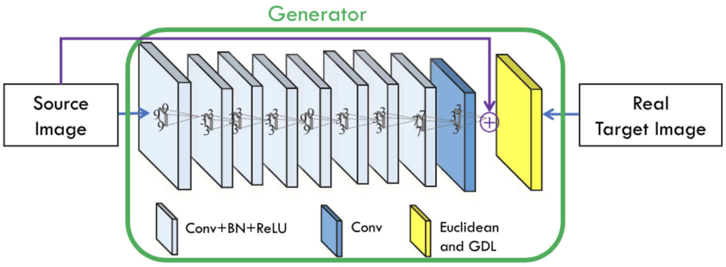

Figure 7: Generator architecture used in the GAN setting for estimation of synthetic target image. Note the solid-purple-line arrow from source image to the “plus” sign, which expresses the long-term residual connection.

where W_i are the convolutional filters in the residual bottleneck unit and x and y are the input and output feature maps respectively. The residual connections help the convergence of very deep CNNs, only in the case in which input and output are highly correlated. With the long-term residual unit (represented by the purple arrow in Figure 7). The residual image is close to being zero making the goal easier to train. For our T1-to-T2 task, where the input is similar to the output, it can be difficult for a deep network to achieve accurate results as the model requires very long-term memory. This required memory is difficult to model during training due to the vanishing gradient and the large number of layers; so, residual learning can help solve this problem by learning a residual map for the last layer. This is all achieved by adding a direct connection from the input to the final layer and then performing an element-wise addition. Finally there is one last difference: the number of iterations. In our case the number of iterations has been considerably reduced because we do not have powerful machines and therefore it should be underlined that with more iterations we would certainly have had results closer to those of the original work and more satisfactory. One last thing to note is that with the code we have re-implemented, and without any changes, you can make other types of modality conversions. Two examples are those made from the original work, i.e. MR-to-CT and 3T-to-7T; in particular, it was not possible to re-implement these tasks due to lack of datasets as already pointed out in the "Dataset choice and its principles" section. Therefore, we can conclude that the differences in the results we have obtained are certainly given by these different implementation choices, but despite this the results we have achieved can be considered satisfactory.

6 Conclusions and potential developments

Our work focuses on the method proposed by [7] and therefore is based on a deep convolutional supervised model for the estimation of a target image from a source image through adverse learning. Especially a particular loss is used to avoid generating blurred target images, i.e. image gradient difference loss. In addition, a long-term residual connection is used to pull the network more easily. Last but not least, ACM is used to improve network performance in order to have context awareness. We tested the proposed model on the task of predicting MRI image sequences and in particular generating T2 from T1CE. Finally, the results obtained show the following conclusion: the proposed model significantly outperforms the current state of the art and also shows that the model can be generalized to the synthesis of other medical imaging modalities with attractive results, as for example in [7] CTs are generated by MRI or even 7T MRIs are generated by 3T MRI.

References

[1] Cuda: Official website.

[2] Python: Official website.

[3] Pytorch: Official website.

[4] Spyridon Bakas, Hamed Akbari, Aristeidis Sotiras, Michel Bilello, Martin Rozycki, Justin S Kirby, John B Freymann, Keyvan Farahani, and Christos Davatzikos. Advancing the cancer genome atlas glioma mri collections with expert segmentation labels and radiomic features. *Scientific data*, 4:170117, 2017.

[5] Spyridon Bakas, Mauricio Reyes, Andras Jakab, Stefan Bauer, Markus Rempfler, Alessandro Crimi, Russell Takeshi Shinohara, Christoph Berger, Sung Min Ha, Martin Rozycki, et al. Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the brats challenge. *arXiv preprint arXiv:1811.02629*, 2018.

[6] Bjoern H Menze, Andras Jakab, Stefan Bauer, Jayashree Kalpathy-Cramer, Keyvan Farahani, Justin Kirby, Yuliya Burren, Nicole Porz, Johannes Slotboom, Roland Wiest, et al. The multimodal brain tumor image segmentation benchmark (brats). *IEEE transactions on medical imaging*, 34(10):1993–2024, 2014.

[7] Dong Nie, Roger Trullo, Jun Lian, Li Wang, Caroline Petitjean, Su Ruan, Qian Wang, and Dinggang Shen. Medical image synthesis with deep convolutional adversarial networks. *IEEE Transactions on Biomedical Engineering*, 65(12):2720–2730, 2018.