Redemptor Jr Laceda Taloma
1704899

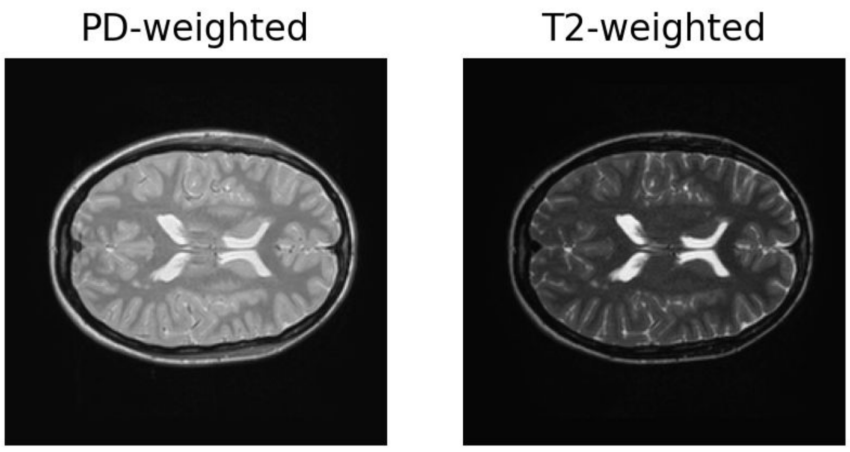
Medical Image Synthesis with DCGANs - Report

I. Abstract

Multiple considerations such as cost and radiation dose may limit the acquisition of certain medical image modalities, which play nowadays a critical role in various clinical applications. The work of D. Nie et al., "Medical Image Synthesis with Deep Convolutional Adversarial Networks" (2018) proposes a DCGAN to estimate a desired imaging modality of brain and pelvic images without incurring an actual scan.
More specifically, my own project implements the authors' DCGAN to synthesize a T2-weighted 3D brain image from the actual PD-weighted 3D brain image as it was taken from the subject. Actually the authors addressed the tasks of generating CT from MRI and generating 7T MRI from 3T MRI images; my different choice wrt to modalities is due only to the availability of datasets online.
The DCGAN is trained by means of a particular loss function whose main component - among the others - is the Image Gradient Difference Loss (GDL). The first experiment is to assess the contribution of GDL in the adversarial learning approach (1); the second experiment adds an input-output skip connection in the generator's architecture in order to leverage the high similarity between PD and T2 images, thus improving the synthesis quality (2); the third experiment applies Auto-Context Model to implement a context-aware deep convolutional adversarial network (3).
General metrics like MAE, MSE and PSNR are used for evaluation.

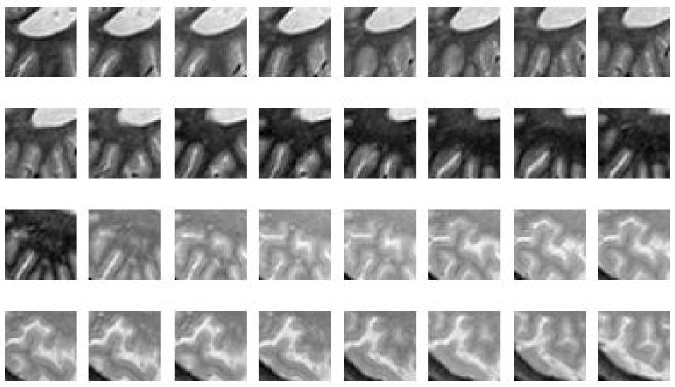
II. Dataset

The IXI brain dataset (https://brain-development.org/ixi-dataset/) provides T2 and PD-weighted images in .nii.gz format collected from nearly 600 healthy subjects. 60 samples are used for training, 3 to check the best model among all training epochs and 26 for final testing. Each 3D volume on which I worked on was cut down to a 128x224x224 matrix (128 grayscale horizontal slices of size 224x224), hence the dataset is finally an array of such matrices. This representation lets to get a horizontal rather than a sagittal or coronal slice from a subject. Final test will evaluate scans of all three plans.



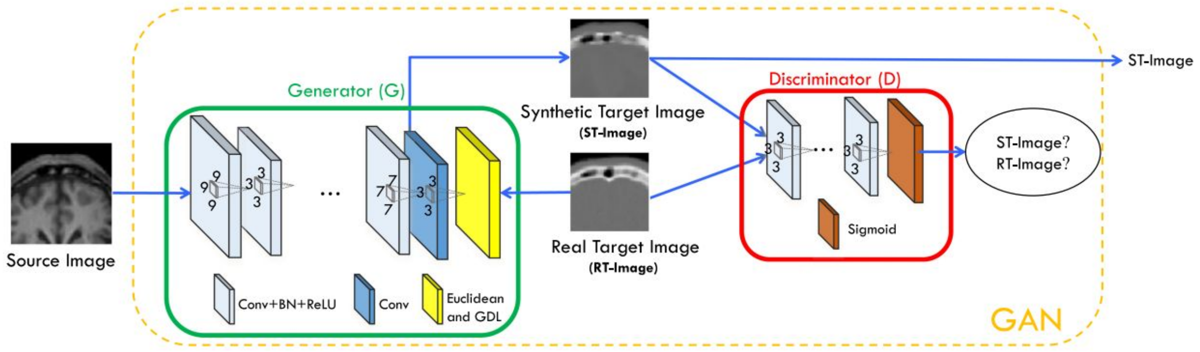
Horizontal scans

The generator and the discriminator the GAN is composed of cannot work well on a 128x224x224 matrix. Indeed, they actually take a 32x32x32 3D patch in input rather than the whole volume in order to focus on smaller details in the image; so the volume of each subject is partitioned into 196 cubes, which means 60x196 = 11.760 samples included in the training set. I was definitely working on a PatchGAN. Normalization and standardization are applied to both T2 and PD-weighted datasets in order let the many activation functions to work properly.



32 PD-weighted horizontal patches

III. Architecture



GAN architecture from the authors' paper

Generator

This model is defined as a FCN that takes as input the source image and tries to return a plausible synthetic target image. This network has 9 layers containing convolution, batch normalization, and LeakyRelu operations. The kernel sizes are 9^3, 3^3, 3^3, 3^3, 9^3, 3^3, 3^3, 7^3, 3^3. The numbers of filters are 32, 32, 32, 64, 64, 64, 32, 32, 1. The last layer only includes 1 convolutional filter, and its output is considered as the estimated target image. Regarding the architecture, Max-Pooling layers are avoided because they reduce the spatial resolution of the feature maps. The generator adopts also dilated convolution to expand the receptive field; in particular the dilation rate for first and last convolution layers is 1, and 2 for all the rest.

Discriminator
The discriminator is a typical CNN architecture including 3 stages of convolution, batch normalization, LeakyRelu and Max-Pooling, followed by one convolutional layer and 3 fully connected layers where the first two use LeakyRelu as activation function and the last one uses the Sigmoid. The filter size in each layer is set as 3x3x3, the numbers of the filters are 32, 64, 128 and 256 for the convolutional layers, and the number of outputs nodes in the fully connected layers are 512, 128 and 1. Two arrays of size 32x32x32 are given in input to the discriminator, which represent the source image and the target image.

GAN
It is precisely a conditional GAN because the T2-weighted output of the generator is conditioned by the PD-weighted input. The GAN is set to return the synthetic target images by the generator and the predictions by the discriminator, but at testing time only the generator is needed.

IV. Loss function

The loss function of the discriminator (*D*) is the binary cross entropy:
$L_D(X, Y) = L_{BCE}(D(Y), 1) + L_{BCE}(D(G(X)), 0)$

The loss function of the generator (*G*) is
$L_{G_{ADV}}(X, Y) = \lambda_1 L_{ADV}(X) + \lambda_2 L_1(X, Y) + \lambda_3 L_{GDL}(Y, G(X))$, where:

- *X* is the source image
- *Y* is the real target image
- *G*(*X*) is the synthetic target image produced by the generator
- *D*(·) is the probability that the input is real according to the discriminator
- *L₁* loss: $L_1(X, Y) = ||Y - G(X)||_1$
- Adversarial loss: $L_{ADV} = L_{BCE}(D(G(X)), 1)$
- 3D Image Gradient Difference loss:
  $L_{GDL}(Y, G(X)) = || \nabla Y_x| - | \nabla G(X)_x||^2 + || \nabla Y_y| - | \nabla G(X)_y||^2 + || \nabla Y_z| - | \nabla G(X)_z||^2$
- Weights: $\lambda_1 = 0.5,\ \lambda_2 = 1,\ \lambda_3 = 1$

GDL minimizes the sum of the squared differences between the magnitudes of the gradients of *Y* and *G*(*X*) along the axises *x*, *y*, *z*. It's designed to keep the regions with high gradients, i.e. the edges of the brain in front of the background and the borders along inner areas of different intensity, possibly generating sharper images.

V. Reconstruction from patches

At testing time an input source volume is divided into 32x32x32 overlapping patches and, for each patch, the corresponding target is estimated by the generator. Then, all generated target patches are merged into a single image to complete the source-target synthesis by averaging the intensities of overlapping regions.
I followed the authors which set the stride to be 8 along each direction of the image for the overlapping target image regions. All this means 8.125 overlapping patches per subject at testing time.

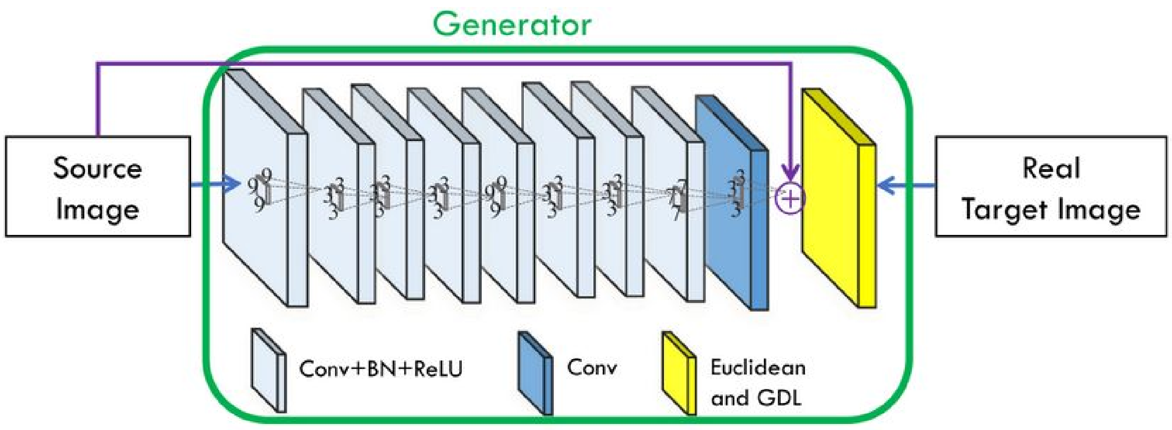
VI. Experiment 1: Image Gradient Difference Loss

Model	Best epoch on validation set	MAE on test set	PSNR on test set
Adv.	40/42	0.109 (0.024)	30.861 (2.044)
Adv. + GDL	78/80	0.105 (0.022)	31.074 (2.267)

Adversarial loss combined with GDL (Adv. + GDL) required double the training epochs before overcoming the performance of the baseline model (Adv.) on validation set. This is due to GDL which needs to be minimized along three directions. Eventually the Adv. + GDL model hits better MAE and PSNR scores on test set.

VII. Experiment 2: speed up training with Residual Learning (ResGAN)

PD-weighted and T2-weighted images in the IXI brain dataset can be aligned perfectly one on top of the other; moreover they are also highly correlated (i.e. very similar), so a skip connection performing a pixel-wise addition between the source image and the output of the last convolutional layer of the generator could in principle speed up the training phase.



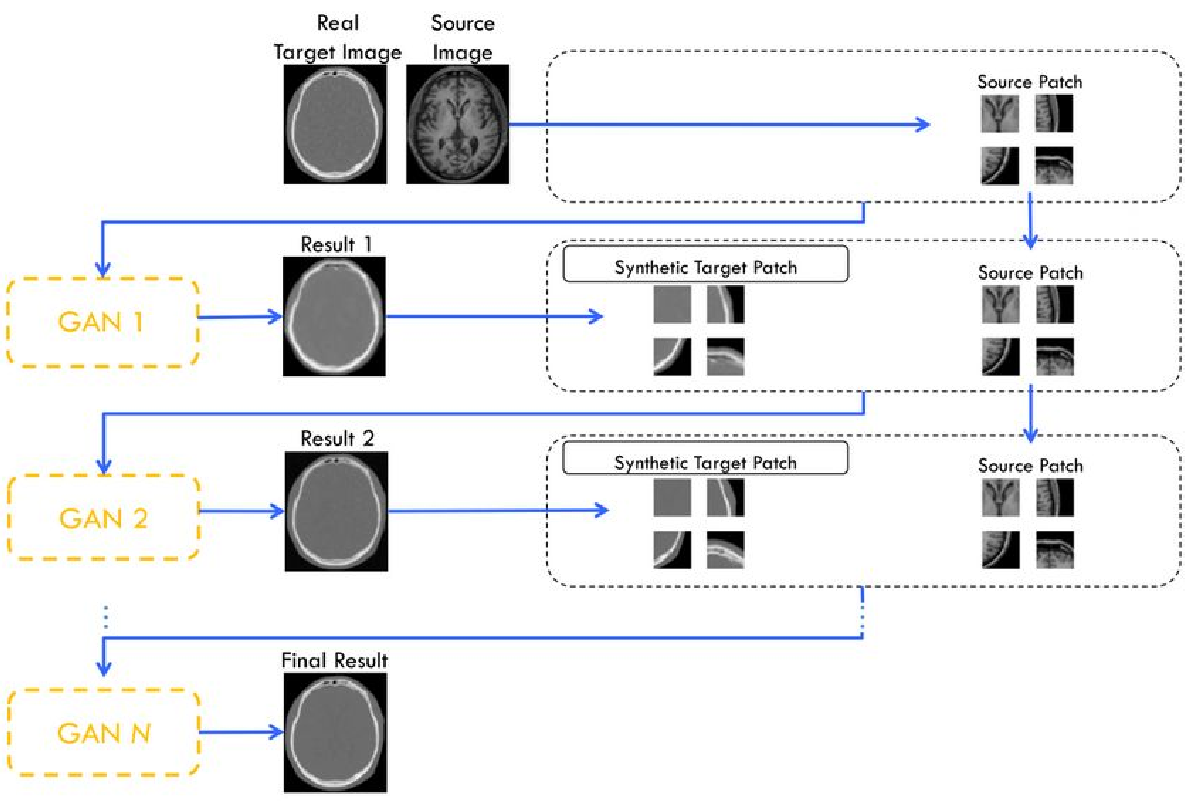
ResGAN architecture from the authors' paper. Notice the purple skip connection

Practically the input-output skip connection reminds the generator that the output is very similar to the input, an information that is lost in the deepest layers of the generative network. Training will be faster than the Adv. + GDL model because the generator just needs to learn epoch by epoch how to correct the output of the last convolutional layer in such a way that, added to the PD-weighted source image, a T2-weighted real target look-alike will be generated.

Model	Best epoch on validation set	MAE on test set	PSNR on test set
ResGAN	40/42	0.101 (0.02)	31.229 (2.154)

VIII. Experiment 3: ResGAN + Auto-Context Model

The authors applied the Auto-Context Model to refine the results by training several GANs iteratively, each one taking in input both the source patch and the synthetic target output of the previous GAN. This is a way to make the model context-aware during training as the last generated patch is concatenated as a second channel with the initial input. The rest of the architecture is the same of ResGAN.
In my own project only two ResGANs are used because of the time limits on available GPUs. The output of the second GAN is considered as the final result.



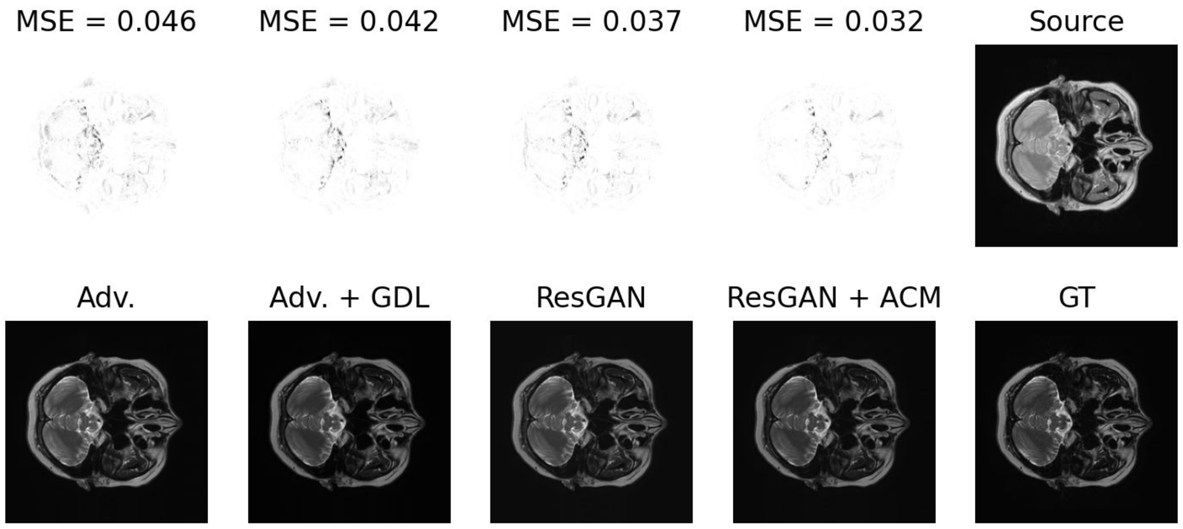
ACM architecture from the authors' paper

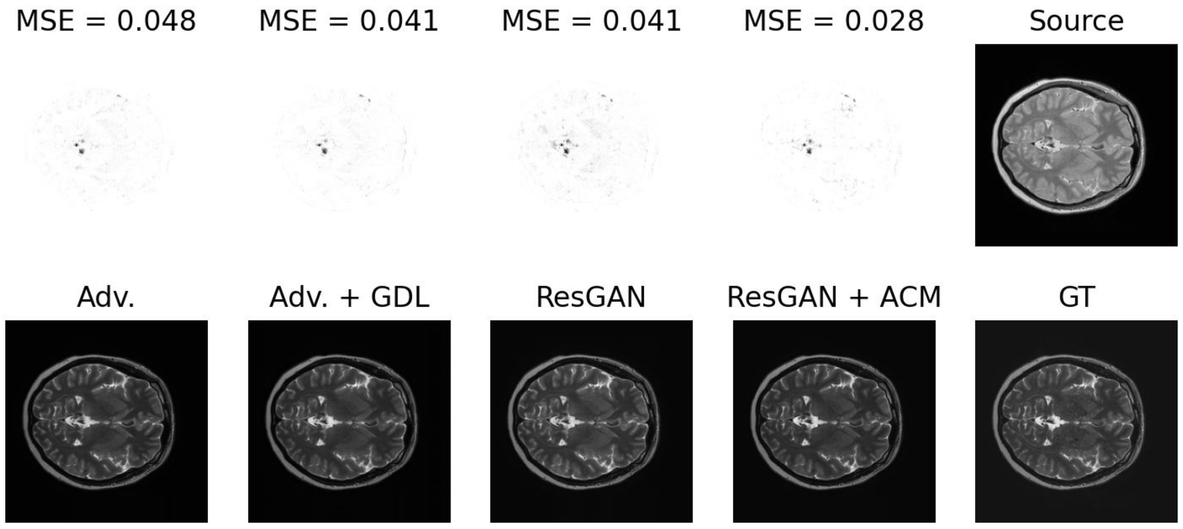
Model	Best epoch on validation set	MAE on test set	PSNR on test set
ResGAN + ACM	40/42	0.102 (0.025)	31.36 (2.232)

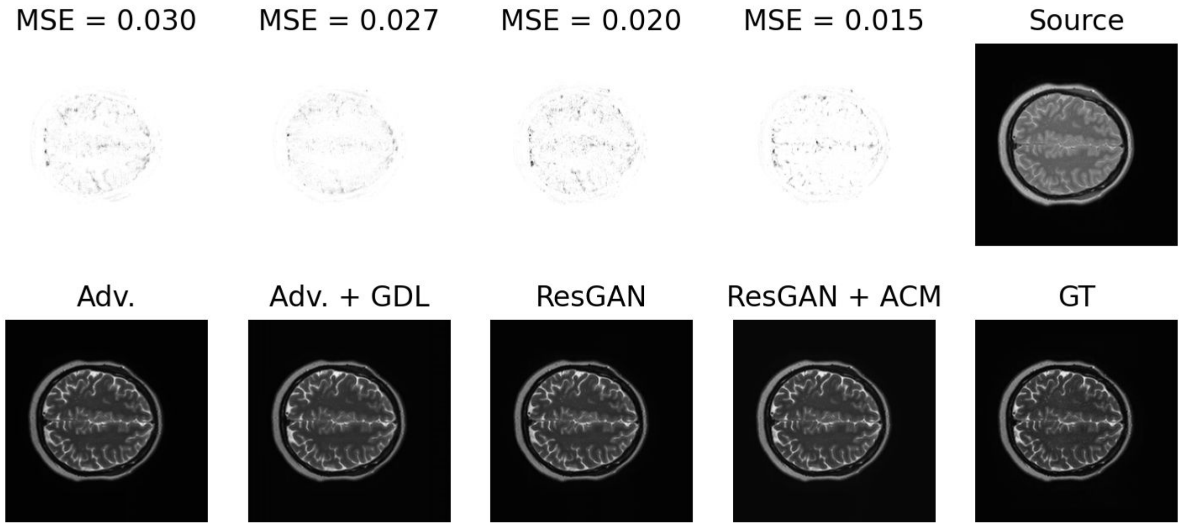
IX. Generated images and difference maps on test set

MSE is computed on a single slice, not on the whole subject's volume.

Horizontal scans



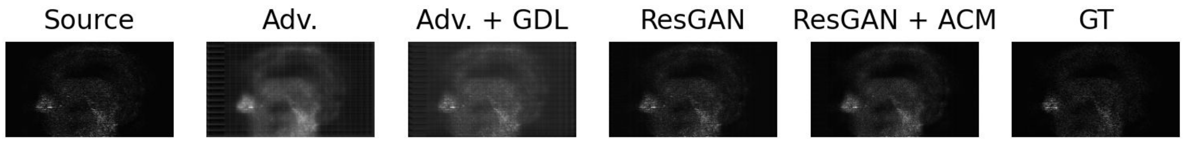


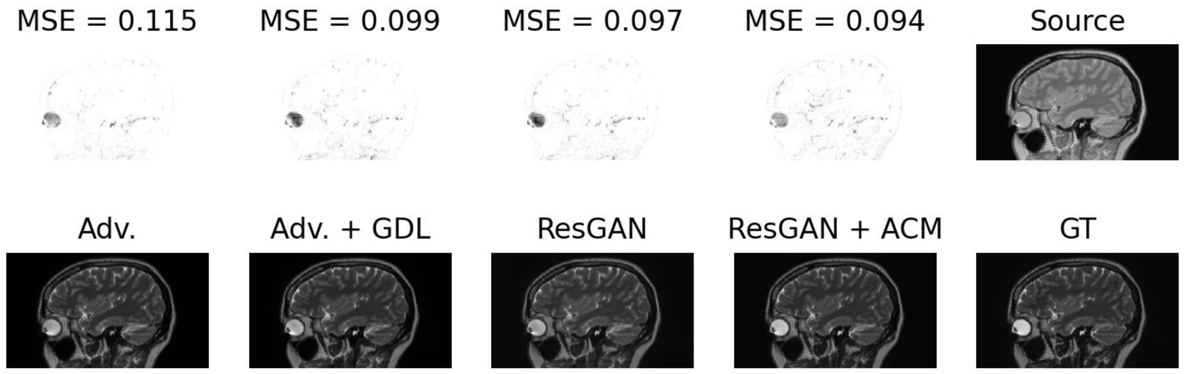


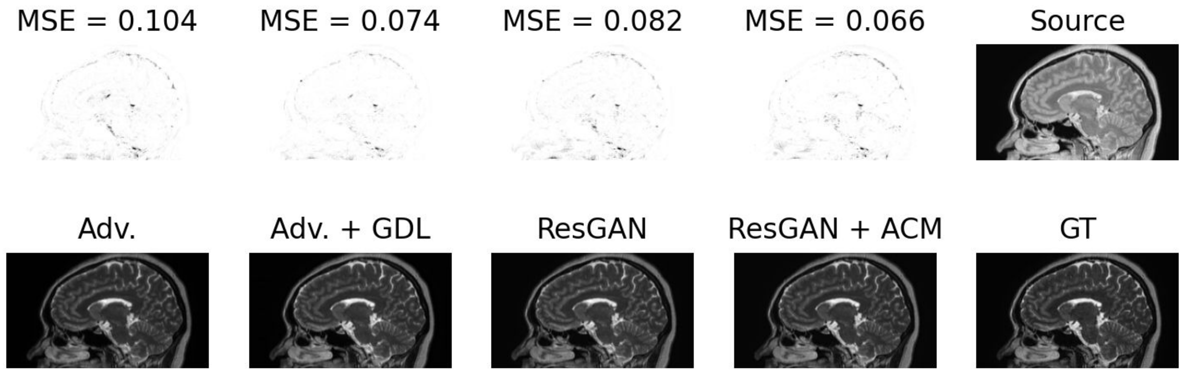
Redemptor Jr Laceda Taloma
1704899

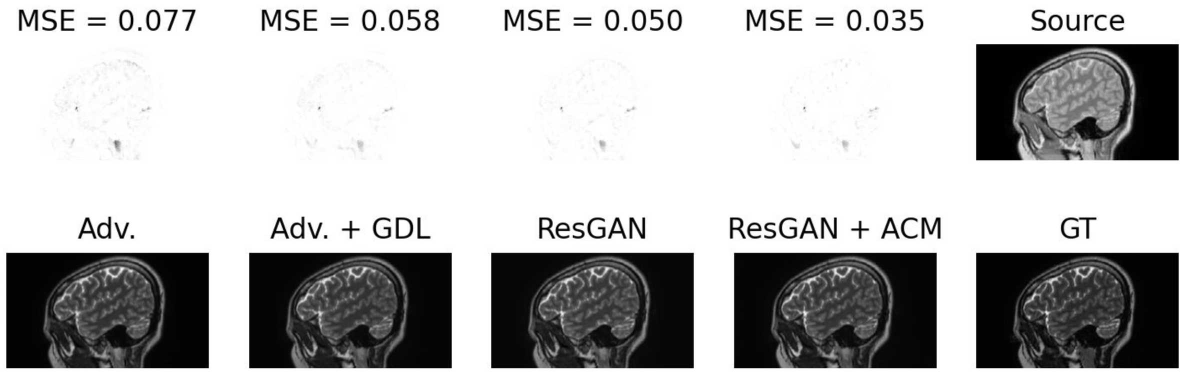
Sagittal scans

This row of images does not have any biomedical significance, but it shows well the contribution of GDL to Adv. and of the skip connection to Adv. + GDL.



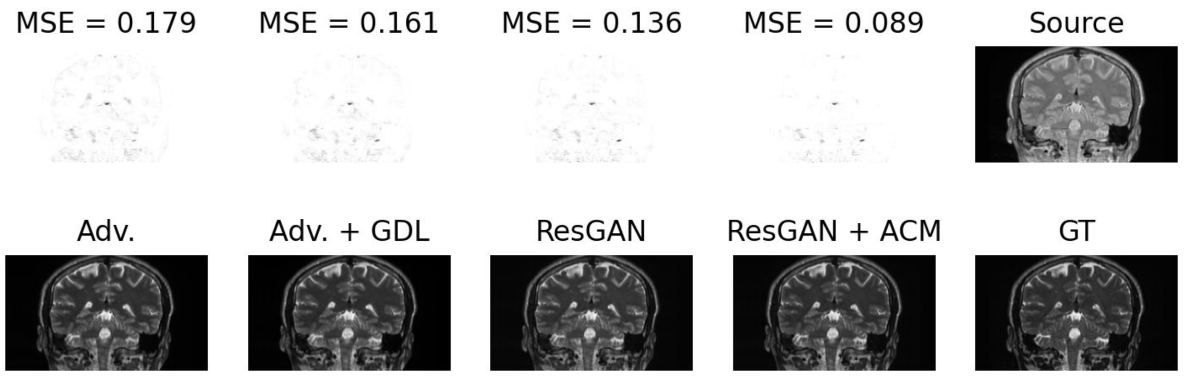


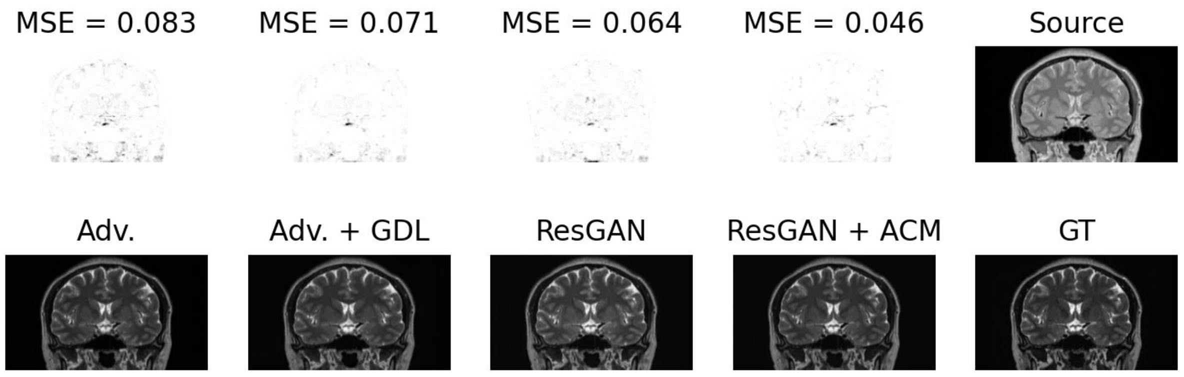


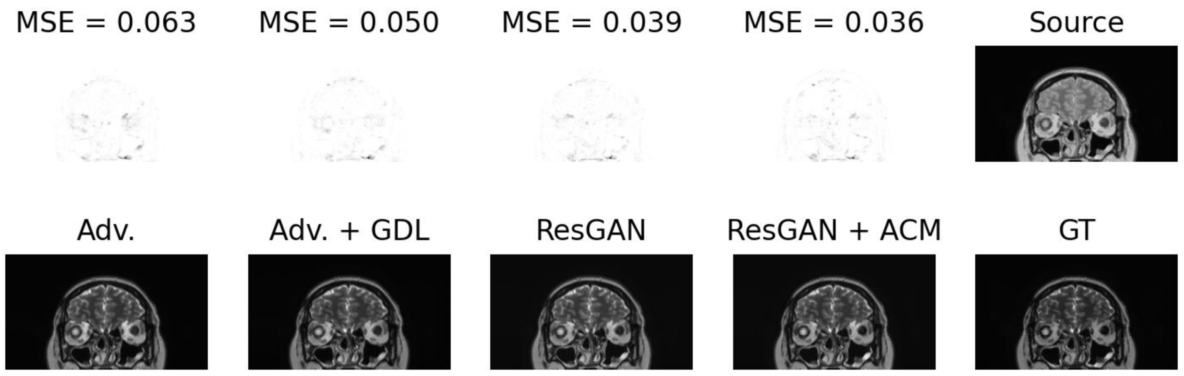


Redemptor Jr Laceda Taloma
1704899

Coronal scans







You can zoom on same slices and others in this Google Colab notebook:
https://colab.research.google.com/drive/18id0T3ocpOfD2m5g1DVA3_sw0uODzIC8?usp=sharing

X. Conclusions

This work repeated the authors’ experiment for estimating a target image from a source image of different modalities via adversarial learning. Moreover, the quality of the generation is enhanced by image gradient difference loss, especially along the borders of the brain. The alignment and the high similarity between PD-weighted and T2-weighted images in the IXI brain dataset is exploited through an input-output skip connection which accelerates training and alleviates the noise due to patch-based reconstruction. Finally, the performance is iteratively improved using 2 iterations of Auto-Context Model, thus implementing a context-aware deep convolutional adversarial network. One may try to explore further ACM iterations.

XI. Source code available on GitHub

https://github.com/brakes312t4/3D-DCGAN-medical-image-synthesis