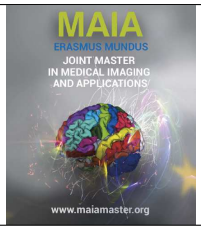
Medical Imaging and Applications

Master Thesis, August 2020



A Qualitative and Quantitative Analysis of state of the art Techniques for MRI Brain Image Synthesis

Pierpaolo Vendittelli, **Supervisors**: Arnau Oliver, Xavi Lladó

Universitat de Girona, 17003, Girona, Spain

Abstract

Medical Image synthesis is nowadays a crucial topic for reducing the costs and the acquisition timing of the images. By reducing these two important factors, much more patients can be tested in less time and get diagnosis faster. Among all the imaging techniques, MRI is one of the most used, therefore, the purpose of this research is to implement, test and analyze the different state of the art techniques usually used for brain image synthesis on two public datasets (WMH and BraTS), trying so to discuss each of them according to two different problems: synthesizing FLAIR sequences starting from T1-Weighted sequences, and the opposite. In this work three main families of architectures are tested, **Deep Convolutional Neural Network**s (DCNN) such as Unet and ResUnet, **Generative Adversarial Networks** (GANs) and **Cycle Generative Adversarial Networks** (CycleGANs). Experimental results performed on both the datasets, showned that the task of synthesizing FLAIR sequences from T1-Weighted sequences was easier than the opposite, and furthermore, it was found that a complex architecture such as CycleGAN was performing worse that more simple architectures both when synthesizing FLAIR and when synthesizing T1-Weighted.

Keywords: Image Synthesis, Brain MRI, Deep Learning, GANs, Domain adaptation

1. Introduction

Medical Imaging is the process of representing the exterior or the interior of a body through visual representation (images) which will be used for clinical analysis and medical intervention. There exist different techniques of acquisition using different physical basis, such as Ultrasounds, Positron Emission Tomography, Radiography, Computed Tomography, Magnetic Resonance Imaging etc. each one with a different objective and used for different purposes (i.e. screening or treatment). Among these, Magnetic Resonance Imaging (MRI) is a technique generating detailed images of the organs and tissues in the body with the usage of a magnetic field. The configuration of the machine allows the acquisition of multiple sequences each one highlighting different properties of the analyzed tissues. The most common sequences are T1-Weighted, PD, T2-Weighted, FLAIR and DWI. The acquisition of these different sequences of the same organ is highly expensive in terms of resource and time. Thus, during the last years research is being driven towards the automatic generation of multiple sequences from another one.

Image synthesis is indeed the process of creating new images from some form of image description which can be either noise or another image. In the medical domain this leads to creation of new data useful to fulfil some important tasks such as domain adaptation (Perone et al., 2019) or improving lesion segmentation (Salem et al., 2019) , as well as data augmentation (Shin et al., 2018) and modalities generation (Lee et al., 2019).

As it will be better described in Section II, since image synthesis is a really broad topic, a lot of different strategies are being developed during the years. These strategies can be divided in two main areas which are: traditional methods (Freeman et al., 2000) and Deep Learning techniques (Vemulapalli et al., 2015), with the exploit of Adversarial Learning techniques, these last (GANs) being a really hot topic nowadays because of the impressive performance they shown in multiple applications (Wang et al., 2020).

Through the usage of advanced Deep Learning techniques, this master thesis project is focused on analyzing and developing strategies for synthesizing Brain MRI images. More specifically, we will focus on doing a qualitative and quantitative evaluation of different and recent state of the art techniques to perform image synthesis, studying the behavior and proposing improvements on the analyzed approaches including standard convolutional neural networks approaches such as Unet and ResUnet in 3D and more advanced Adversarial approaches both in 2D and 3D .
The main tasks we will cover are two: the synthesis of FLAIR images having as input T1-Weighted images, and the opposite problem, the synthesis of T1-Weighted images having as input FLAIR. One of the main difficulties of these tasks is that when a lesion is present in the brain, while it appears clear in the FLAIR modality, in T1-Weighted it can be confused with the gray matter because of its intensities value, although usually lesions in brain appears close (around) the ventricles and the gray matter is the outside part of the brain. Figure 1 shows a clear example of this problem which can confuse neural networks and produce inconsistent results. The green circle on the left represents how the lesion appears in FLAIR images, while on the right how it appears (and can be confused with gray matter) in T1-Weighted images. The rest of this paper is organized as follow: *Section II* gives an overview of the most advanced techniques for Image Synthesis which were used as an input to the work, *Section III* describes the material used and *Section IV* introduces and discuss the background and the implementation of each approach. *Section V* presents a commented analysis of the experiments conducted for each of the developed architectures while *Section VI* discuss the experimental results obtained. In the end, *Section VII* offers a conclusion to the project mentioning some interesting approaches for future work.

2. State of the art

In this section we discuss all the most advanced strategies for image synthesis in general and proceed then analyzing more in depth state of the art techniques related to brain MRI image synthesis and cross-domain adaptation.

2.1. Image synthesis

As mentioned, image synthesis is the process of creating new images starting from some sort of descriptor. It is a really broad topic and a lot of different strategies have been developed over the last years to solve different problems, from generating new lesions for improving segmentation (Salem et al. (2019)) to pure image synthesis (Liu et al. (2018)). We can group these strategies in two main categories which respectively are: Traditional Methods, and Deep Learning techniques, which usually use Adversarial Learning techniques such as GANs.(Yi et al., 2019)

2.1.1. Traditional Methods

Traditional methods are usually atlas/multi atlas-based methods which are able to estimate the intensity distribution with a good approximation and relatively fast as shown by (Miller et al., 1993), and later on (Lauritzen et al., 2019). Atlas image synthesis usually consist in registering the image with a set of co-registered image pairs along with gold-standard segmentation masks.
Other methods include regression methods (Jog et al., 2013) where the synthesis is produced by a regression forest algorithm trained with paired data of both input and target modality.

2.1.2. Deep Learning Techniques

Since the introduction of Generative Adversarial Models by (Goodfellow et al., 2014), image synthesis shifted towards the adversarial paradigm (Xiang et al., 2018), (Nie et al., 2018), (Hiasa et al., 2018) as an example. This because of the privacy issues related to medical protocol and due to imbalanced dataset (because of the lack of positive cases for each pathology). Although not medical imaging related, Snell et al. (2017) replaced the pixelwise losses (Mean Absolute Error - Mean Squared Error) with perceptual losses such as structural similarity index (SSIM - multi scale SSIM) proving to achieve better results when reconstructing the images. In Yi et al. (2019) an exaustive review on the usage of GANs in medical imaging is presented. In particular, it was shown that to tackle the cross modality problem, researchers tend to develop architectures based on the well know pix2pix framework for co-registered data, and on the CycleGAN framework for unregistered data.

Hiasa et al. (2018) used the Gradient Correlation similarity metric as a Gradient-consistency loss between real and generated images to improve the accuracy at the boundaries, while on the other hand Zhang et al. (2018), tackled the volumetric shape problem by adding a shape-consistency loss in order to constrain the geometric invariance of the generated data, using two networks to segment each modality and provide the necessary semantic labels for each modality.

GAN can be used for generating T2-Weighted scans from T1-Weighted scans as proved from Dar et al. (2019), through the usage of cGAN and pGAN, while Yang et al. (2018) uses cGAN for generating FLAIR scans from T1-Weighted images. These works are based on the original work of Zhu (2017) and both provide 2D implementation like most of the cited papers. Yu et al. (2018) provides a method for a 3D conditional GAN which aims not only to eliminate discontinuities in the sagittal and coronal direction due to the synthesization sliceway, but also try to improve the generation

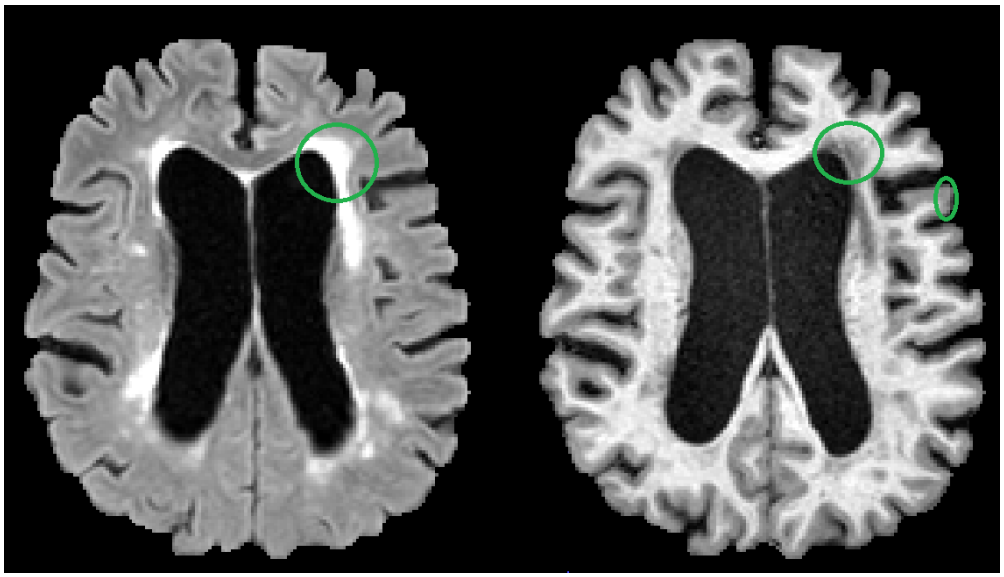


Figure 1: Example of FLAIR modality (on the left) and T1-Weighted modality (on the right). As is possible to see, in FLAIR, the lesions present around the ventricles (green circle) are really bright compared to the rest of the image, while in T1-Weighted image, these brightness is not found. Instead, the lesions appears to have a similar intensity value to the gray matter (green circle)

of synthetic FLAIR images from T1- Weighted images by introducing a global non linear mapping and a local linear mapping from T1-Weighted images to FLAIR. The global mapping determines the similarities from the synthetic image and FLAIR, while the local mapping improves the local details from T1-Weighted images.

3. Materials

The experiments done in this master thesis were conducted on the White Matter Hyperintensities Segmentation Challenge dataset (WMH) and on the Brain Tissue Segmentation Challenge dataset (BraTS) both organized by MICCAI.

3.1. WMH dataset

WMH dataset (MICCAI (2017)) is composed of 60 cases, coming from three different MRI scanners, property of the Vrije Universiteit of Amsterdam.
The first scanner is a 3T Philips Ingenuity, providing 3D T1-Weighted images with Repetition Time(TR)/ Echo Time (TE) of 9.9ms /4.6ms and 3D FLAIR images with Repetition Time(TR)/ Echo Time (TE) / Inversion Time (TI) of 4800ms /279ms /1650 ms.
The second scanner is a 3T GE Signa HDxt, providing 3D T1-Weighted images with Repetition Time(TR)/ Echo Time (TE) of 7.8ms /3.0ms and 3D FLAIR images with Repetition Time(TR)/ Echo Time (TE) / Inversion Time (TI) of 8000ms /126ms /2340 ms.
The third scanner is a 1.5T GE Signa HDxt, providing 3D T1-Weighted images with Repetition Time(TR)/ Echo Time (TE) of 12.3ms /5.2ms and 3D FLAIR images with Repetition Time(TR)/ Echo Time (TE) / Inversion Time (TI) of 6500ms /117ms /1987 ms.

As mentioned, each case presents MRI scan in 3D T1-Weighted and 3D Fluid Attenuated Inversion Recovery (FLAIR) modalities (see Figure 2). In addition, for each patient is given the brain mask as well as the manual annotation of the lesions.

3.2. BraTS dataset

BraTS dataset (MICCAI (2018)) is composed of 285 cases, coming from various scanner from 19 institutions. Each case presents mostly 3T MRI scan in T1-Weighted, T1-Gadolinium (T1Gd), T2-Weighted and FLAIR modalities. To keep the experiments similar between the two datasets, only T1-Weighted and FLAIR modalities were used and Figure 2 as well, shows an example of the dataset.
All the images were pre-processed and pre-registered with a common resolution of $1mm^3$.

4. Analyzed Methods

During the project, several architectures were developed and designed in order to have a qualitative and quantitative analysis of the different techniques for image synthesis. Most of the networks presented here are 3D architectures in which the input usually is a cubic patch of size $32 \times 32 \times 32$ representing a part of the MR volume, while the last 2 are a 2D version in which the input is a square of the size of the whole image. Here we briefly present the various architectures used and give details about the evolution of the project step by step.
There are six different architectures (organized in various setups) that are summarized in Table 1.

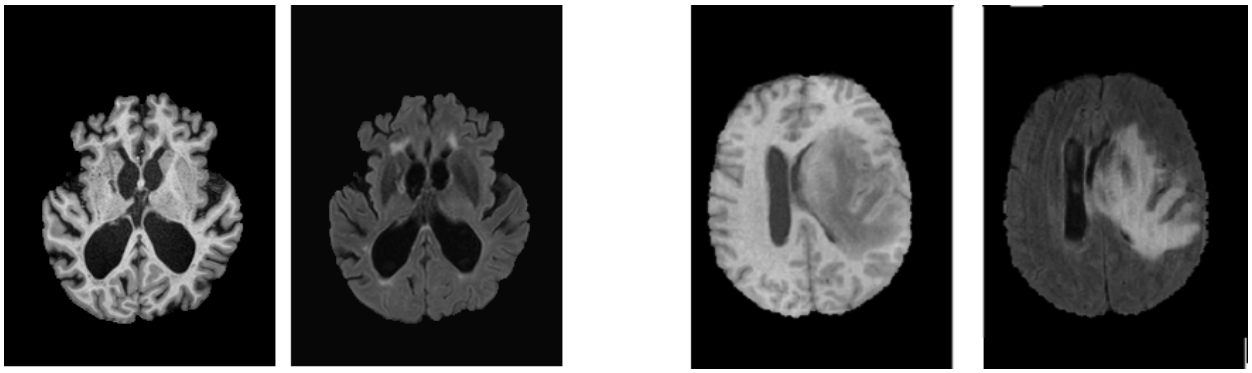


Figure 2: **Left**: WMH dataset, Axial View of MRI scan of one patient. **first image**: T1-Weighted Image, **second image**: FLAIR Image. **Right**: BraTS dataset, Axial View of MRI scan of one patient. **third image**: T1-Weighted Image, **forth image**: FLAIR Image.

Table 1: Summary of the analyzed architectures.

Name	Typology	Dataset	Experiment
Unet	3D	WMH	T1-FLAIR
		BraTS	FLAIR - T1
ResUnet	3D	WMH	T1-FLAIR
		BraTS	FLAIR-T1
GAN	3D	WMH	T1-FLAIR
		BraTS	FLAIR-T1
GAN	2D	WMH	T1-FLAIR
CycleGAN	3D	WMH	T1-FLAIR
			FLAIR-T1
CycleGAN	2D	WMH	T1-FLAIR
			FLAIR-T1

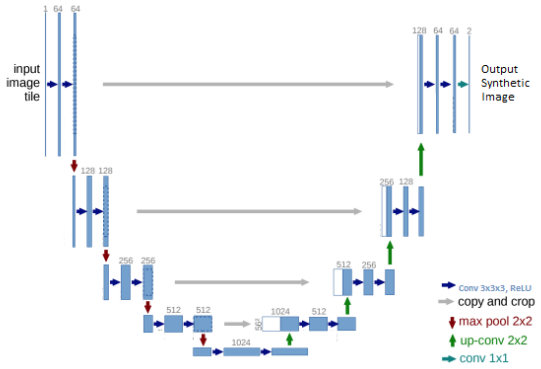


Figure 3: Unet Architecture: as mentioned we can appreciate the U-shape composed by an **encoder** receiving the image in input and extracting features through convolutions and pooling to the latent space and the **decoder** which reconstruct the image.

4.1. Unet shaped architectures

4.1.1. 3D Unet

The first approach which was followed was inspired by the Unet encoder-decoder architecture, proposed by Ronneberger et al. (2015) in the context of biomedical image segmentation (microscopy imaging). The Unet is a Convolutional Neural Network with an Encoder - Decoder shape with the addition of skipping connection between layers. As shown in Figure 3, the Encoder part of the network is composed by 3 Convolutional Layers, each followed by a rectified linear unit (Relu) activation layer and a $2 \times 2 \times 2$ Max Pooling layer with stride 2 for downsampling. In addition, there is a forth Convolutional Layer followed only by a Relu activation layer which drives the input image into the latent space.

The Decoder part is composed by 3 upsampling layers (transpose convolution for upsampling + convolution as pooling) followed by skipping connection layers between the input and the output. For this network each of the filters used is a cube of size 3×3 while the number of filters in each layer varies from 32 to 256 in the encoder part and vice-versa in the decoder part.

4.1.2. 3D ResUNet

Another architecture which was implemented and tested alone in the same set of experiments was an evolution of the 3D Unet, the 3D ResUnet. The main difference is that, to address the vanishing gradient problem, 3D ResUnet uses Residual Blocks in the encoder parts of the Unet in order to exploit the residual connections at each block. This allows a better flow of the gradient through the network layers.

The flow of the architecture is similar to the Unet presented earlier with the addition of Residual Blocks.
A Residual Block is composed by n repetitive layers, each of which presents a 3D Convolutional Layer and a Normalization Layer followed by a Leaky rectified linear unit (Leaky ReLU). The input of the first Convolutional Layer in the block is concatenated to the output of the activation layer.
This architecture has 5 downsampling layers in the Encoder part followed by the same number in the Decoder part as shown in Figure 4. For this network each of the filters used is a cube of size $3 \times 3 \times 3$ while the number of filters (k) varies from 16 to 128 in the Encoder part

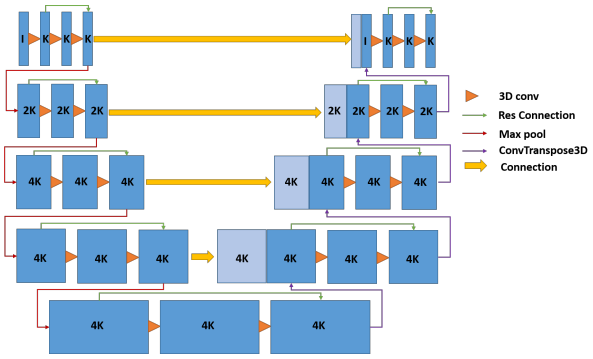


Figure 4: ResUnet Architecture: The shape is similar to the already presented Unet, with the difference of the presence of the residual connection (green arrows). These residual connection help the gradient to backpropagate smoothly so that to make the learning easier for the network.

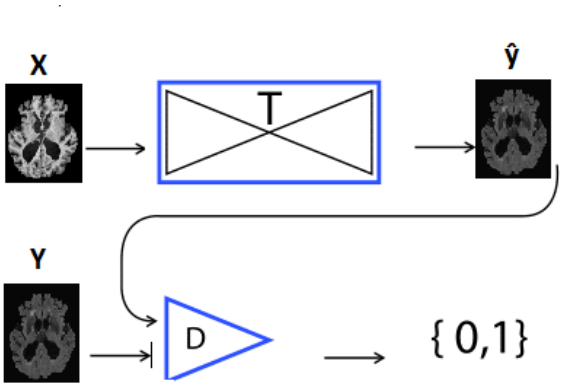


Figure 5: GAN Architecture: T has the shape of an Encoder-Decoder CNN and is one of the two networks presented earlier, while D has the shape of a CNN. The objective of T is to generate images belonging to the target domain, receiving as input images belonging to the input domain. The objective of D is to correctly classify the generated images as "fake" and the real images as "real".

and vice - versa for the decoder part.

4.1.3. Loss Function and Setup

The two networks presented earlier were trained and evaluated in the same modular architecture. The loss function used for training the networks is the L1 Loss, a criterion which measures the Mean Absolute Error (MAE) between each element in the generated set $\hat{y}$ and target y. As shown from Zhao et al. (2015), L1 Loss is usually one of the most used loss functions when it comes to synthesis because it produces less blurried results when compared with L2 (MSE) Loss. It is defined by the following equation:

$$L_1 Loss = \mathbb{E}_{y,\hat{y}}||y - \hat{y}||_1$$

4.2. 3D Generative Adversarial Networks

In 2014 ,Goodfellow et al. (2014) proposed a new type of Neural Network, Generative Adversarial Networks. GANs are composed by two different networks collaborating in an adversarial training. The first network is usually called Generator while the second network is usually called Discriminator. In the first version of GAN, the Generator is so called because it takes as input a noisy distribution and through the training phase produces an output similar to the target.
During the recent years, GANs have been widely used for multiple problems, from Image generation, to Style Transfer, to Domain adaptation. In medical Imaging GANs are used mostly for Segmentation (SeGAN) or for Image translation (MedGAN, cGAN). Since this project is about image translation, we will refer here the Generator as Translator.

As mentioned, GANs can be useful in generating images from random noise distribution, but they can also be used when as input there is an image (see Figure 5). In this setup the generator takes as input the image in the input domain and tries to generate an image in the target domain, as in a standard GAN. A variation of this architecture can appear when the generator takes as input the pair input target, and the output of the network becomes directly conditioned not only form the target, but also from the input. In this case the architecture is the well-known so called Pix2pix, proposed by Zhu (2017).

Figure 6 shows in detail the architecture. The generator takes as input the pair input-target, and according to that, learns a mapping function to translate the input into the target domain. The discriminator instead, tries to understand whether the image it receives as input is the real target or is the generated translation.

4.2.1. Translator and Discriminator

With these preliminaries the Translator is one of the two previously presented networks (Unet/ ResUnet), and it tries to map a source domain image $x \sim p_{source}$ into its ground truth $y \sim p_{target}$.
The Discriminator has the role of a binary classifier, classifying so, if the output of the Translator is an image belonging directly to the target domain (real) or if it comes from another domain and it has been adapted (fake). The training procedure is said to be an adversarial fashion since during each training step, first the discriminator classifies between fake and real samples and then the translator tries to produce a better output.

As shown in Figure 7, the Discriminator D has the shape of a binary classifier, with 3 Convolutional Layers followed each by a rectified linear unit (ReLU) Layer and a Max Pooling Layer. After the Convolutional Layers a series of Fully connected Layers follows each activated by a RelU function.
For the Convolutional Layers the kernel sizes vary from $7 \times 7 \times 7$ in the first layer to $5 \times 5 \times 5$ in the second layer to $3 \times 3 \times 3$ in the third, while the number of filters goes from 64 to 256. The fully connected layers have a

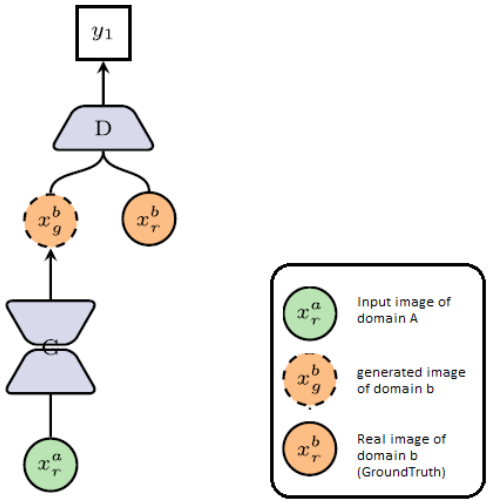


Figure 6: Pix2pix architecture (inspired by the paper Yi et al. (2019)) The difference with a normal GAN is that exists a condition on the input with the target. The generator receives the pair (input - target) as input rather than the only input.

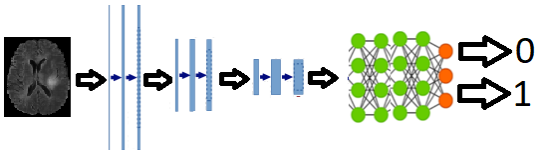


Figure 7: Discriminator architecture: 3 downconvolutional layers + a series of fully connected layers to binary classify the generated image as real or fake.

number of neurons varying from 256 to 1024 in the first three layers, and vice-versa in the last three.

4.2.2. *Loss Function*

Being GANs an architecture composed of two different networks, the Loss Function definition as well as the training setup represents the core of this architecture. The total loss function is composed of two parts: and adversarial part and a feature matching part.

As said, the Discriminator aims to correctly classify the real and the synthetic patches while the Translator aims to fool the Discriminator. Both the networks work in an adversarial fashion following thus, a min - max optimization task on the Adversarial Loss Function:

$$\mathscr{L}_{GAN} = \mathbb{E}_{y \sim p_{data(y)}}[logD(y)] + \mathbb{E}_{x \sim p_{data(x)}}[log(1 - D(T(x))]$$

where the Discriminator D tries to maximize it and the Translator T tries to minimize it.
Since in image to image translation tasks might happen that the translated samples do not produce consistent results but still the Translator fools the Discriminator, a Feature Loss is introduced. This Feature Loss is usually the well known L1 loss (Mean Absolute Error) and it minimizes the differences between the ground truth y and the translated samples $T(x)$ multiplied by a factor λ.

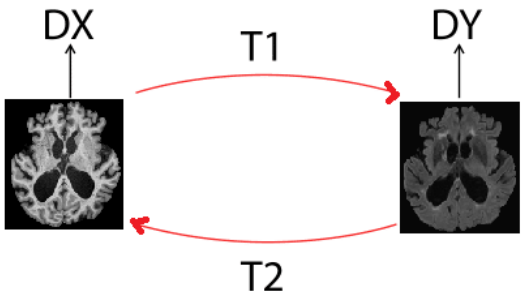


Figure 8: CycleGan scheme: this architecture is formed by 2 Translators (T1 and T2) and 2 Discriminators (Dx and Dy).

The Total GAN loss is therefore composed by a sum of the previous parts as follow:

$$\min_T \max_D \mathscr{L}_{GAN} + \lambda L1$$

with $\lambda = 30$ since it provided a good trade-off in our experimental results. Values of λ smaller than 30 produced less meaningful results, while higher values did not improve with respect to this empiric value.

4.2.3. *Label Smoothing*

Some of the problems that the training of Generative Adversarial Networks can produce, is overfitting and overconfidence. This can occur especially in the Discriminator, during the classification task between the real samples and the generated ones. To overcome this problem, a regularization factor (Wong) during the training of both the Translator and the Discriminator is added to the labels. The Discriminator is originally trained to identify as 1 the patches (or images) coming from real target distribution, while as 0 the patches (or images) coming from the fake (generated) distribution. This is done by replacing the the one-hot encoded label vector 1 as a random distribution vector with values between 0.7 and 1.0 and by replacing the one-hot encoded vector 0 as a random distribution vector with values between 0.0 and 0.3. This smoothing on the labels has the effect of making the Discriminator (in classifying) and the Translator (in generating) less confident when producing its output, reducing so, the possibility of overfitting.

4.3. *CycleGAN*

Zhu (2017) et al. introduced a new method for paired and unpaired image to image translation called CycleGAN. Figure 8 gives a clear view of this CycleGAN architecture. In this configuration there are 2 Translators and 2 Discriminators. The role of the translator is to map the input distribution into the output, while the discriminators have to binary classify the real and the generated samples. Since there are 4 networks, the Translator 1 T_1 will take in input the modality x trying

to generate the modality y, while Translator 2 T_2 does the opposite, generating x from y. Discriminator 1 D_y binary classifies y and $T_1(x)$, while Discriminator 2 D_x binary classifies x and $T_2(y)$.

4.3.1. CycleConsistency Loss and Identity Loss

In addition to this, a cycle consistency criterion (Cycle Consistency Loss) is added. Cycle consistency is based on the fact that one of the two Translators might learn to map the input image into a fixed output distribution, fooling the Discriminator but not producing a representative result. This happens when a Translator (or both) produces a result which is in the target distribution but is not the actual translation of that precise input. To avoid this, Translators are trained to be consistent with each other by imposing $T_2(T_1(x)) \approx x$ and $T_1(T_2(y)) \approx y$.
Furthermore, the goal is also to teach T_1 and T_2 to map the input into the output only when they are different, and to do nothing when in input is the output (Identity Loss). This is done by feeding the images already in modality x to the Translator 2 (T_2) which translates from y to x, because the CycleGAN should understand that the input is already in the correct domain. Therefore, unnecessary changes are penalized. Figure 9 shows the diagram of both the cycle consistency criteria and the identity loss.
Both the Identity Loss and Cycle Consistency Loss are L_1 Losses.

$$\mathscr{L}_{cyc} = \mathbb{E}_{x,\sim p(x)}\|T_2(T_1(x)) - x\|_1$$

$$\mathscr{L}_{id} = \mathbb{E}_{y,\sim p(y)}\|T_1(y) - y\|_1$$

For this configuration, the Total loss is defined as a weighted sum of the aversarial part plus the cyclic consistency part and the identity part as follow:

$$\mathscr{L}(T_1, T_2, D_1, D_2) = \mathscr{L}_{GAN} + \lambda\mathscr{L}_{cyc} + \lambda\mathscr{L}_{id}$$

where $\lambda > 0$.

4.4. Implementation Details

All the codes done in this master thesis work were developed in Python through the Pycharm IDE using Pytorch and the **niclib** library while all the newtorks were trained with an nVidia Titan V GPU available in the Vicorob Lab.

5. Results and Discussion

5.1. Evaluation Metrics

For each experiment the results were evaluated both visually and using different quantitative measures (MSE, SSIM, PSNR), which are usually used when evaluating Image synthesis.

- MSE : Mean Squared Error, defined as

  $$MSE = \mathbb{E}_{y,\hat{y}}\|(y - \hat{y})^2\|_2$$

  where a lower value means better result.

- PSNR: Peak Signal to Noise Ratio, defined as

  $$PSNR = 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right)$$

  where MAX_I is the maximum pixel value in the image and a higher value means better result.

- SSIM: Structural Similarity Index, a perceptual difference between two similar images, defined as

  $$SSIM = \frac{(2\mu_x\mu_y + c_1) \cdot (2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1) \cdot (\sigma_x^2 + \sigma_y^2 + c_2)}$$

  where μ represents the average, σ^2 represents the variance, σ represents the covariance and $c_1 and c_2$ are variables to stabilize division with weak denominators. For SSIM a result of 1.0 means that the two images are identical.

While conducting experiments, these metrics are calculated imagewise, while the average as well as the standard deviation and max/min values are calculated per fold.
To have a correct evaluation of the methods we use a four fold cross-validation strategy on the two datasets presented in Section 3.

The experiments can be summarized in two subsection, one grouping the results obtained synthesizing **FLAIR** from **T1-Weighted** images, and the other grouping the results for the other way around: synthesize **T1-Weighted** images from **FLAIR**. In each of these subsection the results of the different approaches and dataset are analyzed and commented.

5.2. T1-Weighted to Flair

The very first experiment has the goal of synthesizing FLAIR images starting from T1-Weighted Images. For White Matter Hyperintensities dataset are conducted two different experiments, one giving to the networks only one input (the T1-Weighted image) and the other (for the architectures which allow) two inputs, which are the T1-Weighted images plus the binary mask of the lesions. For BraTS dataset, only the T1-Weighted image is given as input to the networks.

5.2.1. Unet 3D and ResUnet 3D

Table 2 show the different setup of configuration for this first experiment.The optimizer used in both the configuration is Adam (lr = 1e-4). We can see that most of the data is similar (i.e. Patience, number of samples, number of Epochs..), although results are different.

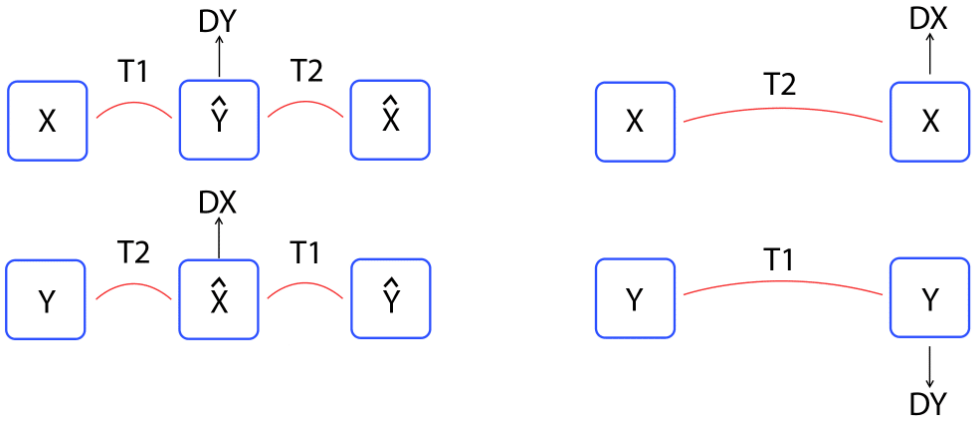


Figure 9: **Left**: Cycle Consistency Loss 1 and 2: The image in the input domain (X) is translated to the target domain ($\hat{y}$) by the first Translator (T1) and then re adapted into the original domain ($\hat{x}$) by the second Translator (T2). MSE is calculated between X and $\hat{X}$. The same process but with the opposite domain is done for the cycle consistency loss 2 (bottom left).**Right**: Identity Loss: The translators (T1 and T2) are trained to not apply any transformation to the input image when this is already in the target domain.

Table 2: Experiments setup for 3D Unet and 3D ResUnet.

Experiment	#Params	#Epochs	#Samples	Patience	Training Time
3D Unet 1 Input WMH	9.14 M	20	2000	**5**	15 m
3D Unet 2 Input WMH	9.14 M	20	2000	5	15 m
3D Unet 1 Input BraTS	9.14 M	20	2000	5	25 m
3D ResUnet 1 Input WMH	3.21 M	20	2000	5	20 m
3D ResUnet 2 Input WMH	3.21 M	20	2000	5	20 m
3D ResUnet 1 Input BraTS	3.21 M	20	2000	5	35 m

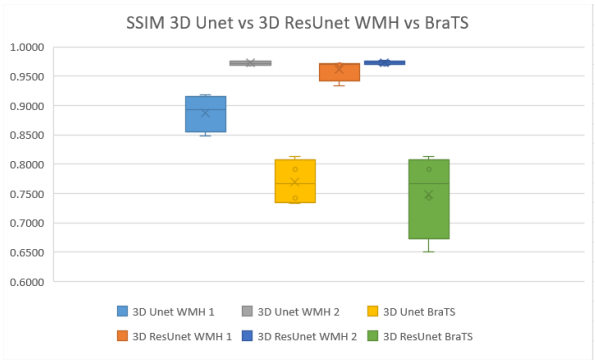


Figure 10: SSIM BoxPlot foldwise calculated on the presented configurations.

Figure 10 represent the distribution **Foldwise** of the calculated SSIM for each of these configurations through boxplot, while Table 3 instead, proposes a more complete comparison of the same configurations by reporting also the qualitative measures MSE and PSNR for both the 3D Unet and 3D ResUnet.

From Table 3 we can see that the best results using this 3D Unet architecture to synthesize FLAIR images from T1-Weighted images are achieved when the

Table 3: Comparison between the three configurations of the two architectures: MSE, SSIM, PSNR

Case	Avg_MSE	std_MSE	Avg_SSIM	std_SSIM	Avg_PSNR
3D Unet 1 Input WMH	0.0019	0.0003	0.8877	0.0323	**56.4474**
3D Unet 2 Input WMH	**0.0017**	**0.0002**	**0.9720**	**0.0034**	56.2893
3D Unet 1 Input BraTS	0.0030	0.0023	0.7699	0.1658	52.4327
3D ResUnet 1 Input WMH	**0.0014**	**0.0002**	0.9611	0.0185	57.3096
3D ResUnet 2 Input WMH	0.0017	**0.0002**	**0.9726**	**0.0034**	55.7748
3D ResUnet 1 Input BraTS	0.0027	0.0022	0.7493	0.0722	**59.3646**

dataset is the WMH and the network is fed with two inputs. Being BraTS dataset with images coming from more scanners, the network seems to not generalize enough well, and in fact, there is a high standard deviation in the SSIM with respect to the experiments done using the WMH dataset.

When comparing the results between Unet 3D and ResUnet 3D from Table 3 we can notice three main objects: when using 2 inputs for WMH dataset, the two architectures are comparable; when passing one input with WMH, the ResUnet can perform better than Unet, but for BraTS dataset, the results are equally bad; about this last configuration we can appreciate the same difficulty that the networks present foldwise to generalize between the different scanners, even though we see a higher variance when using ResUnet (Figure 10).

Figure 11 shows a comparison between 3D FLAIR (on the left), 3D ResUnet (in the middle) and 3D Unet (on the right). From these three images we can appreciate both good things, comparison between the two results and drawbacks of these architectures. The yellow circle represents the lesion around the ventricles which was correctly synthesized by both the networks, but the red circle shows a failure of both the architec-

Table 4: Different Configurations for GAN.

Experiment	#Parameters	#Epochs	#Samples	Patience	Training Time
GAN 1	13.86 M	20	1536	5	45 m
GAN 1 Soft L	13.86 M	20	1536	5	45 m
GAN 1 Soft L BraTS	13.86 M	20	1536	5	65 m
ResGAN 1	7.86 M	20	1536	5	55 m
ResGAN 1 Soft L	7.86 M	20	1536	5	55 m
ResGAN 1 Soft L BraTS	7.86 M	20	1536	5	75 m
GAN 2	10.25 M	20	1536	5	61 m
GAN 2 Soft L	10.25 M	20	1536	5	61 m
ResGAN 2	11.85 M	20	1536	5	59 m
ResGAN 2 Soft L	11.85 M	20	1536	5	59 m
2D GAN	1.85 M	200	1500	20	13 m

tures which confused a piece of gray matter as a lesion. The green circle in the end, shows one of the differences between both the architectures: although the metrics are similar for both of them, and definitely both the synthetic images look synthetic, the one generated by the 3D ResUnet architecture looks more similar to the original FLAIR by being a bit more smooth on the intensities changes with respect to the image generated by 3D Unet.

5.2.2. GAN

The natural further step in developing this family of experiments is to implement both the previous networks in an Adversarial contest and see whether there are improvements or not. As mentioned in Section 3 were implemented two versions of GAN, one 3D and another one 2D. Furthermore, two more 3D architectures were developed in order to have a more extensive comparison between 3D Unet and 3D Resunet, and in the experiments the effect of label smoothing introduced from Salimans et al. (2016) was analyzed as well. Due to the fact that for WMH dataset the results of the networks alone are better when passing two inputs to the networks, the configurations of one input were avoided for this comparison. For all the 3D configurations the Optimizers used are Adam (lr = 1e-6) for the Translator and Adam (lr = 2e-4) for the Discriminator. For the last configuration (2D GAN) the learning rate was linearly decreased after the 100th epoch and the implementation was taken from the github repository provided by Zhu (2017).
Table 4 synthesizes the different configurations tested for GAN.

Figure 12 shows a comparison of the SSIM for all the different tested configurations. Surely the worst performance are achieved by the 2D configuration of GAN (pix2pix). This might be due to the fact that the number of samples was not enough for the network to correctly learn and generalize, even for the WMH dataset. Indeed, it achieved (0.49 ± 0.19) as a mean value for SSIM. Tests for BraTS dataset for this configuration were not done, because the general trend shows that WMH performs better than BraTS, therefore we did not think that this test would be useful. More in detail about the 3D configurations. Some of the tests (GAN 2 and ResGAN 2 wuth BraTS) are missing. This because some of the preliminary experiments on the second dataset produced disappointing results, thus it seemed not useful and redundant to perform further tests. Figure 13 shows a comparison between the proposed GAN using as Translator the Unet described above (GAN 1), and a variation of the Unet (GAN 2). This variation of the Unet uses 4 Convolutional Blocks (a block is formed by conv - pooling - activation - conv - pooling - activation) and 4 UpConvolutional Blocks (a block is formed by upconv - batchnorm - activation - upconv - batchnorm - activation). It is interesting to notice how when using hard labels, the two version of GAN perform in a similar way, but the improvement achieved using Soft Labels in GAN 1 is not visible in GAN 2. Instead, the usage of soft Labels with the second vesrion of GAN worsen the results as shown in Table 5.

Table 5: GAN 1 vs GAN 2 vs ResGAN 1 vs ResGAN 2 vs 2D GAN Hard Labels vs Soft Labels WMH vs BraTS.

Experiment	Avg_MSE	std_MSE	Avg_SSIM	std_SSIM	PSNR
GAN 1 Hard	0.0015	**0.0002**	**0.9697**	**0.0120**	57.0659
GAN 2 Hard	**0.0014**	0.0004	0.9608	0.0211	58.4175
ResGAN 1 Hard	**0.0014**	0.0004	0.9356	0.0540	**59.6918**
ResGAN 2 Hard	0.0337	0.0355	0.8007	0.1609	55.5008
GAN 1 Soft	0.0016	**0.0005**	**0.9710**	**0.0107**	57.4216
GAN 2 Soft	**0.0015**	**0.0005**	0.7813	0.1454	**61.0257**
ResGAN 1 Soft	0.0017	**0.0005**	0.8851	0.1267	59.4640
ResGAN 2 Soft	0.0159	0.0064	0.7385	0.1196	48.6462
GAN 1 Soft BraTS	**0.0142**	0.0196	0.7093	**0.0149**	**52.9729**
ResGAN 1 Soft BraTS	0.0310	**0.0064**	**0.7210**	0.1196	48.6462
2D GAN WMH	2.1215	0.2245	0.4858	0.1911	37.2948

Table 6: CycleGAN configuration.

Experiment	Parameters	Epoch	Samples	Patience	Training Time
3D CycleGAN	27.72 M	20	1536	5	79 m
2D CycleGAN	13.2 M	200	1500	20	25 m

Figure 13 shows as well the comparison between the proposed GAN which uses the earlier presented 3D Resunet as Translator(ResGAN 1) and a variation of the ResUnet (ResGAN 2). This variation of the 3D ResUnet is composed by 1 Convolutional Block (presented earlier) and 3 Residual Unit (each unit is composed by a convolution - batchorm - activation) and 3 UpResBlock (each block is composed by a up convolution - batchnorm- activation) plus a final convolution block. About the BraTS dataset, Figure 14, shows that even though ResGAN 1 achieves a better result (as maximum SSIM), the variance of GAN 1 is smaller.

5.2.3. CycleGAN

The last family of the experiments done for synthesizing FLAIR from T1 Weighted images was the CycleGAN architecture, both in 3D and in 2D, and the set of configurations tested is reported in Table 6.

For the 3D CygleGAN configuration, the experiment was only one, because it was selected the best network among the 3D GANs (which ended up to be GAN 1 with SoftLabels), but with only one input to the Translators. This is due to the fact that cycle

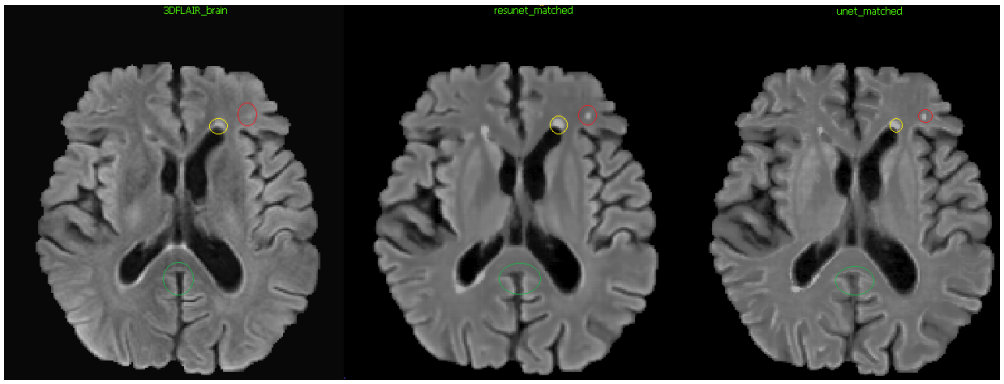


Figure 11: Comparison of two images of WMH dataset. The original FLAIR image is on the left, the 3D ResUnet synthesized image is in the middle and on the right there is the 3D Unet synthesized image.

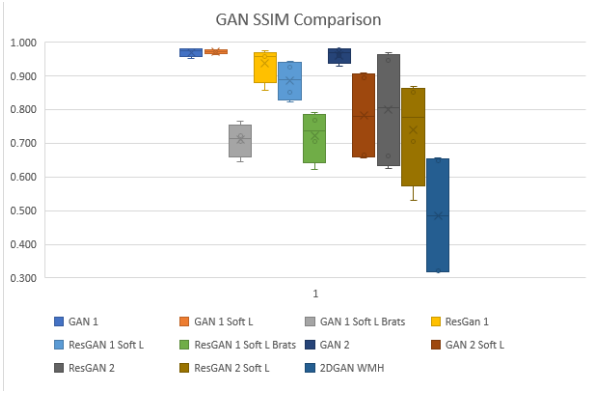


Figure 12: Results of the experiments for GAN using WMH dataset (GAN1, ResGAN1, GAN2, ResGAN2, 2D GAN, GAN 1 Soft Labels, GAN 2 Soft Labels, ResGAN 1 Soft Labels, ResGAN 2 Soft Labels) and BraTS dataset (GAN 1 Soft Labels BraTS, ResGAN 1 Soft Labels BraTS).

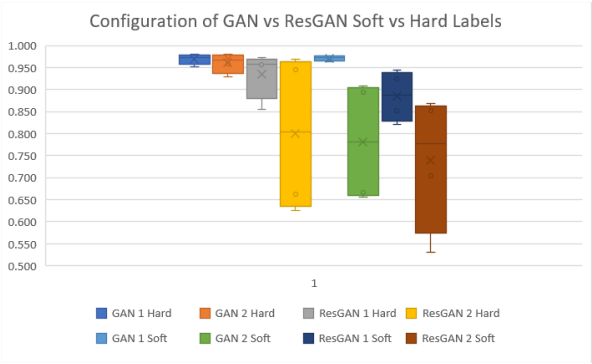


Figure 13: Comparison on WMH dataset between GAN 1, GAN 2, ResGAN 1 and ResGAN 2 while using Hard and Soft Labels.

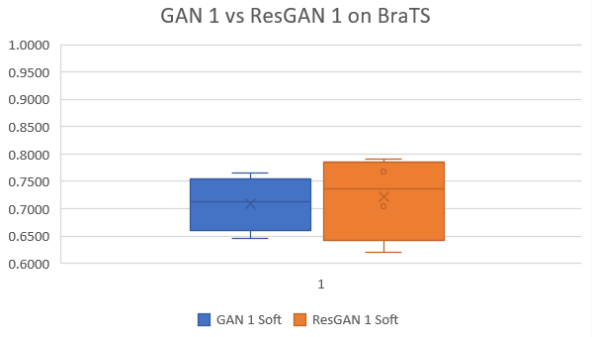


Figure 14: GAN 1 vs ResGAN 1 on BraTS dataset.

consistency needs to be respected. If T_1 receives as input the T1-Weighted images plus the lesion and it outputs only FLAIR, then when calculating the cycle consistency loss ($T_1(T_2(\text{FLAIR}))$, we need T_2 to output two values, but for the way it is designed, this was not a feasible and optimal idea. Instead, reducing the input to only the T1-Weighted images to the first translator was a considered as a more optimal solution from the programming point of view. Figure 16 shows this comparison between the 2D and the 3D implementation of the CycleGAN architecture. As is possible to see (from the numbers in Table 7 as well) the results were not good. This might be due to the fact the this architecture (3D) was too complex for the task and the number of patches was not enough for the networks. More in detail, what was possible to see from the images in result, due to this limited number of samples, and due to the cycle consistency constrain, both the networks were not able to learn any mapping function. In fact (both for 2D and 3D implementation) T_1(T1-Weighted) was really similar to T1-Weighted images. Figure 15 shows a visual comparison of most of these architectures. As is possible to see, GAN1 with the usage of soft labels (top row, third element) outperforms the other architectures, producing a result really similar to the target FLAIR, the lesions around the ventricles are well positioned and the image obtains the same contrast as a real FLAIR image. Right after it is placed ResGAN1 with the usage of soft labels (bottom row, first element), which still produces a good results, although more smoothed than GAN1, therefore details can be less appreciated. GAN2 (bottom row, second element) which from Figure 12 shows good results

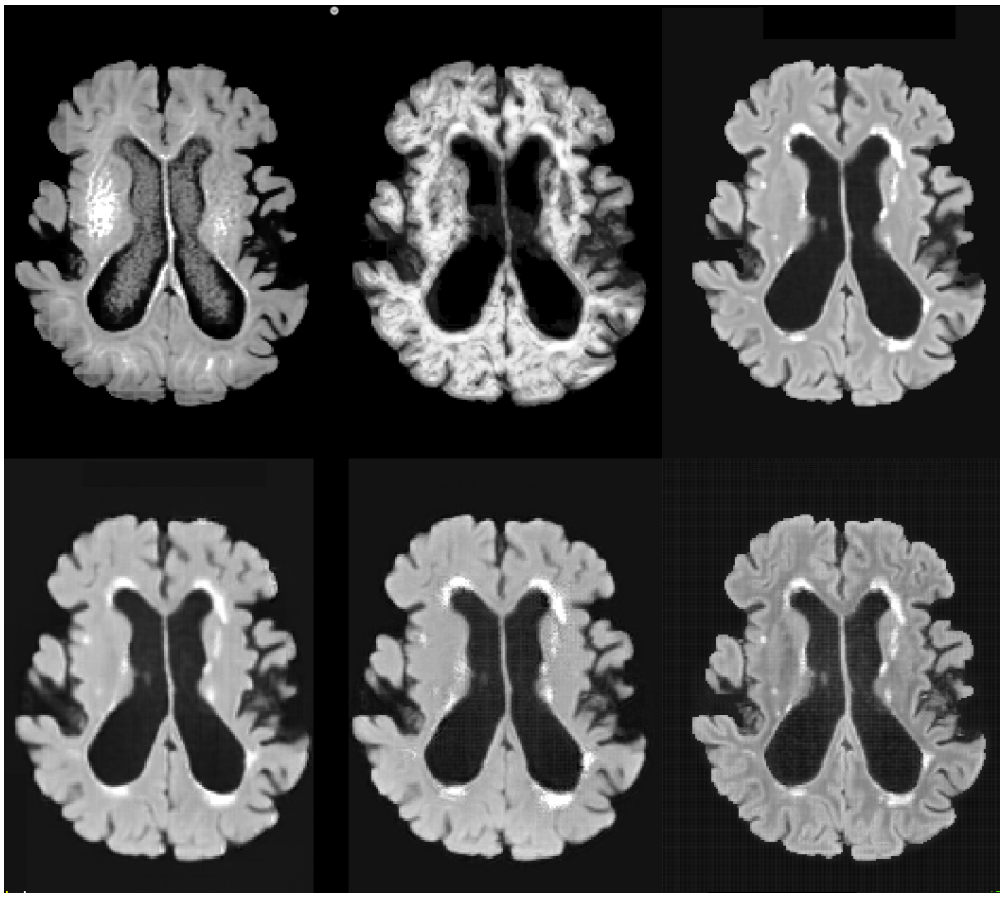


Figure 15: comparison of some of the GANs and CycleGANs on WMH dataset. **Top row**: ResGAN2 (Soft Labels), 3D CycleGAN, GAN1 (Soft Labels). **Bottom row**: ResGAN1 (Soft Labels), GAN2 (Soft Labels), ResGAN2 (Hard Labels). As we can see the Cyclegan cannot map T1-Weighted images into FLAIR, while the others mostly do it. It is interesting to notice how GAN1 outperforms the other architectures, even though ResGAN1 produces a fair result. GAN2 suffers from label smoothing and indeed the image has some artifacts, especially around the ventrivles, while ResGAN2 suffers a bit from the checkboard effect both with Soft Labels and Hard Labels.

Table 7: CycleGAN results.

Experiment	Avg_MSE	std_MSE	Avg_SSIM	std_SSIM	Avg_PSNR
3D CycleGAN	2.6020	4.1941	**0.4686**	**0.0553**	**20.2208**
2D CycleGAN	**1.1121**	**3.6555**	0.3386	0.1002	35.4372

when compared with GAN1, suffers from the effect of label smoothing and produces some artifacts close to the ventricles. Together with ResGAN2 both with hard (top row, first element) and soft labels (bottom row, third element), we notice the presence of the so called "checkboard effect", an artifact which is common when dealing with 3D CNN. The main reason for this effect to occur is because of the transpose convolution in the Decoder part of the network. By replacing it with a block made of upscaling + convolution (as for GAN1 and ResGAN1) this effect disappear. As mentioned above, the 3D CycleGAN (top row, second element) cannot learn a valid mapping function, therefore does almost no operation (a part from smoothing and adding noise) to the input image, therefore this result is similar to the T1-Weighted image.

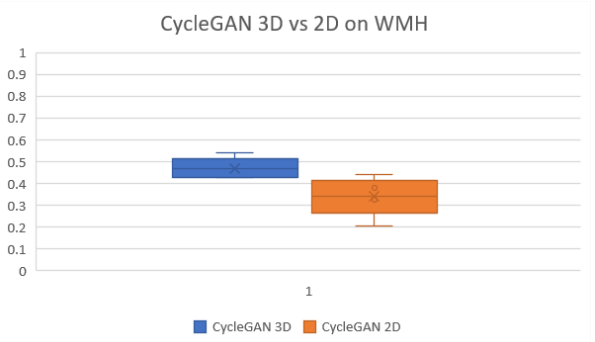


Figure 16: 3D vs 2D implementation of CycleGAN.

5.3. *Flair to T1-Weighted*

The second group of experimental tests that we performed, examines the previously presented architectures in synthesizing T1-Weighted images from FLAIR images. This is basically a specular analysis of most of the previously cited configurations of the other experiment. However, some of the configurations were avoided due to time consuming and due to the fact

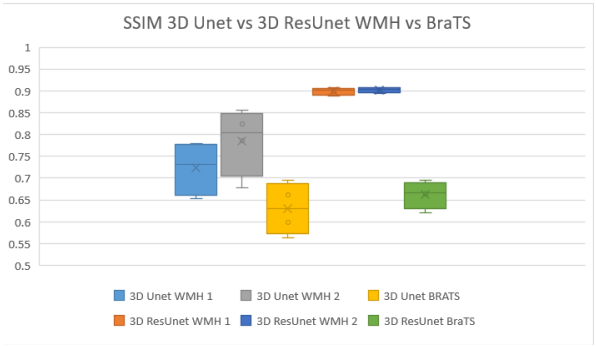


Figure 17: SSIM BoxPlot foldwise calculated on the presented configurations.

Table 8: Comparison between the three configurations of the two architectures: MSE, SSIM, PSNR

Case	Avg_MSE	std_MSE	Avg_SSIM	std_SSIM	Avg_PSNR
3D Unet 1 Input WMH	**0.0022**	**0.0001**	0.7741	**0.0547**	53.7906
3D Unet 2 Input WMH	0.0027	0.0005	**0.7859**	0.0673	**54.9755**
3D Unet 1 Input BraTS	0.0175	0.0065	0.6303	0.0772	50.8214
3D ResUnet 1 Input WMH	0.0010	0.0001	0.8988	0.0071	60.4702
3D ResUnet 2 Input WMH	**0.0009**	**0.0000**	**0.9020**	**0.0056**	**60.7154**
3D ResUnet 1 Input BraTS	0.0033	0.0022	0.6620	0.0193	50.2883

that the obtained results (especially in more complex architectures such as cycleGAN) were not good enough to justify another trial.

5.3.1. Unet3D and ResUnet 3D

Table 2 shows the setup of these configuration of experiments as well, and the results can be seen from Figure 17. These first results show that synthesizing T1-Weighetd images from FLAIR images is a more difficult task with respect to the opposite, even if to the network are fed two inputs as shown in Table 8, in which we actually notice that there is almost no difference in SSIM between the two configurations but instead the Average MSE is better when feeding only one input to the network. This general worsening in the results generating T1 Weighted images from FLAIR happens also when using the BraTS dataset, justifying so the fact that synthesizing T1 -Weighted images form FLAIR is a more difficult task with respect to the opposite.
When we change network an we switch to the 3D ResUnet, we can appreciate this worsening as well, even though the best Average SSIM is still 0.9020 ± 0.0056, so an average result. Furthermore, when analyzing Figure 17 we notice that not only the results of ResUnet are better in general, but also less dispersed from the median value. This means that in general, the network generalizes better than the 3D Unet when it comes to synthesize T1-Weighted images from FLAIR.

Table 9: Gan Configurations FLAIR to T1-Weighted.

Experiment	#Parameters	#Epochs	#Samples	Patience	Training Time
GAN 1 Soft L	13.86 M	20	1536	5	45 m
GAN 2 Hard L	10.25 M	20	1536	5	61 m

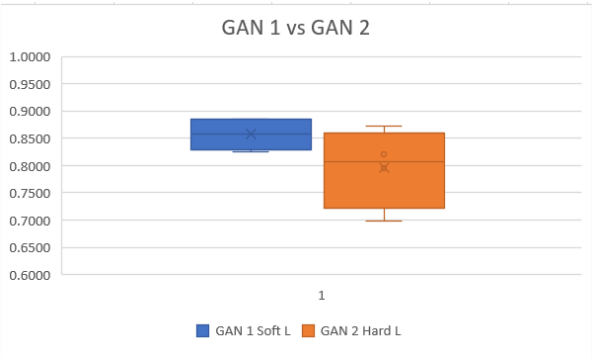


Figure 18: GAN 1 vs GAN 2 FLAIR to T1.

5.3.2. GAN

With respect to the previous experiment (T1-Weighted to FLAIR), this subsection of experiments only shows 2 configurations for synthesizing T1-Weighted images from FLAIR, as shown in Table 9. The fact that only these two configurations are presented is both due to the time consuming, and to the fact that since we generally see a worsening in the performances when synthesizing T1-Weighted images, it was worth to test only the best configurations on the easiest dataset (WMH).

5.3.3. CycleGAN

This last configuration ends the set of experiments done for this master thesis project. Results are presented in Table 11.

When comparing the 2D implementation to the 3D implementation, we notice that in general the 3D is better, yet the results are meaningless since the networks (as mentioned for the other experiment) do not learn any mapping. Another reason for this can be the fact that the discriminator outplays the translators quickly, but this happens no matter the architectures used.

6. Discussion

According to what has been presented in the Experiments and Results Section, we are able to produce some conclusions on the analysis.

Methods - Families. These sets of experiments done allow us to rank the different architectures according to the task itself (through the calculated metrics), but also according to the complexity of the network, training time. The general trend of the results, shows that the best approach to all the test was the **3D GAN** in a configuration with the 3D Unet as Translator and the usage

Table 10: Gan comparison of MSE, SSIM, PSNR.

Experiment	Avg_MSE	std_MSE	Avg_SSIM	std_SSIM	PSNR
GAN 1 Soft L	**0.0013**	**0.0003**	**0.8570**	**0.0101**	48.8080
GAN 2 Hard L	0.0036	0.0008	0.7966	0.0335	**54.2240**

Table 11: CycleGAN results.

Experiment	Avg_MSE	std_MSE	Avg_SSIM	std_SSIM	Avg_PSNR
3D CycleGAN	8.26452	6.798724	**0.402639**	**0.053957**	29.97478
2D CycleGAN	**3.644269**	**0.0811**	0.3063	0.118516	**39.7191**

of soft labels. Most probably, this setup is the best to solve this type of problems, according both to the results achieved, and without sacrificing too much time for training. **CycleGAN** (3D and 2D) along with 2D GAN (pix2pix) achieved the worst results overall, this might be due to the high complexity of the networks, combined with few amounts of datasamples (especially in 2D) which do not allow the translators to map the input distribution into the output distribution maintaining a correct cycle consistency within itself. **3D Unet** and **3D ResUnet** achieved as well a good result and **3D ResUnet** even outperformed **3D Unet** when synthesizing T1-Weighted images from FLAIR, and they are the configuration needing the least time to train, but the speedup in time do not compensate the worsen in the performance (although good) when compared to 3D GAN configuration.

T1-Weighted to FLAIR synthesis. As an overall result, we can notice that for each of the presented architectures and for both the datasets, trying to synthesize FLAIR from T1 is a relatively easy task. When comparing the numbers, we also notice that, when is possible to add two inputs to the network, as the lesions in T1 can be confused with the White matter itself, producing more realistic FLAIR (with also correct position of the lesions) is possible. Indeed, in the configurations (CycleGAN) and dataset (BraTS) which did not allow this additional input, results were less good.

FLAIR to T1-Weighted synthesis. On the other hand, trying to synthesize T1 from FLAIR it appeared to be a more difficult task than the previous one, for all the architectures and without any distinction of dataset or number of inputs. This might be due to the fact that generating T1-Weighted from FLAIR sequencies alone (without other modalities such as T2-Weighted or PD (Lee et al., 2019)) might actually not be possible.

Datasets. The presented study evidenced that for these architectures, the WMH dataset was easier to work with, even though it presented less cases. The reason why BraTS dataset achieved worse results, might be due to the fact that the images are coming from 19 scanners, therefore it is more difficult for the network to generalize well on so many different scanners, while for WMH dataset, the task is easier since the images are coming only from three scanners.

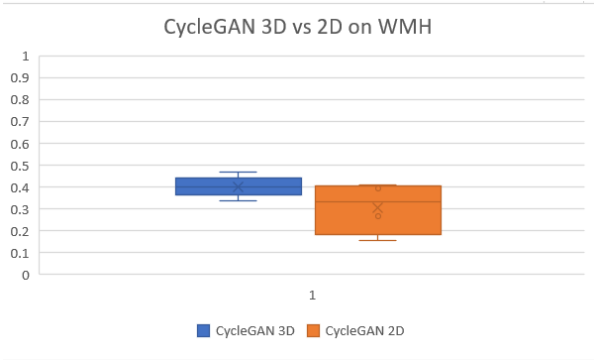


Figure 19: 3D vs 2D implementation of CycleGAN.

Table 12: Performance comparison of the analyzed architectures.

Network	Typology	Training Time	Result rank
Unet	3D	15 m	3
ResUnet	3D	20 m	2
GAN	3D	53 m (as average)	1
GAN	2D	13 m	4
CycleGAN	3D	75 m	5
CycleGAN	2D	25 m	6

1 Input v 2 Input. For the dataset (WMH) where it was possible to test two different configurations for the same problem, it is evidenced that providing the lesion mask as well as the modality in input to the network made the synthesis problem easier, especially for T1 to FLAIR. The fact that for this specific problem, the difference in SSIM when comparing 2 inputs vs 1 input is higher than the rest of the experiments, is due to the fact that the lesions are represented in a very different way in the two modalities. In FLAIR these are white, and easy to spot, while in T1 these can be confused with the White Matter itself, therefore when receiving 1 input, the network is not able to synthesize the lesions correctly.

7. Conclusions and future work

The main goal of this master thesis was to analyze state of the art techniques for synthesis - cross modality problems in Brain MRI images, more specifically, the synthesis from T1-Weighted images to FLAIR and viceversa was examined. To tackle this task, different architectures were developed, in order: 3D Unet, 3D ResUnet, 3D GAN (with variations on Translators and labels), 2D GAN, 3D CycleGAN and 2D CycleGAN. All these architectures were tested on two well known international brain dataset (WMH and BraTS), the first one with images from three sanners, the second one with images from 19 scanners. The analysis done shown that 3D GAN with the usage of label smoothing achieves really good results for WMH dataset when it comes to synthesize FLAIR from T1-Weighted images, along with 3D Unet and 3D ResUnet, this last performing in a more robust way when synthesizing T1-Weighted im-

ages from FLAIR. About BraTS dataset, all the architectures in general were less performing, and as mentioned in the discussion section, this might be due to the big amount of different scanners. In general, 3D Unet, 3D ResUnet and 3D GAN were really good performing in both the tasks. As a drawback, some of the results were not exactly as expected, showing that for this type of problem, the CycleGAN configuration was not able to generalize well and thus produced the worst results, both in 2D and in 3D configuration. A possible solution to this would be to apply some sort of data augmentation in order to have more samples to train with. The problem due to the cycle consistency might indeed be related to this lack of samples. As a further step to this work, the implementation and the testing of the Perceptual Similarity index as a loss function (Snell et al., 2017) which shown good results in other fields of research. A final step might be the implementation and the testing of a new state of the art technique Collagan (Lee et al., 2019) which will produce a more exaustive research and comparison.

Acknowledgments

First of all, I would like to thank my family for all the support I received during these two years away from them. Surely, without them I wouldn't be able to achieve anything in my life. Then, I would like to thank my professors and supervisors Dr Xavier Lladó and Dr Arnau Oliver for all the care they took with me during these months. I would like to thank also the PhD students in the VICOROB lab, Albert, Liliana, Kaisar, for all the hours spent listening to my doubts and for the practical help they provided me. Last, but not least, this is to me. Because at the end, if you really try to achieve something, it happens.

References

Dar, S.U., Yurt, M., Karacan, L., Erdem, A., Erdem, E., Çukur, T., 2019. Image synthesis in multi-contrast mri with conditional generative adversarial networks. IEEE transactions on medical imaging 38, 2375–2388.

Freeman, W.T., Pasztor, E.C., Carmichael, O.T., 2000. Learning low-level vision. International journal of computer vision 40, 25–47.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative adversarial nets, in: Advances in neural information processing systems, pp. 2672–2680.

Hiasa, Y., Otake, Y., Takao, M., Matsuoka, T., Takashima, K., Carass, A., Prince, J.L., Sugano, N., Sato, Y., 2018. Cross-modality image synthesis from unpaired data using cyclegan, in: International workshop on simulation and synthesis in medical imaging, Springer. pp. 31–41.

Jog, A., Roy, S., Carass, A., Prince, J.L., 2013. Magnetic resonance image synthesis through patch regression, in: 2013 IEEE 10th International Symposium on Biomedical Imaging, IEEE. pp. 350–353.

Lauritzen, A.D., Papademetris, X., Turovets, S., Onofrey, J.A., 2019. Evaluation of ct image synthesis methods: From atlas-based registration to deep learning. arXiv preprint arXiv:1906.04467 .

Lee, D., Kim, J., Moon, W.J., Ye, J.C., 2019. Collagan: Collaborative gan for missing image data imputation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2487–2496.

Liu, G., Si, J., Hu, Y., Li, S., 2018. Photographic image synthesis with improved u-net, in: 2018 Tenth International Conference on Advanced Computational Intelligence (ICACI), IEEE. pp. 402–407.

MICCAI, 2017. Miccai White Matter Hyperintensities 2017. URL: `https://wmh.isi.uu.nl`.

MICCAI, 2018. Miccai BraTS 2018. URL: `https://www.med.upenn.edu/sbia/brats2018/data.html`.

Miller, M.I., Christensen, G.E., Amit, Y., Grenander, U., 1993. Mathematical textbook of deformable neuroanatomies. Proceedings of the National Academy of Sciences 90, 11944–11948.

Nie, D., Trullo, R., Lian, J., Wang, L., Petitjean, C., Ruan, S., Wang, Q., Shen, D., 2018. Medical image synthesis with deep convolutional adversarial networks. IEEE Transactions on Biomedical Engineering 65, 2720–2730.

Perone, C.S., Ballester, P., Barros, R.C., Cohen-Adad, J., 2019. Unsupervised domain adaptation for medical imaging segmentation with self-ensembling. NeuroImage 194, 1–11.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation, in: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (Eds.), Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, Springer International Publishing, Cham. pp. 234–241.

Salem, M., Valverde, S., Cabezas, M., Pareto, D., Oliver, A., Salvi, J., Rovira, À., Lladó, X., 2019. Multiple sclerosis lesion synthesis in mri using an encoder-decoder u-net. IEEE Access 7, 25171–25184.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X., 2016. Improved techniques for training gans, in: Advances in neural information processing systems, pp. 2234–2242.

Shin, H.C., Tenenholtz, N.A., Rogers, J.K., Schwarz, C.G., Senjem, M.L., Gunter, J.L., Andriole, K.P., Michalski, M., 2018. Medical image synthesis for data augmentation and anonymization using generative adversarial networks, in: International workshop on simulation and synthesis in medical imaging, Springer. pp. 1–11.

Snell, J., Ridgeway, K., Liao, R., Roads, B.D., Mozer, M.C., Zemel, R.S., 2017. Learning to generate images with perceptual similarity metrics, in: 2017 IEEE International Conference on Image Processing (ICIP), IEEE. pp. 4277–4281.

Vemulapalli, R., Van Nguyen, H., Kevin Zhou, S., 2015. Unsupervised cross-modal synthesis of subject-specific scans, in: Proceedings of the IEEE International Conference on Computer Vision, pp. 630–638.

Wang, L., Chen, W., Yang, W., Bi, F., Yu, F.R., 2020. A state-of-the-art review on image synthesis with generative adversarial networks. IEEE Access 8, 63514–63537.

Wong, W., . What is label smoothing. URL: `https://towardsdatascience.com/what-is-label-smoothing-108debd7ef06`.

Xiang, L., Li, Y., Lin, W., Wang, Q., Shen, D., 2018. Unpaired deep cross-modality synthesis with fast training, in: Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support. Springer, pp. 155–164.

Yang, Q., Li, N., Zhao, Z., Fan, X., Chang, E.I., Xu, Y., et al., 2018. Mri cross-modality neuroimage-to-neuroimage translation. arXiv preprint arXiv:1801.06940 .

Yi, X., Walia, E., Babyn, P., 2019. Generative adversarial network in medical imaging: A review. Medical image analysis 58, 101552.

Yu, B., Zhou, L., Wang, L., Fripp, J., Bourgeat, P., 2018. 3d cgan based cross-modality mr image synthesis for brain tumor segmentation, in: 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018), IEEE. pp. 626–630.

Zhang, Z., Yang, L., Zheng, Y., 2018. Translating and segmenting multimodal medical volumes with cycle-and shape-consistency generative adversarial network, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 9242–9251.

Zhao, H., Gallo, O., Frosio, I., Kautz, J., 2015. Loss func-

tions for neural networks for image processing. arXiv preprint arXiv:1511.08861 .

Zhu, Taesung Park, P.I.A.A.E., 2017. Mri cross-modality neuroimage-to-neuroimage translation. Proceedings of the IEEE International Conference on Computer Vision (ICCV) , 2223–2232doi:arXiv:1801.06940[cs.CV].