UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE FÍSICA

A Patch-Wise Generative Adversarial Network for PET-MR Image Generation with Feature Attribution for Detection of Focal Cortical Dysplasia

Helena Santos Sousa

Mestrado Integrado em Engenharia Biomédica e Biofísica
Perfil em Engenharia Clínica e Instrumentação Médica

Dissertação orientada por:
Dr. Raquel Conceição
Dr. Emma Robinson

2021

Resumo

Atualmente, estima-se que mais de 50 milhões de pessoas em todo o mundo sofram de epilepsia, sendo que um terço dos casos correspondem a epilepsias refratárias, onde as convulsões epiléticas não conseguem ser tratadas com o uso de fármacos. Deste modo, o procedimento cirúrgico aparece como solução para o tratamento destes casos, onde o tecido responsável pela origem das convoluções é removido. As Displasias Corticais Focais (FCDs, do inglês Focal Cortical Dysplasias) são malformações do desenvolvimento cortical que constituem o tipo mais comum de lesões corticais responsáveis por epilepsia refratária em crianças. Estas lesões têm uma manifestação em neuroimagens altamente heterogénea, ocorrendo em diferentes regiões do cérebro e com diferentes níveis de visibilidade. Deste modo, nos casos de epilepsia refratária, um terço de todas as lesões não conseguem ser corretamente identificadas por especialistas de neuroimagem. Adicionalmente, é comum existir um desacordo entre peritos de imagem sobre o que poderá ser considerado evidência de FCDs nas diversas modalidades de imagem, e o que poderá ser identificado apenas como a intrínseca heterogeneidade do tecido cerebral (que apresenta típicas variações saudáveis do cortéx). Este desacordo é também agravado pelo uso de diferentes equipamentos e métodos de imagem médica, nos diversos estabelecimentos hospitalares e clínicos, propulsionando o aparecimento de variados tipos de ruído associados a processos de aquisição de imagem. Como consequência de uma equívoca identificação da localização de uma FCD, o planeamento pré-cirúrgico é frequentemente realizado de forma incorreta, resultando numa cirurgia malsucedida que se traduz num tratamento ineficaz para os pacientes. No entanto, recentemente, as Redes Adversárias Generativas (GANs, do inglês Generative Adversarial Networks) demonstraram o seu poder na deteção de anomalias em neuroimagens. Estes modelos utilizam técnicas de aprendizagem automática para a deteção de subtis padrões em imagens médicas, associados a doenças ou lesões, que poderão ser impercetíveis à natureza humana, e que visam ser robustos aos problemas relacionados com diferentes equipamentos e aquisição de imagens mencionados anteriormente. Assim, esta dissertação propõe o uso destes poderosos modelos computacionais para a deteção de lesões em imagens médicas, com a possibilidade de constituir a base para futuros projetos que visem a implementação de ferramentas auxiliares para a deteção automática de FCDs, proporcionando um apoio adicional a peritos de imagens aquando do planeamento cirúrgico, revelando as regiões de interesse onde poderão existir lesões. Deste modo, este trabalho aplica dois modelos de GANs - WGAN e CycleGAN - para deteção de anomalias (FCDs) em imagens de tomografia por emissão de positrões (PET do inglês positron emission tomography) e Ressonância magnética (MR do inglês Magnetic Ressonance), de pacientes epiléticos. Estas neuroimagens possuíam anotações clínicas (denominadas em inglês por labels) registadas por peritos de imagem, que indicavam as possíveis regiões cerebrais onde as lesões se localizavam, para cada paciente. Foi então possível criar máscaras binárias, para todas as imagens, das lesões encontradas pelos especialistas, que indicam as localizações das mesmas, e que foram usadas no treino dos modelos. Isto permitiu que os modelos de GANs usados neste projeto focassem a sua atenção nestas regiões de interesse, aprendendo a distinguir padrões associados a FCDs em neuroimagens de MR e PET.

Assim, duas técnicas distintas de deteção foram utilizadas: deteção por reconstrução de imagens (usando um modelo designado por WGAN) e deteção por translação de imagem (usando um modelo designado por CycleGAN). Ambas estas técnicas passam por treinar GANs com uma base de dados que possuí um número reduzido de exemplos 3D de neuroimagens disponíveis, o que motiva a adoção de um treino realizado em porções (*patches*, do inglês). Este treino em patches é definido por repartir aleatoriamente a imagem total em diversas porções 3D de tamanho mais reduzido desta imagem original, de modo a diminuir a memória computacional requerida para treinar estes modelos e, simultaneamente, funcionando como uma técnica que possibilita aumentar o número de exemplos de imagens disponível para o seu treino (uma imagem corresponderá a vários exemplos de treino consoante o número de diferentes patches extraídos). No caso da deteção por reconstrução, o objetivo passa por treinar o modelo com patches de imagens saudáveis, permitindo que este aprenda a distribuição característica do domínio de uma imagem saudável (sem FCDs). Deste modo, quando é apresentado ao modelo imagens com lesões, este deverá demonstrar um error de reconstrução de imagem elevado nas zonas onde se encontra uma lesão. No caso da técnica de translação de imagem, a CycleGAN foi treinada com patches de imagens saudáveis e com lesões, com o objetivo de aprender a translação entre uma imagem com lesão e uma imagem saudável. Assim, a deteção de lesões é possível através de um mapa de diferenças calculado entre a imagem original dos pacientes e a sua "versão saudável", que resultou da translação realizada pela CycleGAN. Este mapa de diferenças apresentará aglomerados nas regiões da imagem que corresponderão às anomalias identificadas. No caso da WGAN, o mapa de diferenças será calculado entre a imagem original e a sua reconstrução, onde os maiores erros ao reconstruir a imagem serão evidenciados neste mapa, detetando assim a localização das lesões. Nesta dissertação, ambos os modelos de GANs foram inicialmente treinados com ambas as modalidades de imagem (PET e MR disponíveis de cada paciente), com o intuito de analisar o seu impacto no desempenho nos modelos em detetar lesões bem como examinar eventuais problemas associados à utilização de multimodalidades, em relação a modalidades individuais de imagem, durante o treino de modelos de GANs. Assim, ambos os modelos foram treinados usando ambas as modalidades ou usando apenas cada modalidade de imagem individualmente. Para avaliar o desempenho de ambos os modelos na deteção de lesões, estes foram testados em dois pacientes com FCDs de visibilidade distinta, não presentes nos exemplos usados para o treino destes modelos. Deste modo, um dos pacientes de teste possuía uma FCD de grande visibilidade em ambas as modalidades de imagem, e o segundo paciente de teste possuía uma FCD muito subtil em ambas as imagens de PET e MR. Os resultados obtidos demonstraram que ambos os modelos (WGAN e CycleGAN) treinados com ambas as modalidades e com as modalidades individuais, foram capazes de detetar, nos mapas de diferenças, a FCD mais facilmente visível nas neuroimagens de um dos pacientes de teste. No entanto, para o caso das lesões bastante subtis, estes modelos mostraram maior dificuldade em as localizar nos mapas de diferença, não sendo capazes de as identificar de uma forma precisa em todos os modelos treinados. Através dos resultados obtidos, foi também possível observar a dificuldade que os modelos GANs têm em treinar com multimodalidades de imagem.

O treino destes modelos mostrou ser mais instável (difícil de atingir um equilíbrio entre as suas funções de custo), comparado ao treino de modelos GANs que usam as modalidades de imagem individualmente. Deste modo, estudos recentes serão discutidos brevemente na conclusão deste trabalho, que mencionam novas técnicas de fusão de modalidades de imagem, que visam encontrar novas estratégias para melhorar o treino de modelos multimodais e consequentemente o seu desempenho, bem como hipóteses de passos futuros a considerar para uma melhor deteção de FCDs mais subtis. Através deste projeto, foi então possível demonstrar o grande potencial que estes novos modelos formados por GANs possuem para constituírem a base de ferramentas auxiliares a peritos de imagem para deteção de lesões como as FCDs.

Palavras chave: Displasia Cortical Focal, Redes Adversariais Generativas, Detecção de Lesões, Ressonância Magnética, Tomografia por Emissão de Positrões

Abstract

More than 50 million people worldwide suffer from epilepsy with a third of those being diagnosed with drug-resistant epilepsy where the seizures cannot be treated through pharmacotherapy. In these cases, surgical removal of the epileptic brain tissue in patients is presented as an effective solution for treatment. However, for surgery success, it is vital that the accurate location of epileptic regions in the brain are known. Neuroimaging, specifically magnetic resonance imaging (MRI) and positron emission tomography (PET), commonly are the doctor's allies in identifying these lesions' locations responsible for the seizures. Focal cortical dysplasias (FCDs) are the most common type of cortical lesions responsible for drug-resistant epilepsy in children. These lesions have highly heterogeneous masses, occur in different brain regions and result in different levels of visibility, corresponding to the second most intractable type of lesion in adults with epilepsy. Moreover, among drug-resistant epilepsy cases, a third of these lesions cannot be correctly identified by neuroimaging experts, resulting in unsuccessful surgical planning and consequently ineffective treatment for patients. Recently, Generative Adversarial Networks (GANs) have demonstrated their value in neuroimaging anomaly detection. Therefore, this work proposes the application of two different GAN methods – WGAN and CycleGAN - for anomaly detection of FCDs, in PET-MRI data of epileptic patients. A 3D patch-basis anomaly detection approach was therefore developed, inspired by previous works, to detect FCDs location by deconfounding acquisition noise and normal cortical variabilities in PET-MR brain scans of epilepsy patients. Therefore, the GAN models applied two different approaches for lesion detection: detection through reconstruction (WGAN) and detection through translation (CycleGAN). Moreover, the combination of PET and MR modalities was studied and compared to training the networks with individual imaging modalities instead. Through the results, it was possible to understand and correct some issues GAN models have when training with multimodal 3D data. However, both methods for anomaly detection were able to detect diseased brain areas in patients with very visible FCDs, although failing to identify them in patients with very subtle lesions. Recent studies will be briefly discussed in the conclusion, which propose new approaches and architectures for multimodality training, with great potential to improve the performance of the networks for anomaly detection in future works.

Keywords: Focal Cortical Dysplasia, Generative Adversarial Networks, Lesion Detection, Magnetic Resonance Imaging, Positron Emission Tomography

Acknowledgments

Firstly, my deeply gratitude goes out for my supervisors Dr. Raquel and Dr. Emma for helping me relentlessly throughout this long journey and encouraging me to learn as much as I can, which I certainly did. I would also like to thank Professor Alexander Hammers, Dr. Siti Yaakub, Dr. Colm McGinnity, Dr. Jorge Cardoso, and all my colleagues in the METRICS Lab that received me with kindness and were always ready to help. A special thank you to Mariana Silva for letting me share this masters' project alongside her. A thank you to my friends for making this academic journey a little easier, and of course my family, who always supports me in whatever professional path I choose.

List of Abbreviations

BCE - Binary Cross Entropy.

CycleGAN - Cycle Generative Adversarial Network.

TE - Echo Time.

FCD - Focal Cortical Dysplasia.

FID - Fréchet Inception Distance.

GANs - Generative Adversarial Networks.

TI - Inversion Time.

MRI - Magnetic Resonance Imaging.

MAE - Mean Absolute Error.

PSNR - Peak-Signal-to-Noise Ratio.

PET - Positron Emission Tomography.

TR - Repetition Time.

TSE - Turbo Spin Echo.

WGAN - Wasserstein Generative Adversarial Network.

Index

List of Figures

List of Tables

Chapter 1

Introduction

This Masters dissertation project, entitled "A Patch-Wise Generative Adversarial Network for PET-MR Image Generation with Feature Attribution for Detection of Focal Cortical Dysplasia", was developed throughout 2020/2021 and represents the culmination of my path throughout the Integrated Masters in Biomedical Engineering and Biophysics.

This work aimed to apply machine learning techniques to a brain PET-MRI dataset of epileptic subjects to detect FCDs, the most common cause of treatment-resistant epilepsy in children [1] and second most intractable origin of seizures in epileptic adults [1] .

This project was proposed by Dr. Emma Robinson, lab lead of the METRICS Lab in King´s College London, UK, which focuses on machine learning methods for translational medical imaging with applications in the field of neurology, including neurodevelopment and cortical surface processing. Through a recent partnership with Professor Alexander Hammers and Dr. Jonathan O'Muircheartaigh, both with extensive expertise in epileptic neuroimaging, it was possible to have access to labelled epilepsy PET-MRI data, with cortical lesions identified as FCDs.

FCDs are malformations of cortical development, with highly heterogeneous manifestations in medical imaging, occurring in different regions of the brain and presenting different levels of visibility depending on factors, such as age. These FCDs' characteristics combined with the complex shape and structure of the brain plus its healthy tissue variation across individuals, makes these lesions extremely challenging to identify on neuroimages, even for clinical experts. In fact, for drug-resistant cases, a third of all the lesions responsible for seizures cannot be identified [2], most of them being FCDs. This results in an unsuccessful surgical planning for the removal of the epileptic tissue of the brain and a consequently ineffective treatment for the patients.

Having this in mind, a diagnostic tool to support reviewing of Magnetic Resonance Imaging (MRI) and Positron Emission Tomography (PET) of patients with treatment-resistant epilepsy would be very useful. This tool could help neuroimaging experts to differentiate healthy brain variations from diseased tissue (the epileptic seizure origin), providing to experts more information about possible abnormalities in scans, which can improve detection rates of brain pathologies.

Recently, machine learning methods, such as GANs, have demonstrated their value in anomaly detection in neuroimaging. Therefore, this work will build upon past achievements based on GAN methods for localising the origin of epileptic seizures [3–6], and the modelling of healthy brain variation [7–10] to create a generalisable tool to detect FCDs. These networks will be applied in PET-MR scans of epileptic patients to create anomaly detection methods, which could contribute to a solution for the cases where surgical planning is unsuccessful.

Moreover, a framework like this could also address the detection of cortical paediatric disorders

in general since it proposes to tackle key problems of age-related tissue contrast and between-subject cortical heterogeneity. These problematic factors are strong contributors to covering subtle pathologies, and therefore preventing precise comparisons of healthy versus pathological tissue, between the exact cortical regions across patients and healthy controls.

This work will also take advantage of both neuroimaging modalities available (PET and MR) to detect FCDs, since recent studies have shown that complementary review of FDG-PET overlaid on MRI can improve detection of FCD lesions [11, 12]. Therefore, in this project, a PET-MR brain dataset of twenty-two epileptic patients with a wide range of ages will be used to train and test the developed networks. A patch-basis training (described in section 3.4.2) will also be implemented in the networks to tackle the reduced number of data available and the computational cost of 3D data.

A 3D patch-basis anomaly detection approach was therefore developed, inspired by the work of *Yaakub et al.* [13], to detect FCDs location by deconfounding acquisition noise and normal cortical variabilities in PET-MR brain scans of epilepsy patients. To build these GAN approaches for anomaly detection in PET-MR data, it was necessary to first explore the basic structure of such networks. This included building simple machine learning networks that are the basis of GANs architectures, for image reconstruction and translation tasks. The following section 1.1 (Dissertation Outline) describes in more detail the content of this dissertation, including the several steps taken to build the networks for anomaly detection in PET-MR data.

1.1 Dissertation Outline

Chapter 2 describes vital background theory necessary to fully understand GANs and their use for anomaly detection tasks that are explored throughout this project. Additionally, a brief literature review about GANs is mentioned in this chapter, including state-of-the-art papers about anomaly detection that served as an inspiration for this project.

Chapter 3 starts by describing the models built for image reconstruction, using T2 MR images of neonates from the dHCP dataset. These models include Autoencoders, U-nets and a WGAN. This first stage is essential since these image reconstruction models constitute the basic structures of more advanced networks for anomaly detection. Therefore, this allowed for an initial practice of the basic concepts of machine learning regarding GANs, creating an initial framework to then build upon in the next stages. Chapter 3 then describes the adaptation and further development of the previous built U-net for image translation tasks (with 2D and 3D data), using T1 and T2 MR scans belonging to neonates from the dHCP database. Therefore, T1 and T2 MR images were used to train and test different image-to-image translation models that were then compared: a U-net and a CycleGAN. The WGAN and CycleGAN models built in this chapter, for either image reconstruction or image translation, served as a basis to further develop and modify these models for the anomaly detection task explored in Chapter 4.

Chapter 4 adapts the models previously built in Chapter 3 to two different 3D patch-wise anomaly detection GANs: a 3D WGAN and a 3D CycleGAN. These two different networks were applied to a PET-MR dataset and used different methods for anomaly detection – detection through reconstruction (WGAN) and detection through translation (CycleGAN). Following the work of [13] and [14] it was, therefore, possible to implement a patch-based approach of these networks to identify cortical lesions in epilepsy patients.

Chapter 5 presents the conclusions drawn from this study, explores the limitations of these experiments, and considers future work on this subject.

Chapter 2

Theoretical Concepts and Literature Review

Machine learning methods arise as true contenders for developing automated solutions that can bring to life new tools for detecting cortical malformations in the brain. Specifically, a type of machine learning algorithms - GANs - have gained a lot of interest for anomaly detection in recent years. Some of the most relevant research that has demonstrated the utility of GANs in this area will be presented in section 2.2 (literature review). However, to understand how these networks have shown potential, it is essential to comprehend their basic theoretical concepts presented as follows.

2.1 Generative Adversarial Networks (GANs)

GANs were firstly introduced in 2014 by *Goodfellow et al.* [15] and are a class of machine learning techniques that consist of two simultaneously trained models: one model (the Generator) trained to generate fake data, and the other model (the Discriminator) trained to discern the fake data from real examples.

This architecture therefore puts two or more neural networks (usually convolutional neural networks) against each other in adversarial training, where one of those networks takes the role of a generative model that captures the data distribution, and a discriminator model that estimates the probability that a sample came from the training data rather than the generator network. The discriminator therefore has the goal of classifying the input sample as real (if it comes from the original dataset – ground truth) or fake (if it was produced by the generator). Figure 2.1 shows a general representation of the elements, inputs and tasks performed in a GAN model.

In GANs, the input of the generator depends on the type of GAN model, however, the original GAN model described, uses a vector of pure random noise sampled from a prior distribution, commonly gaussian or a uniform distribution. The output of the generator is then compared to the real sample that was drawn from the real data distribution, and through the training process it becomes more similar to the ground-truth samples. Therefore, while traditional convolutional neural networks for object recognition learn patterns in images, GANs train their generator to create those patterns from scratch [16].

In other variations of GANs, such as conditional GANs [17], the input of the generator is also constituted of an additional factor (such as an image), conditioning the generator and discriminator based on this input. In simpler terms, this allows the model to direct the generator network to synthesize a specific desired fake example, therefore modifying the original GAN model for targeted data generation. These

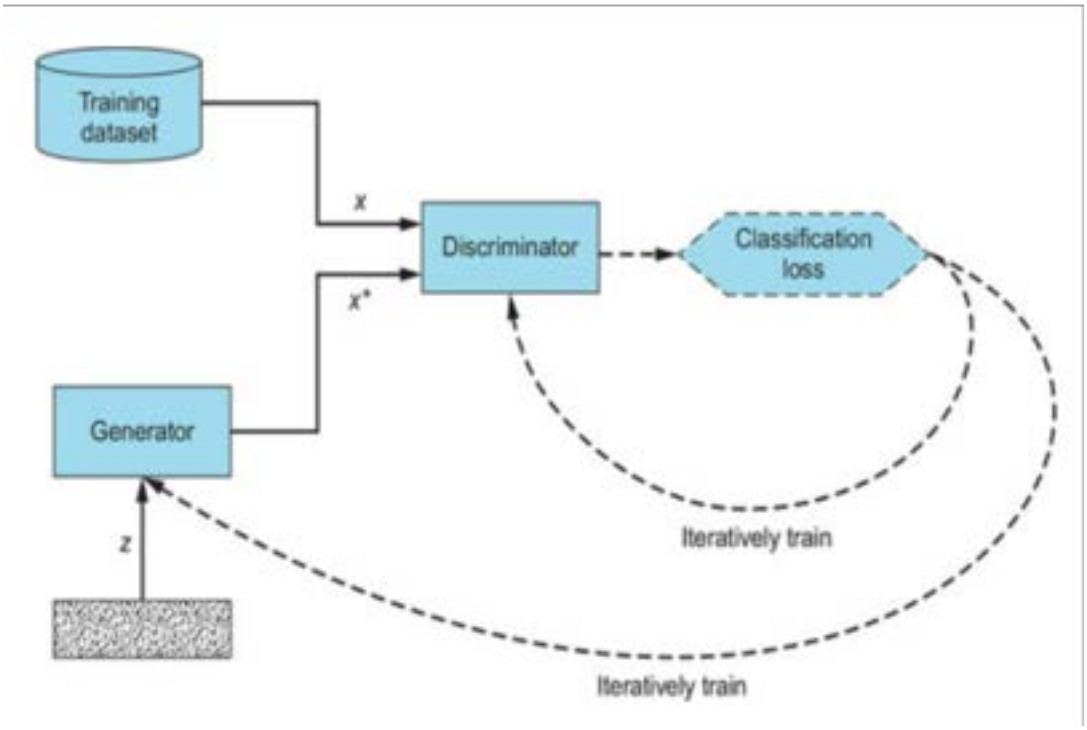

Figure 2.1: Overall structure of a GAN. From a training dataset, the original samples (x) serve as the input to the discriminator, as well as the generated/fake samples (x*), which come from the generator model. The Generator receives as input a random noise vector (z) that creates the fake samples. The discriminator model outputs a classification of the data as either real or fake, and the classification error is used to iteratively train the networks. Retrieved from [16].

conditional GANs are commonly used in, for example, medical image synthesis and image-to-image translation. Figure 2.2 shows a representation of a GAN model applied to the specific task of generating images of digits.

Therefore, since both the generator and discriminator models are implemented using neural-networks, each with its own loss function, the training of GANs uses a gradient-based optimisation algorithm. The parameters used to define the neural networks (defined as weights) are updated during training using backpropagation of the error (obtained from the loss function of the model), and according to the defined learning-rate of the networks. During training, the discriminator model aims to minimize the loss for the real and the fake samples it receives (aiming to correctly identify which samples are the ground-truth and which are generated), while the generator strives to maximize the Discriminator's loss for the generated/fake samples it produces [16].

As a result, GANs have two key factors that make them differ from traditional convolutional neural networks. Firstly, the loss function (J) of a conventional network is defined in respect to its own trainable weights (Θ), which is expressed as $J(\Theta)$. However, in GANs, the generator and discriminator have loss functions that depend on both the network's weights. This results in a generator's loss function represented as $\mathrm{J}^{G}(\Theta^{G}, \Theta^{D})$, and a Discriminator's loss function represented as $\mathrm{J}^{D}(\Theta^{G}, \Theta^{D})$. The second differentiating factor in GANs is that each network (the generator and the discriminator) can only tune its own parameters when training, instead of entire model parameters, as it happens in traditional neural networks. As a results, each network (generator and discriminator) only controls a part of what determines the entire GAN model loss [16].

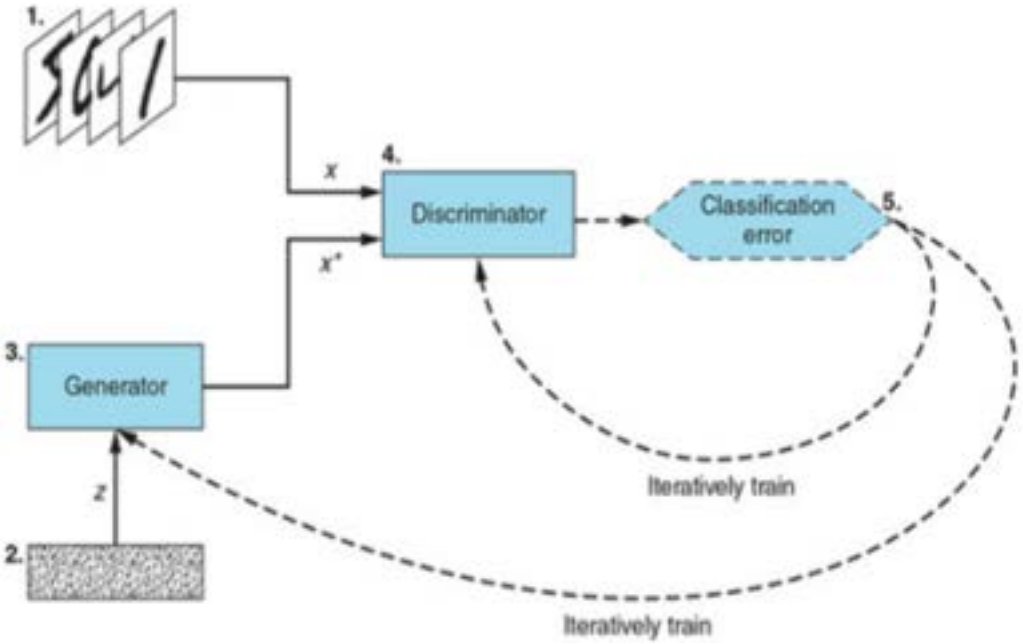

Figure 2.2: Overall structure of a GAN for image generation of digits. (1) represents the original samples (x) that will be inputted in the discriminator. (2) represents the input noise for the generator (z). (3) the generative model where its output will be "fake" images (x*). (4) the discriminator model where its output will be a classification of the digit image as either real or fake. (5) the classification error of the discriminator that will be used to iteratively train the networks. Retrieved from [16].

The architectures of the convolutional neural networks that constitute the generator and discriminator models in GANs varies according to the overall task of the GAN model. However, some of the most popular networks implemented as generators are based on Autoencoder networks or U-nets - which have also been applied in chapters 3 and 4 of this work.

2.1.1 Autoencoder

The Autoencoder is a neural network that consists of an encoder (typically built with convolutional layers to downsample data) and a decoder (typically built with transposed convolutional layers to upsample data). This network can learn how to map data to a compressed representation of itself (designated latent space) and reproduce it back to its original representation/dimensions. As a result, this network is capable of learning a mapping (through the encoder) from an input space (typically an image) into a latent space, as well as a mapping (through the decoder) from the latent space to the input space. Training an Autoencoder consists of passing an input data, such as an image, through the model, and measuring the error of its reconstruction (the decoded latent representation of the input image). The encoded and decoded learned mappings are therefore trained to get reconstructed images as close as possible to the original inputs [16, 18].

Figure 2.3 illustrates an Autoencoder network, with the representations of the encoder, decoder and latent space for image reconstruction of digits.

These Autoencoder networks, although seemingly simple, have many practical applications, such as one-class classifier for anomaly detection tasks, where it is possible to analyse a reduced representation of the data (latent space) of the trained network to check for similarities with a target class [16].

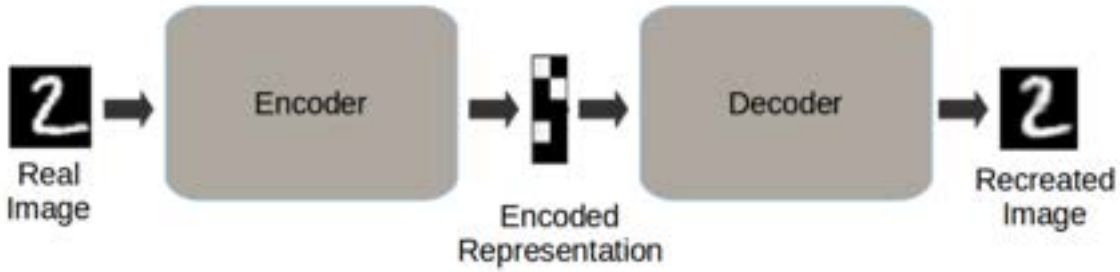

Figure 2.3: Example of an Autoencoder network. The encoder and decoder are usually composed of convolutional and transposed convolutional layers, respectively. The encoded representation illustrates the latent space. The input of the Autoencoder is an image that passes through the model to be reconstructed. Retrieved from [19].

2.1.2 U-net

From the Autoencoder network, a similar architecture was developed, the U-net [20]. This network architecture consists of the same encoder and decoder paths in the model but introduces skip-connections.

The skip-connections were implemented to improve reconstruction detail in the Autoencoder architecture. Therefore, they have the ability to recapture the finer details of the original images (the spatial information lost during encoding/downsample) in their reconstructions [20].

Nowadays, the U-net is a very popular network for biomedical image segmentation tasks and is usually implemented as a generator in GANs [20].

Figure 2.4 shows the U-net architecture, a similar network to the Autoencoder but with added skip-connections.

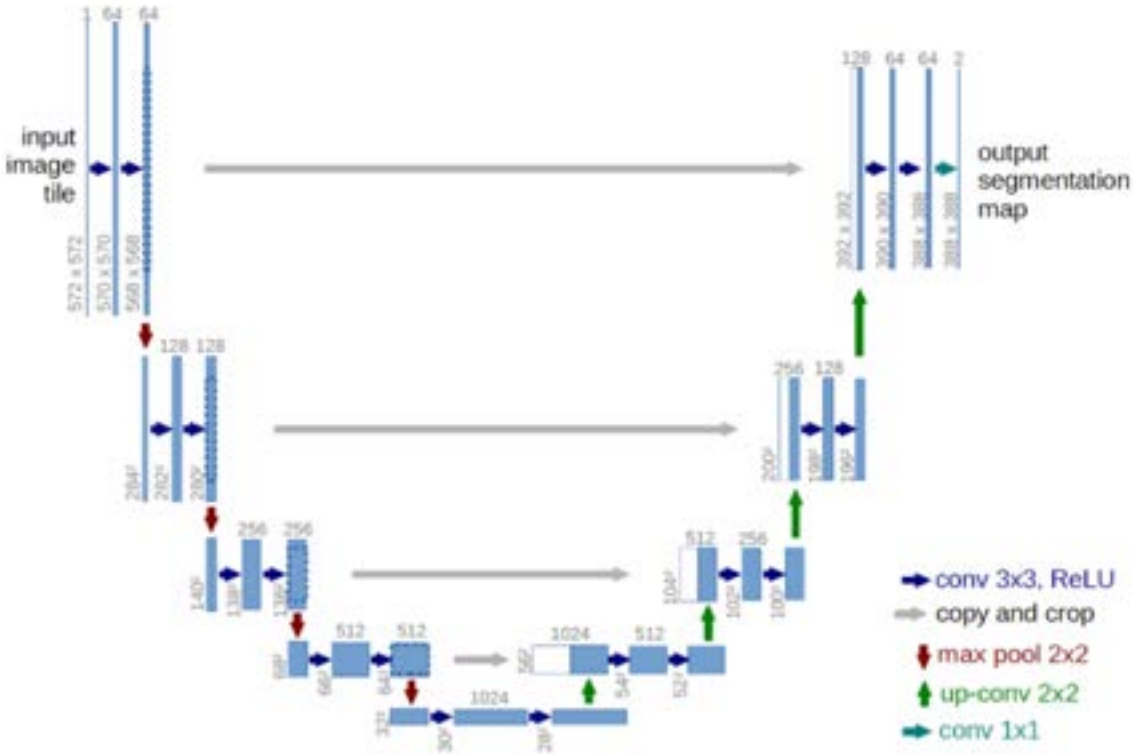

Figure 2.4: Example of a U-net architecture. The middle grey arrows between layers (with copy and crop description) correspond to the skip connections U-nets implement. Retrieved from [20].

The general structure, training process and network architectures described here are therefore the

basis of the functioning of GANs. However, since their introduction by *Goodfellow et al.* [15], many different variations of the original GAN structure have appeared for various tasks. In particular, two of those variations: WGAN and CycleGAN will be described briefly in sections 2.1.1 and 2.1.2, respectively, since they are the methods used in the following chapters 3 and 4.

2.1.3 WGAN

The WGAN was introduced in 2017 by *Arjovsky et al.* [21] and represents an extension of the GAN architecture, which distinguishes itself by using a Wasserstein distance (also denominated as earth mover's distance) as a loss function [21]. The Wasserstein distance has proven to be a more optimal measure compared to the original GAN model, therefore improving training stability of these networks [22] and generating higher-quality samples [16].

The discriminator network in the WGAN is called critic instead, which tries to estimate the earth mover's distance, and aims to reach for the maximum difference between the original and the generated distribution in the loss function. This critic network scores the "realness" or "fakeness" of a given input (usually an image), instead of classifying the input as real/original or fake/generated - the strategy described in the original GAN model [16]. In contrast, the generator in the WGAN tries to minimize the distance between the distribution of the real data, observed in training, and the distribution in the generated samples.

In practice, the implementation of a WGAN maintains the basic foundations of the conventional GANs described before, only with minor changes to its training (such as the use of a Wasserstein distance as a loss function) and architecture (the discriminator network is designated as critic, which outputs a score for the real and generated samples and does not use a sigmoid function, unlike typical GANs). Further details about the differences of this network compared to the original GAN model are described in the work of [21].

2.1.4 CycleGAN

Introduced in 2017 [23], a CycleGAN is an extension of the GAN architecture that involves the simultaneous training of two generator models and two discriminator models.

This model is commonly used for image-to-image translation tasks, one of the most revolutionary applications of GANs [23, 24]. This is based on the challenge of translating a representation of one image into another, such as trying to translate one medical image modality into another image modality (for example PET to MRI and vice versa). As a result, for this image translation between two different domains, the model learns to extract characteristic features of both these domains, discovering the underlying relationship between them.

The CycleGAN framework combines two sets of GANs (each with a generator and a discriminator) to learn a mapping from domain X to domain Y (generator G^X) and vice versa (generator G^Y), with the generators GX and GY being trained by discriminators DX and DY, respectively. Therefore, in the CycleGAN, one generator takes images from the first domain (X) as input and outputs images for the second domain (Y). The other generator takes images from the second domain (Y) as input and generates images for the first domain (X). The discriminator networks have then the same goal of determining the plausibility of the generated samples from the generator networks and update them accordingly [25].

Figure 2.5 shows the CycleGAN framework for PET-to-CT image translation task with the associated losses.

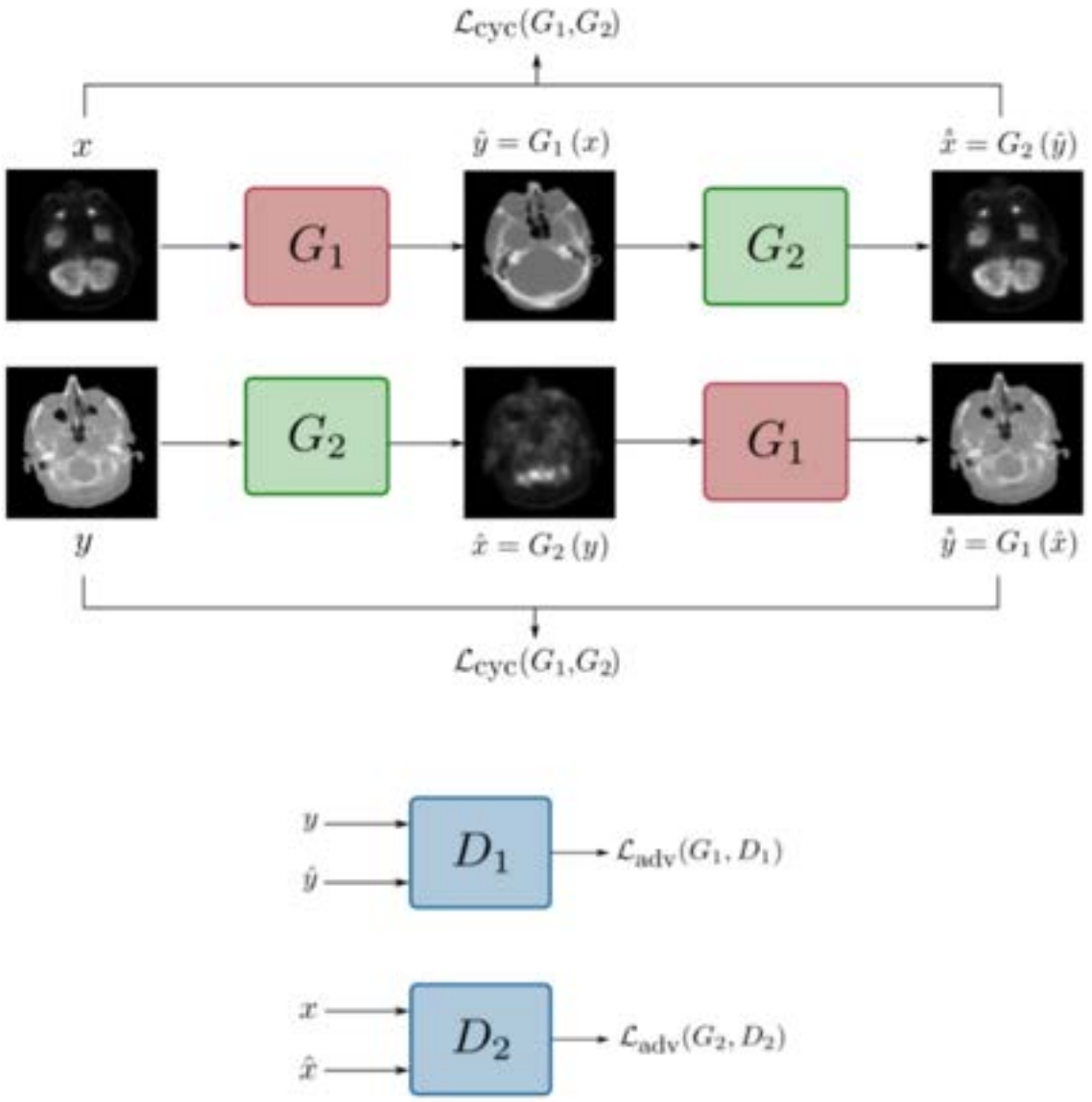

Figure 2.5: General scheme of the building blocks and losses of CycleGAN. G1 and G2 represent the Generators and D1 and D2 the discriminators. The x and y represent the original images of different domains that we want to translate while $\hat{x}$ and $\hat{y}$ represent the translated images from the generators of both the original x and y, respectively. The translation of the images $\hat{x}$ and $\hat{y}$ back to their original domain are represented by $\hat{\hat{x}}$ and $\hat{\hat{y}}$, respectively. The cycle consistency loss and adversarial loss are also represented in the figure by L_{cyc} and L_{adv} . Retrieved from [26].

Besides being possible to use unpaired data to train CycleGANs, another differentiating aspect of this model is the cycle consistency characteristic of CycleGANs. This represents the idea that the image output from the first generator could be used as input for the second generator in the model and so both the output of the second generator and the original image should match each other. Therefore, CycleGANs encourage this cycle consistency aspect by adding another loss (cycle consistency loss) that is responsible for measuring the difference between the output of the second generator and the original image (and the reverse), therefore acting as a regularization of the generator models, and so driving the generation process in the new domain towards image translation [23].

In CycleGANs, the U-net (represented in Figure 2.4) is commonly used as the generator network. As for the network used for the discriminators, CycleGANs usually use a PatchGAN [24], which classifies each N x N patch of the image and averages all the scores of the patches to get the final score for the

image, instead of classifying the image as a whole, like standard GANs.

The PatchGAN discriminator is a convolutional network that generates an output of a 70x70 array, instead of producing a single scalar vector as typical discriminators do. This 70x70 array maps to a patch of the original input image. The mean of this output is then calculated to predict if the whole image is "real" (the ground-truth image) or "fake" (a network generated image). The authors of [24] defend the use of this discriminator architecture with a 70x70 receptive field since it has less parameters, therefore being easier to train than a full-image discriminator. The patch size of 70x70 was found to be effective in multiple image-to-image translation tasks and is therefore the standard size used.

Once these underlying concepts of GANs are comprehended, it is possible to further understand why GAN-based approaches are implemented in tools for feature attribution and lesion/pathology detection, the main themes addressed in this dissertation. The following literature section will therefore discuss the use of GANs in medical imaging for anomaly detection.

2.2 Literature Review: Machine Learning for Anomaly Detection

Anomaly detection is the task of identifying outliers from the normal examples in a dataset, detecting the patterns that deviate from the general pattern present in the dataset. This topic has been explored extensively over the years for different areas, with several methods for anomaly detection being proposed depending on the type of dataset and abnormality [27, 28]. More recently, machine learning techniques have been extensively implemented for anomaly detection approaches.

The work of *O'Muircheartaigh et al.* [29] is an example of machine learning applied for anomaly detection, where a Bayesian regression technique was implemented to detect focal white matter injuries in MRI of neonates. The model was firstly trained to estimate brain tissue intensity of MR scans, and then calculate voxelwise deviations between the neonate's observed MRI and the intensities predicted by the model, to identify injuries. With this technique, from 408 neonate images, it was possible to correctly identify anomaly areas in 83% of the T2-weighted MR scans and in 76% of the T1-weighted scans.

Tan et al. [11] also showed the potential of machine learning in lesion detection by using multimodal feature sampling and applying a support vector machine classifier, improving the detection of FCDs in MR and PET data. The morphology and intensity-based features that characterised the FCD lesions in the images were calculated on the cortical surfaces and fed into the classifier. This classifier was able to outperform quantitative MRI analysis as well as multimodal visual analysis in detecting FCDs, by using combined features from both MR and PET modalities.

Similarly, the work of [30], developed a neural network classifier using surface-based features (such as grey-white matter intensity contrast, cortical thickness, FLAIR signal intensity, etc.) to identify FCDs in a paediatric population. This approach consisted in optimising the ability of finding and quantifying the cortex area depending on how much they differ from a naturally healthy cortex. Overall, this method then used the established surface-based features in the trained neural network model, to classify the cortical regions as either containing anomalies or not. The results from the classifier showed a correct identification of FCDs with a sensitivity of 73%.

However, as deep learning methods grew in popularity, traditional machine learning models were pushed aside, and data-driven approaches prevailed in anomaly detection of diseases (such as multiple sclerosis, Alzheimer's, epilepsy, tumours, etc.) in neuroimaging [14, 31]. Specifically, various deep learning techniques have been proposed for anomaly detection using artificial neural networks, with state-of-the-art methods commonly focused on GANs, Autoencoders and their variations [14]. Recent studies using GANs [13, 14, 32] have particularly demonstrated exciting potential in anomaly detection

tasks in medical imaging.

2.2.1 Anomaly Detection using GANs

The AnoGAN introduced in [33] was one of the pioneering works that successfully implemented a 2D patched-based convolutional GAN to achieve anomaly detection in optical coherence tomography images of the retina. This approach detected abnormalities by learning a model of healthy tissue and seeking anomalies as outliers from this distribution [32].

Further works followed, transferring this concept to the field of neuroimaging for anomaly detection in brain MR images [34]. Other examples include *Chen et al.* [35] that took advantage of an adversarial auto-encoder to learn the data distribution of healthy brain MR images to then highlight potential lesions. Works by *Sun et al.* [14] and *Yaakub et al.* [13] have also demonstrated that several types of GANs and approaches can be used for anomaly detection.

More specifically, two different methods for anomaly detection using deep learning have relevance for this work: anomaly detection through image reconstruction and through image translation.

In terms of anomaly detection using reconstruction, the work of [36] demonstrated the efficacy of this method using the reconstruction error of an autoencoder to detect anomalies in the MNIST dataset (a handwritten digit database). The goal was to minimize the reconstruction error of normal examples at the same time as maximizing the same error for any anomalies in the data. As a result, the model was trained to not be able to reconstruct the specific set of anomalies given during testing – in this case the digits 3 and 5 – as is illustrated in Figure 2.6.

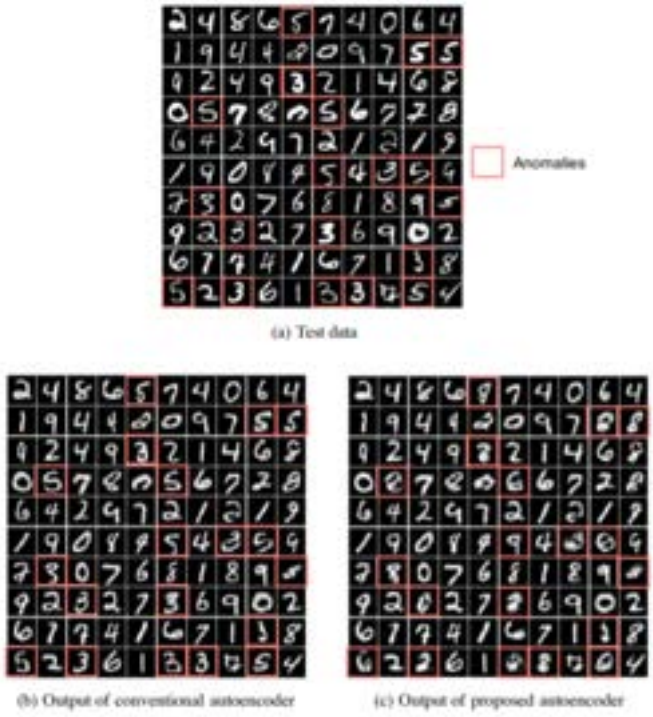

Figure 2.6: (a) The test dataset, including the anomaly digits 3 and 5. (b) Reconstruction outputs of a conventional autoencoder - can reconstruct any input. (c) Reconstruction outputs of the proposed autoencoder which fails to reconstruct anomalies. Retrieved from [36].

In the medical context, for detecting anomalies through reconstruction, models are applied to learn the healthy distribution of the data available. Therefore, the input of the networks should consist of only healthy samples, with its output being a reconstruction of that same healthy data. Once that healthy distribution is learnt, the models can reconstruct anomaly-free images but not diseased ones. Therefore, when presented with anomaly samples, the network will not be able to reconstruct the lesion regions of the data sample. The detection of the anomaly region can then be identified by evaluating the discrepancy between the input image and its reconstruction – which will reveal a higher reconstruction error on the anomaly location. The region that does not follow the learnt healthy distribution can be regarded as abnormal [14].

On the other hand, detecting anomalies through translation consists of giving the models both healthy and diseased labelled data samples. With this strategy, networks are trained to learn how to translate between healthy and diseased samples. This results in the network being able to detect anomalies by computing the difference between a diseased sample and its translation to healthy – where the network will translate only the diseased region to a healthy version of it.

The work of *Yaakub et al.* [13] had previously demonstrated that it is possible to use translation methods to detect FCD location within a brain lobe region in PET-MR datasets. In this work, a deep generative modelling of PET from MR was used to synthesise a model of healthy brain tissue and detect lesions as outliers. Therefore, a GAN was implemented to synthesise pseudo-normal PET scans from T1 MRIs for the identification of possible regions of hypometabolism. Firstly, a 3D patched-based GAN is trained to learn the mapping between T1 MRIs and the PET scans in control data (healthy individuals). After this, the previous network is used to generate the pseudo-normal FDG PET scans in patients suffering from epilepsy, based on the patients' T1 MRI. To track the hypometabolism areas in the brain, the patient's real PET scan was subtracted from the generated pseudo-normal PET scan. *Yaakub et al.* [13] represented these stages in Figure 2.7. The results from this work showed that the proposed GAN method was able to detect hypometabolic regions with high sensitivity of about 93% and 75% in MRI-positive patients and MRI negative patients, respectively, at a lobar level. Figure 2.8 shows examples of hypometabolic clusters detected in both MRI positive and MRI negative patients. This method explores the discrepancy between both FDG PET scans: the apparent normal PET scan (which was generated by the GAN model from the patient's normal T1 MRI scan) and the actual abnormal PET scan of the patient.

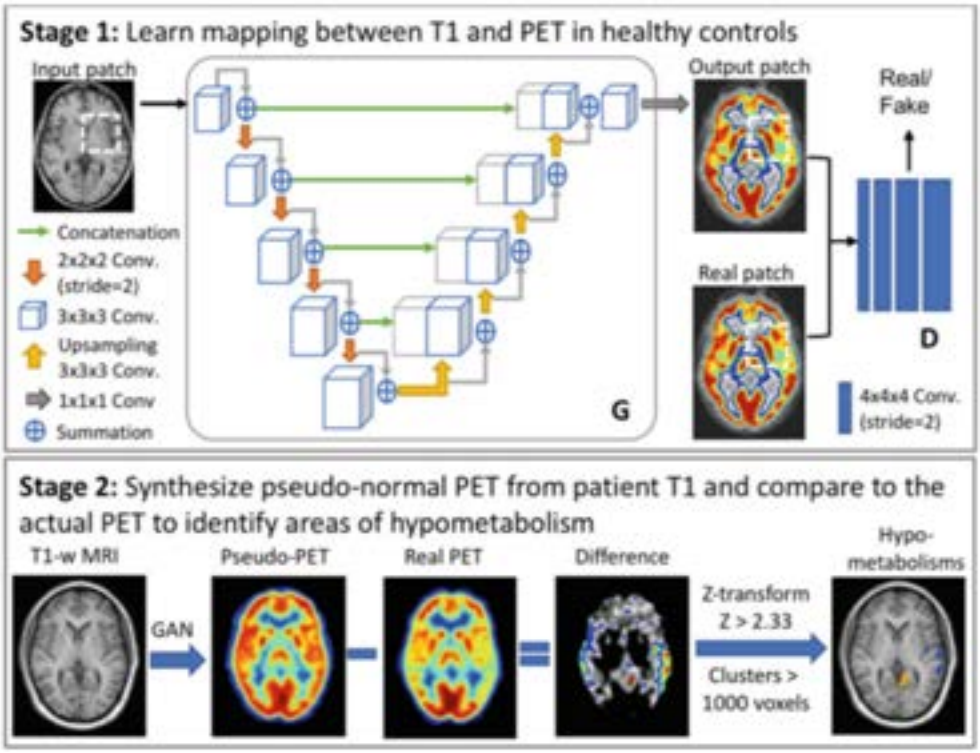

Figure 2.7: The different stages of the work of [13] for identifying hypometabolism in patients with epilepsy. Stage 1 represents a 3D-patch GAN architecture for estimating pseudo-normal PET from MRI. G stands for Generator and D for Discriminator. Stage 2 represents the identification of hypometabolic clusters in patients [13].

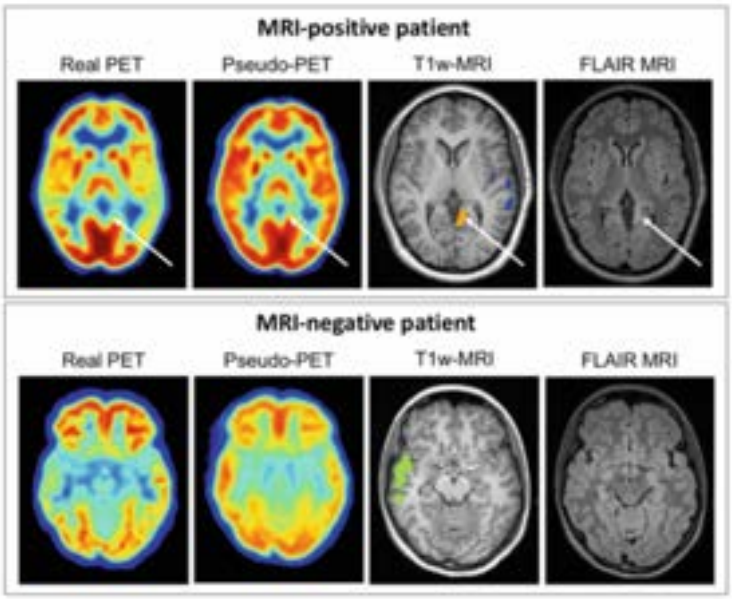

Figure 2.8: Examples of images of MRI-positive (top) and MRI-negative (bottom) scans of patients with detected hypometabolic clusters. The real [18F] FDG PET and pseudo-PET scans as well as the T1 MR scan with clusters of hypometabolism overlaid and FLAIR MR images highlighting the hypometabolism corresponding to the FCD for the MRI-positive case (white arrows) [13].

Similarly, the work of [14] demonstrated the use of a CycleGAN to perform translation of diseased MR scans to a healthy version of it. This network additionally applies an anomaly-mask loss (illustrated in Figure 2.9 as L_AM), which focuses the network's attention on the diseased region of a scan since there are masks of the lesions available with the dataset.

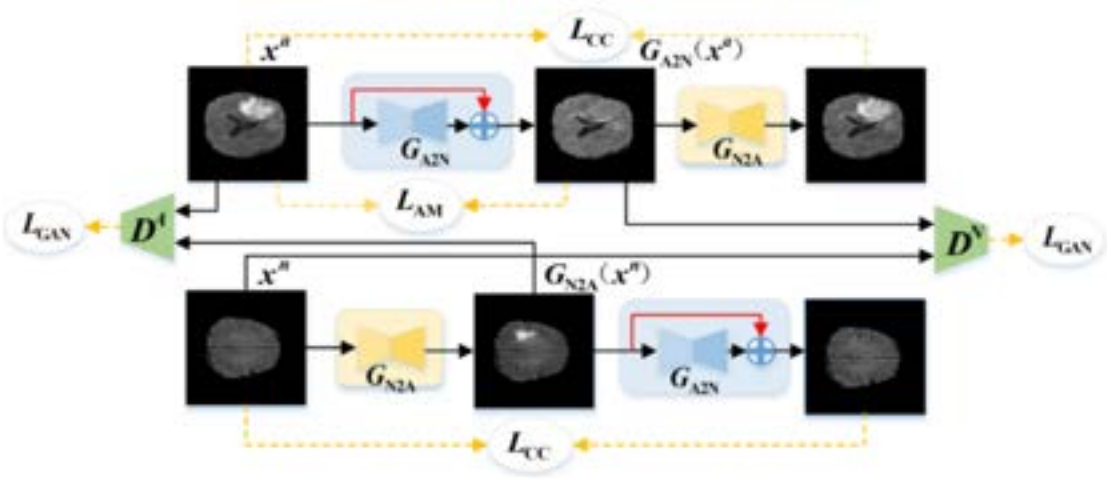

Figure 2.9: The proposed ANT-GAN model for lesion detection from [14]. The abnormal and normal MRI slices correspond to xa and xn respectively. G_{A2N} and G_{N2A} illustrate the generators that aim to translate abnormal to normal data and vice versa, respectively. D^A and D^A are in turn the discriminators that classify in "real" (original scans) or "fake" (network generated scans) images. From the discriminators, 2 losses result (represented by L_{GAN}). L_{CC} represents the cycle consistency loss – between the original image and its reconstruction - and L_{AM} represent the lesion mask loss – if the generator modifies a known healthy region of the image during the translation to diseased, it receives a heavy L2 penalty [14].

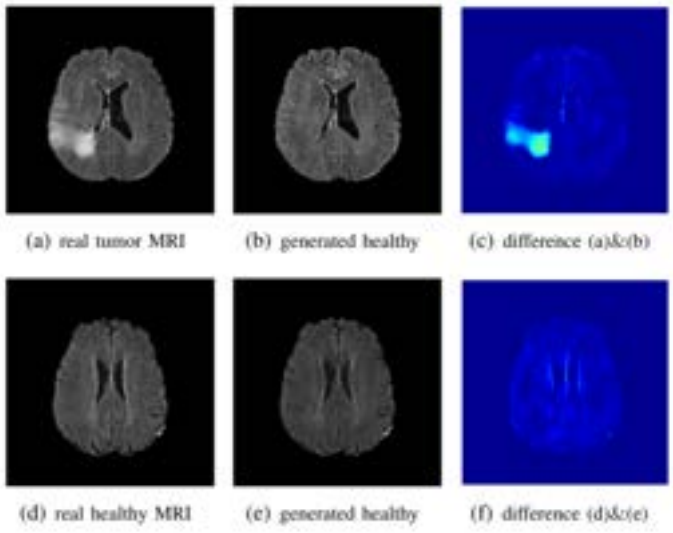

Figure 2.10: Results produced by the ANT-GAN model from [14]. (a) and (d) represent the original images with and without lesions, respectively. (b) and (e) illustrate the output of the generators of the model and (c) and (f) show the difference maps between the images, highlighting the lesions or lack of them [14] .

It is then possible to identify regions where lesions are present by getting the difference map (Figure 2.10(c)) between the original scan (the diseased image - Figure 2.10(a)) and the translated one (that should correspond to a healthy version of the scan where the lesion is not present - Figure 2.10(b)).

Through this literature review it is hopefully possible to understand the impact of GANs in applications such as anomaly detection, specifically the potential in detecting lesions or regions of interest in epilepsy, therefore emerging as a tool to help imaging experts in focusing on probable lesions, essential for a correct surgery planning.

Therefore, the following chapters of this dissertation focus on the implementation of GANs for anomaly detection in epileptic patients - starting with chapter 3, by exploring the basis of what constitutes more complex GANs, developed for anomaly detection in chapter 4, and taking inspiration from the works explored in this literature review section.

Chapter 3

MR Image Reconstruction and Translation

3.1 Motivation

The present chapter is dedicated to building, applying, and benchmarking different networks for image reconstruction and translation tasks, which will serve as the basic structure of more advanced networks, such as GANs, implemented in Chapter 4 for PET-MR anomaly detection. All experiments were built using the Pytorch and MONAI machine learning frameworks [37, 38], and trained using a NVIDIA Titan RTX GPU with 24 GB of RAM.

Firstly, 2D neural network-based models - an Autoencoder, U-net and WGAN - were applied for image reconstruction of neonatal MRI brain data (T1 and T2-weighted scans), belonging to the developing human connectome project (dHCP) [39]. Furthermore, some of these networks were modified for 2D and 3D T2-to-T1 image translation tasks, as well as a CycleGAN - a network specifically designed for image-to-image translation. These networks were chosen since relevant papers [13, 14] described in section 2.2 commonly use them as a basis for other complex GANs for anomaly detection tasks - explored in Chapter 4.

Additionally, two different training approaches were tested for image translation: whole-image-based training and patch-based training. This comparison is motivated by the aim to use patch-based approaches in Chapter 4, since it has proven to better learn global image context and is commonly used for data augmentation and to save computational memory resources. To evaluate the quality of the translated images, three different metrics were used, following the work of *Yaakub et al.* [13] : FID (Fréchet Inception Distance), MAE (Mean Absolute Error) and PSNR (Peak-Signal-to-Noise Ratio).

Therefore, in this chapter, section 3.2 describes the dataset used for image reconstruction and image translation tasks. In section 3.3, the different architectures of the networks are illustrated and hyperparameters used in training described. Finally, in sections 3.4 and 3.5, results are presented, evaluated, and discussed, respectively.

3.2 Dataset Structure

3.2.1 Data Acquisition and Pre-Processing

The dataset used for reconstruction and translation tasks consisted of T1 and T2-weighted MR scans of neonates, belonging to the dHCP dataset [39]. This data was collected at St. Thomas Hospital, London, on a Philips 3T scanner using a 32-channel dedicated neonatal head coil [40]. For image acquisition, subjects were not sedated but imaged during natural sleep. T2 images were obtained using a Turbo Spin

Echo (TSE) sequence, in two stacks of 2D slices (in sagittal and axial planes), with the following parameters: repetition time TR=12s, echo time TE=156ms, SENSE factor 2.11 (axial) and 2.58 (sagittal) with overlapping slices (resolution $0.8 \times 0.8 \times 1.6$ mm) [41]. T1 images were acquired using an IR (Inversion Recovery) TSE sequence with the same resolution and with parameters: TR=4.8s, TE=8.7ms, SENSE factor 2.26 (axial) and 2.66 (sagittal) [41].

Motion correction and super-resolution reconstruction techniques were employed combining *Cordero-Grande et al.* [42] and *Kuklisova-Murgasova et al.* [43], resulting in isotropic volumes of resolution $0.5 \times 0.5 \times 0.5$ mm^3 (with T1 and T2 scans having dimensions of 196 x 230 x 196 voxels).

All scans had also been previously pre-processed (including data normalisation between 0 to 1), before being used for image reconstruction and translation tasks described in this chapter.

3.2.2 Image Reconstruction Dataset

The data used for image reconstruction involved 850 (454 male and 396 female) 3D T2-weighted MR images of neonates scanned between 28 and 45 weeks of age (with a mean scan age of 40 weeks). From the 850 images, 10 axial slices from the centre of the brain were chosen to obtain 2D data. Figure 3.1 illustrates three examples of axial slices obtained from the dataset.

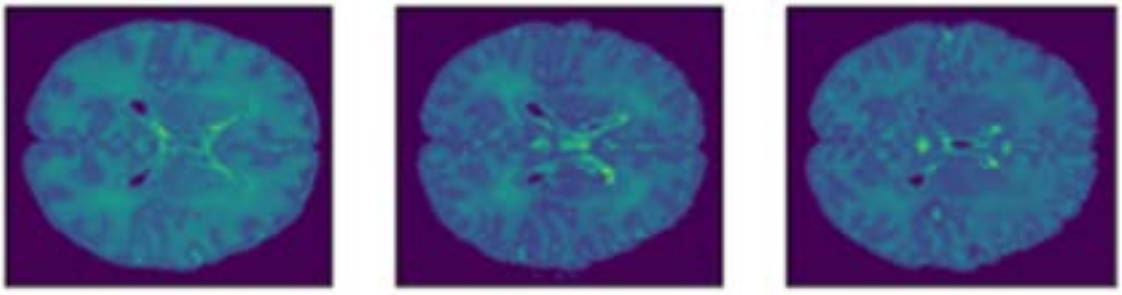

Figure 3.1: Example of 3 different T2 MR slices of subjects belonging to the dHCP dataset used in the image-reconstruction task. All slices plotted using the same colour map.

In total, 8500 T2-weighted MR 2D image slices (with dimensions of 196 x 230 pixels) were used, with 8000 samples implemented for training and 500 samples used for testing the networks.

3.2.3 Image Translation Dataset

For image translation, a total of 279 subjects (163 males and 116 females with mean age scan of 40 weeks), each with a corresponding pair of T1 and T2 3D MR scans, were used - illustrated in Figure 3.2.

For the 2D networks, 10 middle axial slices from all the images were selected for training and testing. Therefore, a total of 2790 2D image pairs of T1 and T2 scans (with dimensions 256 x 256 pixels) were used. For training, 80% of the pairs of images (2232 pairs) were selected randomly and the remaining 20% were implemented for testing (558 pairs).

For 3D networks, the total 279 pairs of 3D images (T1 and corresponding T2 scan, for each subject) were used, with total dimensions of 256 x 256 x 256 voxels. Since 3D data presents more challenges to machine learning networks, a validation set was additionally included to aid in the evaluation of the performance of the networks through training. Therefore, 5% of the data was used for validation and

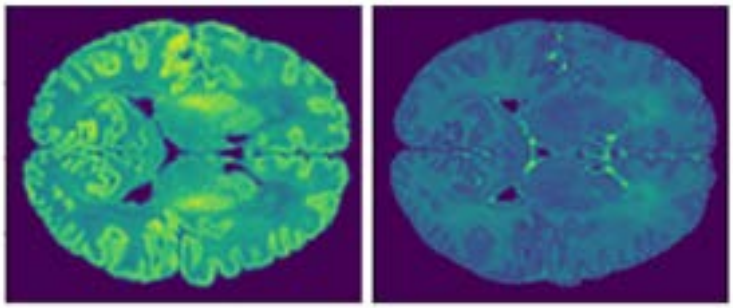

Figure 3.2: Paired T1 and T2 MR slices belonging to the same subject and same slice. Left image – T1 MR slice – and right image – T2 MR slice. Both plotted using the same colour map.

testing, with the remaining 90% used for training. This corresponds to 251 pairs of images for training, 14 image pairs for testing, and 14 image pairs left for validation.

3.3 Network Architectures

Of the three image reconstruction networks, an Autoencoder network (described in section 2.1) was firstly implemented, following the code from [44], which uses convolutional and transposed convolutional layers to first encode and then decode information (such as images). The architecture used for this network is represented in Figure 3.3, with the network layer parameters presented in Table A.1 in the appendix.

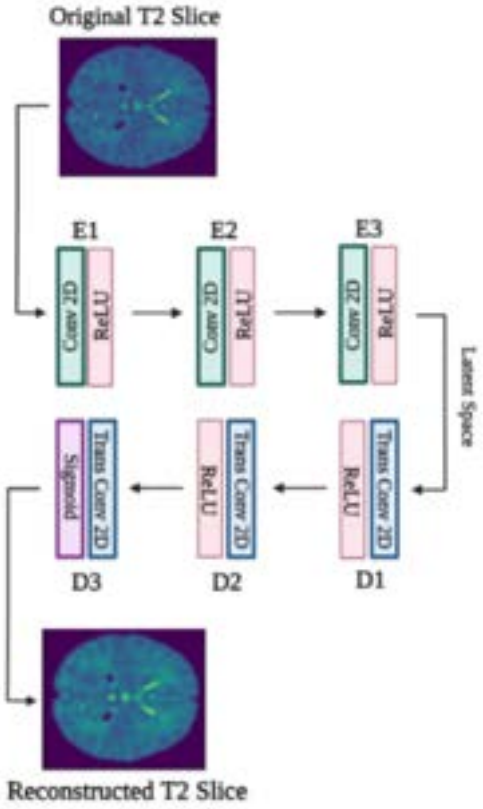

Figure 3.3: Overview of the Autoencoder architecture used. The autoencoder consisted of 3 encoder layers E1-E3, and by 3 decoder layers D1-D3. The latent space corresponds to the stage at which the image is in its most compressed form.

Secondly, a 2D U-net was implemented using the previously built Autoencoder architecture of Figure 3.3 and adding skip-connections to it. This U-net architecture is therefore represented in Figure 3.4, with the network layer parameters (filter size, stride, padding, etc.) presented in Table A.2 in the appendix.

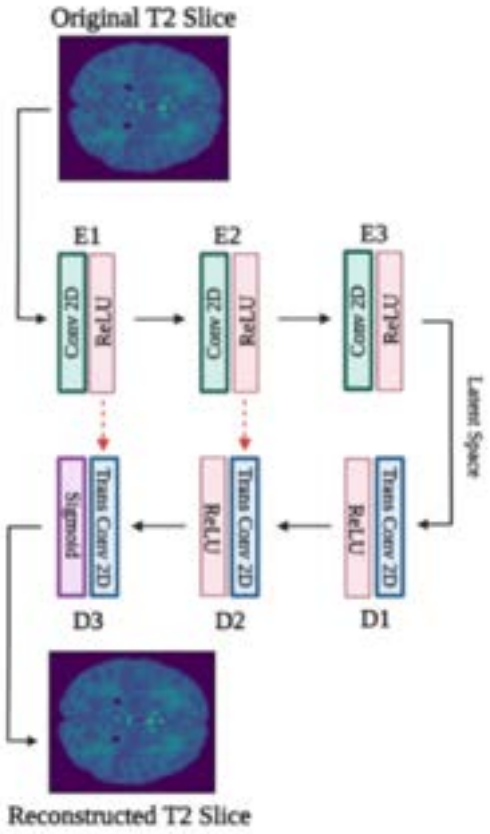

Figure 3.4: Overview of the U-net architecture used. The U-net consisted of the same autoencoder architecture of Figure 3.3 but with the added skip connections (represented by the orange arrow). The latent space corresponds to the stage at which the image is in its most compressed form.

A 3D U-net was also implemented for image translation with its architecture illustrated in Figure 3.5 and the network layer parameters (filter size, stride, padding, etc.) presented in Table A.3 in the annex. In this case, all the network's layers specific for handling 2D data were modified for 3D (such as the convolutional layers), and instance normalisation layers [45] were additionally added. These layers are commonly used in network's architectures, to help improve training and performance.

Next, a 2D WGAN was implemented since it is, in turn, composed of a U-net as its generator and an additional convolutional neural network as its critic - whose architecture follows the work of [46]. The generator reconstructed the images and the critic scored the images passed through the network as either real (original T2 images) or fake (reconstructed T2 images), giving feedback to the generator on how similar the reconstructions were to the original images. The WGAN's generator architecture is therefore the same as the U-net illustrated in Figure 3.4 and the critic's architecture is presented in Figure 3.6, with the network layer parameters (filter size, stride, padding, etc.) presented in Tables A.2 and A.4, respectively, in the appendix.

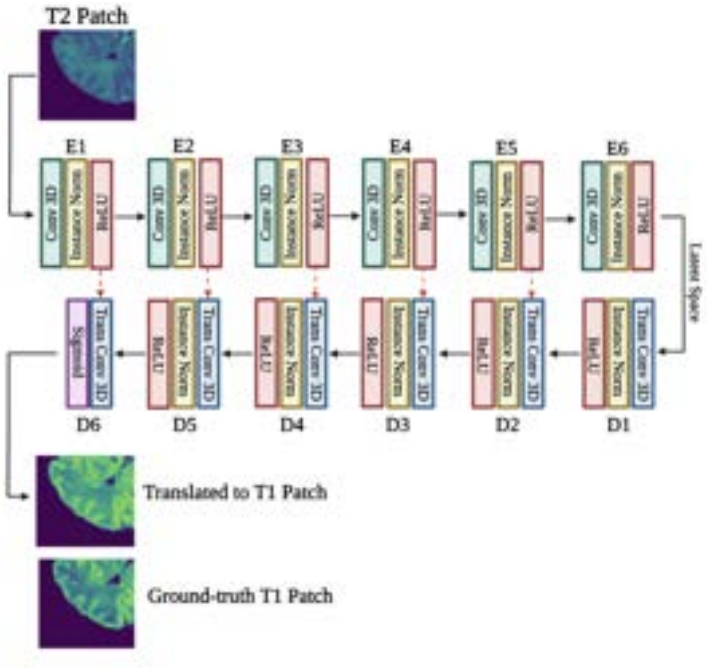

Figure 3.5: Overview of the 3D-Unet architecture used. The U-net consisted of encoder layers E1-E6, and by decoder layers D1-D6, with added normalisation layers and activation function ReLU. The last layer of the network was a sigmoid function. The red arrows between the E layers and D layers represent the skip connections. The latent space corresponds to the stage at which the image is in its most compressed form.

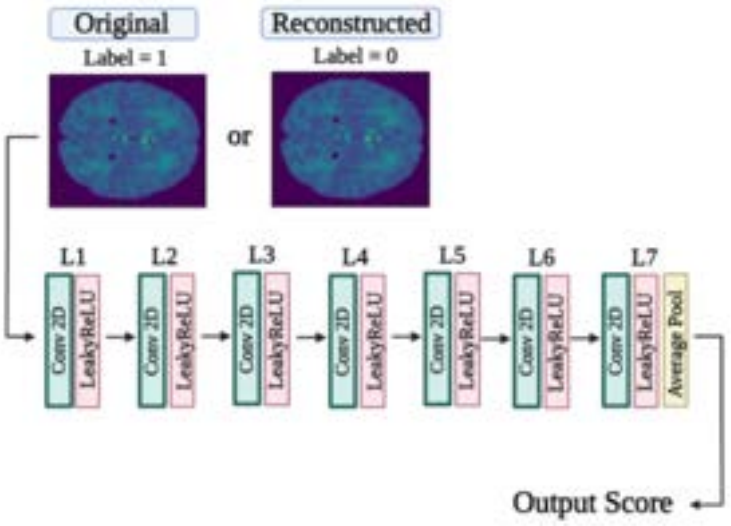

Figure 3.6: Overview of the Critic network architecture used in the WGAN. The Critic consisted of convolutional layers with a LeakyReLU activation function (with a negative slope of 0.2), and a final layer of adaptative average pooling.

Finally, a 2D CycleGAN was implemented. Its architecture consists of 2 generators and 2 discriminators. The generators' architecture is based on a U-net (illustrated in Figure 3.7) and the discriminators follow the architecture of a PatchGAN (illustrated in Figure 3.8). The layer parameters (filter size, stride, padding, etc.) for the generators are presented in Table A.5 and for the discriminators in Table A.6, in the appendix.

Additionally, the 2D CycleGAN was modified to become a 3D CycleGAN and implemented for patched-based training in the image translation task. The architecture of the CycleGAN was modified to fit the 3D patch data by replacing convolutional layers and normalisation layers for their 3D versions.

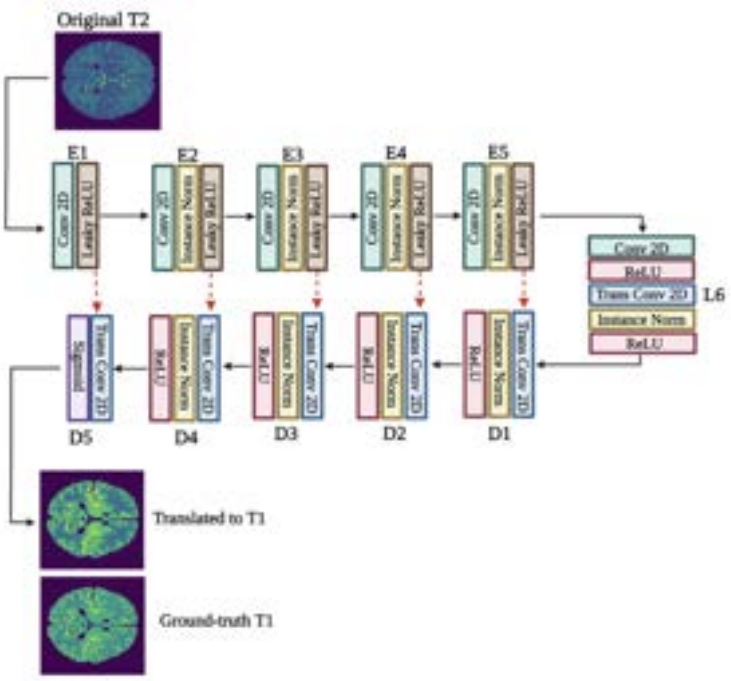

Figure 3.7: Overview of the 2D-Unet architecture used for the generator of the 2D CycleGAN. The U-net consisted of encoder layers E1-E6, and by decoder layers D1-D6, with added normalisation layers and activation function ReLU and LeakyReLU (with a negative slope of 0.2). The last layer of the network was a sigmoid function. The red arrows between the E layers and D layers represent the skip connections.

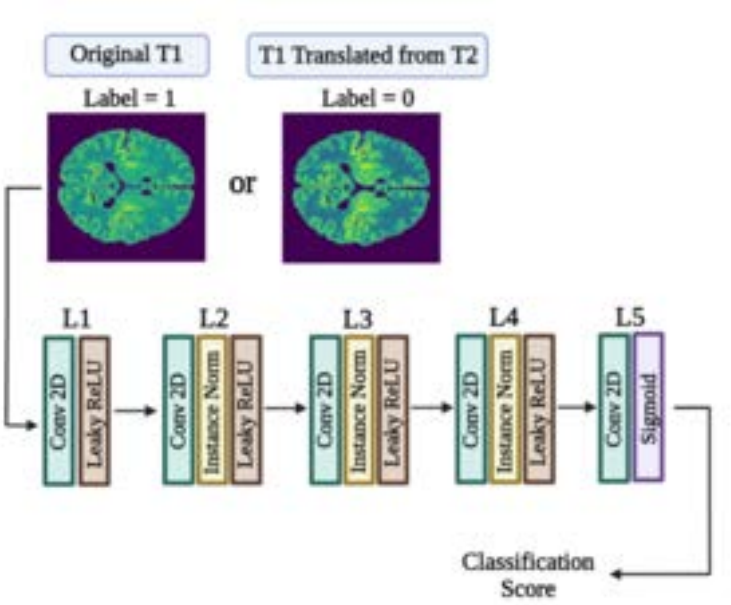

Figure 3.8: The PatchGAN discriminator used in the 2D CycleGAN. It consists of layers L1-L5 built with convolutional operations, normalisation and activation functions (LeakyReLU with negative slope of 0.2).

Therefore, the generators and discriminators of the 3D CycleGAN have the same architectures illustrated in Figure 3.7 and Figure 3.8, respectively, with the difference of having 3D layers instead of 2D.

The layer parameters (filter size, stride, padding, etc.) for the generator are represented in Table A.5 in the appendix, and the layer parameters for the discriminator are identical to the 2D version (but using the parameters in the 3D layers) and are presented in Table A.6.

3.4 Experimental Set Up

3.4.1 Goal

The experiments can be divided considering two different goals: image reconstruction and image translation. For image reconstruction, the Autoencoder, U-net and WGAN networks were trained and tested to reconstruct 2D T2 MR images. After evaluating the results for this task, the network that showed visibly better reconstructions (the U-net) was modified for translating T2-to-T1 MR scans, alongside a CycleGAN. For image translation, there was also the goal to compare and optimise two different training methodologies: a 2D slice training approach and a patched-based training approach (with 3D data). The 2D approach consisted of passing through the networks the entire slice while the patched-based approach trained the networks only with patches belonging to the whole 3D image.

Translated images were evaluated using image quality metrics: MAE, PSNR and FID, with the goal of comparing all networks through analysing the differences in quality of 2D and 3D translations.

The MAE is calculated as the mean of the absolute error between each pixel value in the ground truth and generated image (with values ranging between 0 and 1). Although this metric does not reveal anything about similarities in structures of the image, a large error means the intensity values at each voxel differ a lot from those in the original image. The PSNR metric evaluates the generated image noise compared to the ground-truth image, in decibels (dB). It is often used to measure image quality after encoding and decoding losses, with higher PSNR values indicate better image quality results. Finally, the FID score is used to evaluate the quality of the generated images and measure the similarity between two different images (the ground-truth and the translation), with lower FID values indicate better quality and similarity of the generated images [25].

These evaluation metrics helped in understanding the best practices to implement in the networks and what needs to be optimised in Chapter 4, when dealing with anomaly detection with PET-MR data.

3.4.2 Methodology and Training Parameters

i) Image Reconstruction

To train the image reconstruction networks, a slice-based approach was implemented. For all models, the same number of epochs and batch size was defined (30 in both cases) and the loss function (MAE [47]), optimisers (Adam [48]) and remaining parameters used for training the networks, were kept as in the original examples [44, 46, 49]. Hyperparameters used for all models are presented in Table 3.1.

Table 3.1: Parameters and loss functions used to train the 2D reconstruction networks. Hyperparameters chosen to train include: batch-size, learning rate, the β parameter of the Adam optimiser chosen, the critic iterations (the number of iterations of the critic per generator iterations), and the λ values applied to the gradient penalty.

	Autoencoder	**U-Net**	**WGAN**
Input Image Size	196 x 230	196 x 230	196 x 230
Epochs	30	30	30
Batch-Size	30	30	5
Learning Rate	1e-3	1e-3	1e-4
Adam Optimiser	β = (0.9, 0.999)	β = (0.9, 0.999)	β = (0.9, 0.999)
Critic Iterations	-	-	5
λ Gradient Penalty	-	-	10
Loss Function	MSE loss	MSE loss	Wasserstein distance loss

At test time, all images from the testing set were passed through the trained models for reconstruction, and three examples derived from each network were randomly selected to display the reconstruction results. To better compare the results among the models, all the 3 distinct models were trained for the same number of epochs.

ii) Image Translation

2D slice approach (2D translation)

For the T2-to-T1 MR image translation task, a U-net and a CycleGAN were compared, for the task of translating, T2 MR slices, to resemble the appearance of T1 MR slices. In all instances the anatomy should be preserved between input and translated slices.

For the U-net, the same loss functions and optimisers, used for image reconstruction, were maintained. However, the network loss was now computed between the output of the networks (the translated MR slice) and the corresponding ground-truth T1 MR scan. For the CycleGAN, the loss functions, optimisers and other hyperparameters reflect those used in the original code from [50]. Information about the parameters used to train both networks can be found in Table 3.2.

Table 3.2: Parameters and loss functions used to train the 2D networks. Hyperparameters chosen to train include: batch-size, patch-size, initial learning rate, epoch decay (after how many epochs the learning rate starts to decay linearly to 0), the β parameter of the Adam optimiser chosen, the λ values applied to the L1 loss and identity loss.

	U-Net	**CycleGAN**
Input Image Size	256 x 256	256 x 256
Epochs	200	100
Batch-Size	18	1
Learning Rate	1e-3	0.0002
Epoch Decay	-	100
Adam Optimiser	β = (0.9, 0.999)	β = (0.5, 0.999)
L1 Loss	-	0.001
λ Id	-	0.5
Loss Function	MSE loss	L1 distance: Cycle-consistency and Identity loss MSE loss: Discriminators

The U-net and CycleGAN networks were trained until the validation loss and image quality metrics in validation samples stabilised, indicating that the image quality was no longer improving. At test time, all 2D images were passed through the networks to obtain translations. The results were evaluated by calculating the mean of the image quality metrics used, for all test images.

iii) Patched-based approach (3D translation)

The U-net and CycleGAN previously used in the whole-image based training were modified for a patch-based approach.

In this training method, patches were randomly selected from the whole 3D images. Therefore, the networks learned to translate a T2 patch to a T1 patch and not the complete 3D T2 MR image to a complete 3D T1 MR image. Regarding the random patch selection, a Random Spatial Crop Samples transform [51] was used. The transform crops the image in a random location, with a chosen size, to generate a list with a defined number of sampled patches (see Figure 3.9 for an example).

For inference, the sliding window method [52] was used (with default parameters) since the goal was

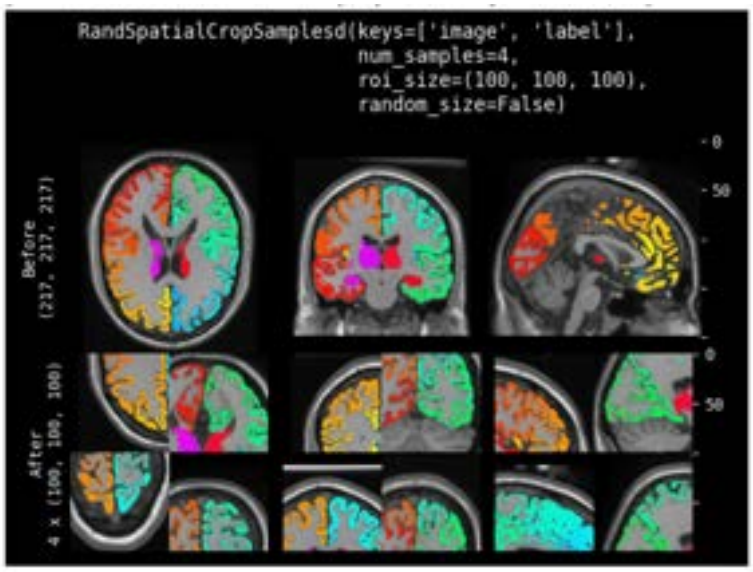

Figure 3.9: Illustration showing an example of the RandSpatialCropSamplesd function performing random sampling of patches. The parameters of the function were defined for the patch size to be 100x100x100 and to sample 4 patches. The before (whole image of size 217x217x217) and after (4 random patches sampled of size 100x100x100) are represented in the image. Retrieved from [51].

to train the network using patches but translate the whole 3D image in testing. This works by passing the whole image as a series of patches through the model and joining the outputs so the whole image can be constructed again. Figure 3.10 illustrates the sliding window method used for inference.

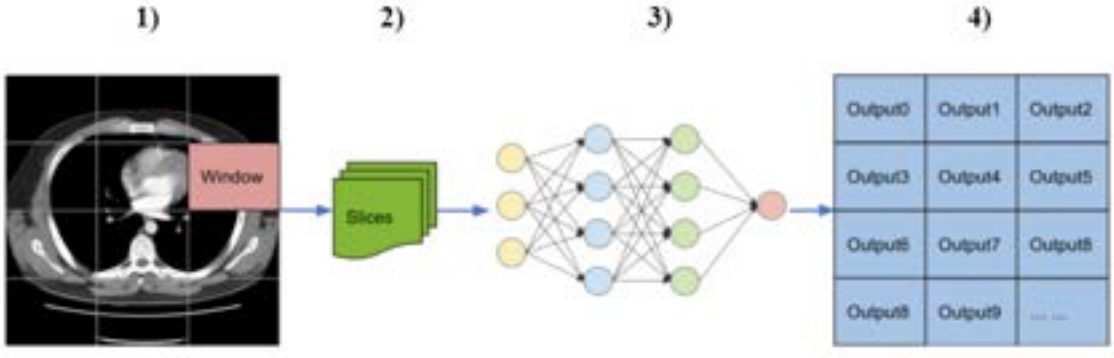

Figure 3.10: Operation method of the sliding window function. 1) Generation of slices from window. 2) Construction of batches. 3) Passing patches through network. 4) Connection of all outputs. Retrieved from [53].

For both 3D networks, a patch size of 128 x 128 x 128 was chosen to train the networks, with a sample number set to 5. The loss functions, optimisers and other parameters used for training the 3D U-net and CycleGAN are represented in Table 3.3.

Table 3.3: Parameters and loss functions used to train the 3D networks. Hyperparameters chosen to train include: batch-size, patch-size, initial learning rate, epoch decay, the β parameter of the Adam optimiser chosen, the λ values applied to the L1 loss and identity loss.

	U-Net	**CycleGAN**
Full Image Size	256 x 256 x 256	256 x 256 x 256
Input Patch Size	128 x 128 x 128	128 x 128 x 128
Epochs	800	200
Batch-Size	1	1
Learning Rate	1e-3	0.0002
Epoch Decay	-	100
Adam Optimiser	β = (0.9, 0.999)	β = (0.5, 0.999)
L1 Loss	-	0.001
λ **Id**	-	0.5
Loss Function	MSE loss	L1 distance: Cycle-consistency and Identity loss MSE loss: Discriminators

3.5 Results

3.5.1 Image Reconstruction

Figures 3.11, 3.12 and 3.13 represent the results of the image reconstructions, after 30 epochs of training, obtained from the 2D networks: AE, U-net and WGAN, respectively.

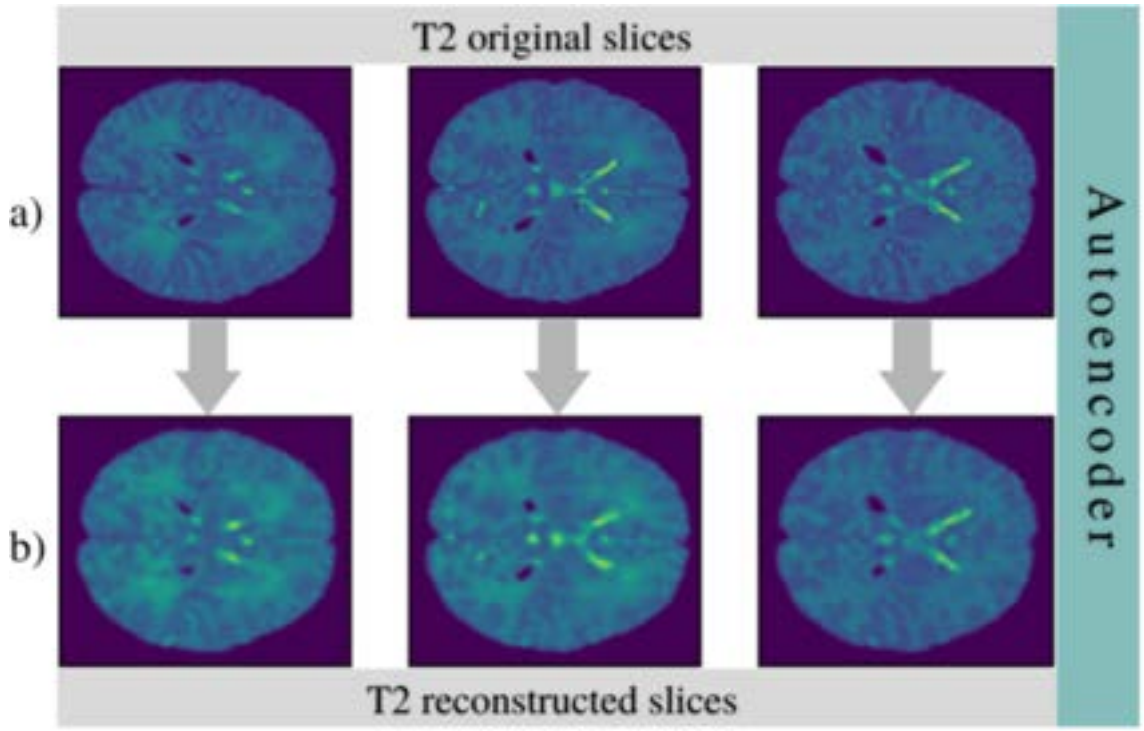

Figure 3.11: Results of image reconstruction using the autoencoder network after 30 epochs. a) Randomly selected examples of the T2 MR ground-truth slices. b) The corresponding reconstructions obtained by the autoencoder network.

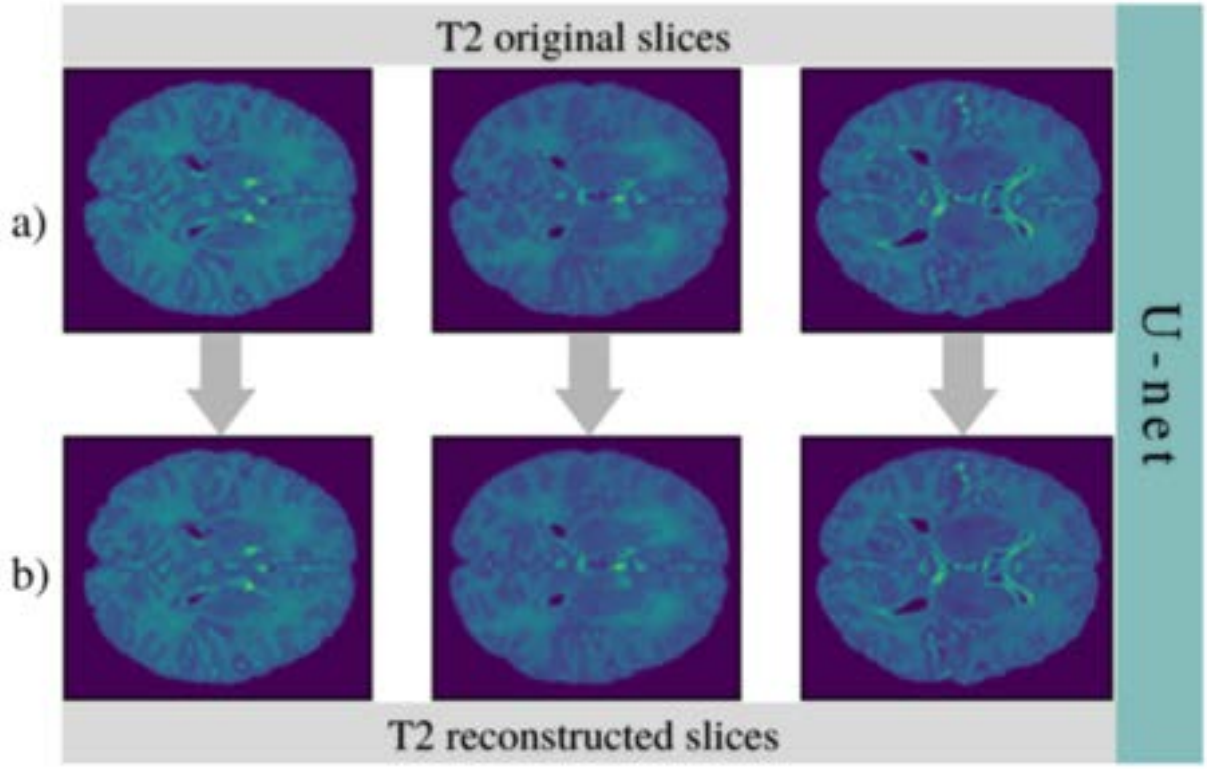

Figure 3.12: Results of image reconstruction using the U-net network after 30 epochs. a) Randomly selected examples of the T2 MR ground-truth slices. b) The corresponding reconstructions obtained by the U-net network.

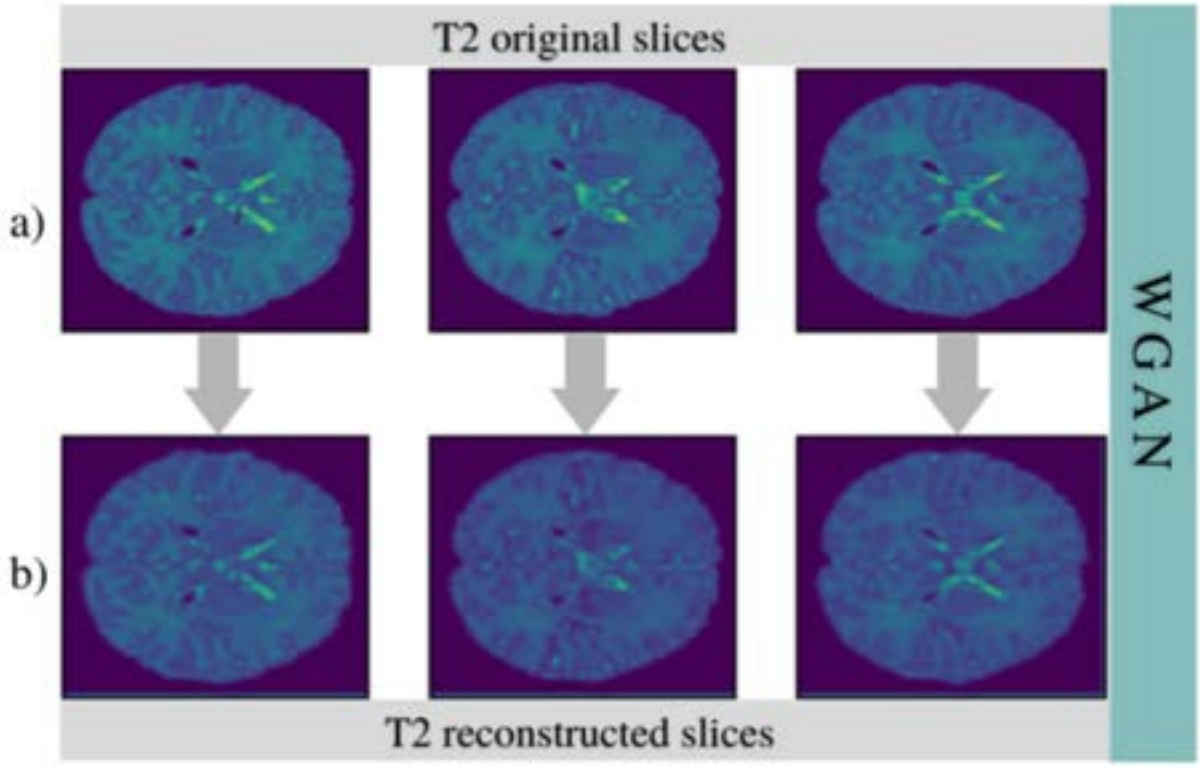

Figure 3.13: Results of image reconstruction using the WGAN after 30 epochs. a) Randomly selected examples of the T2 MR ground-truth slices passed through the WGAN for reconstruction. b) The corresponding reconstructions obtained by the WGAN.

3.5.2 Image Translation

i) 2D Networks

Figures 3.14 and 3.15 represent the results of the T2-to-T1 MR image translation obtained from the 2D networks: U-net and CycleGAN, respectively. The quality evolution of the translated T1 MR images through the several training epochs is also shown.

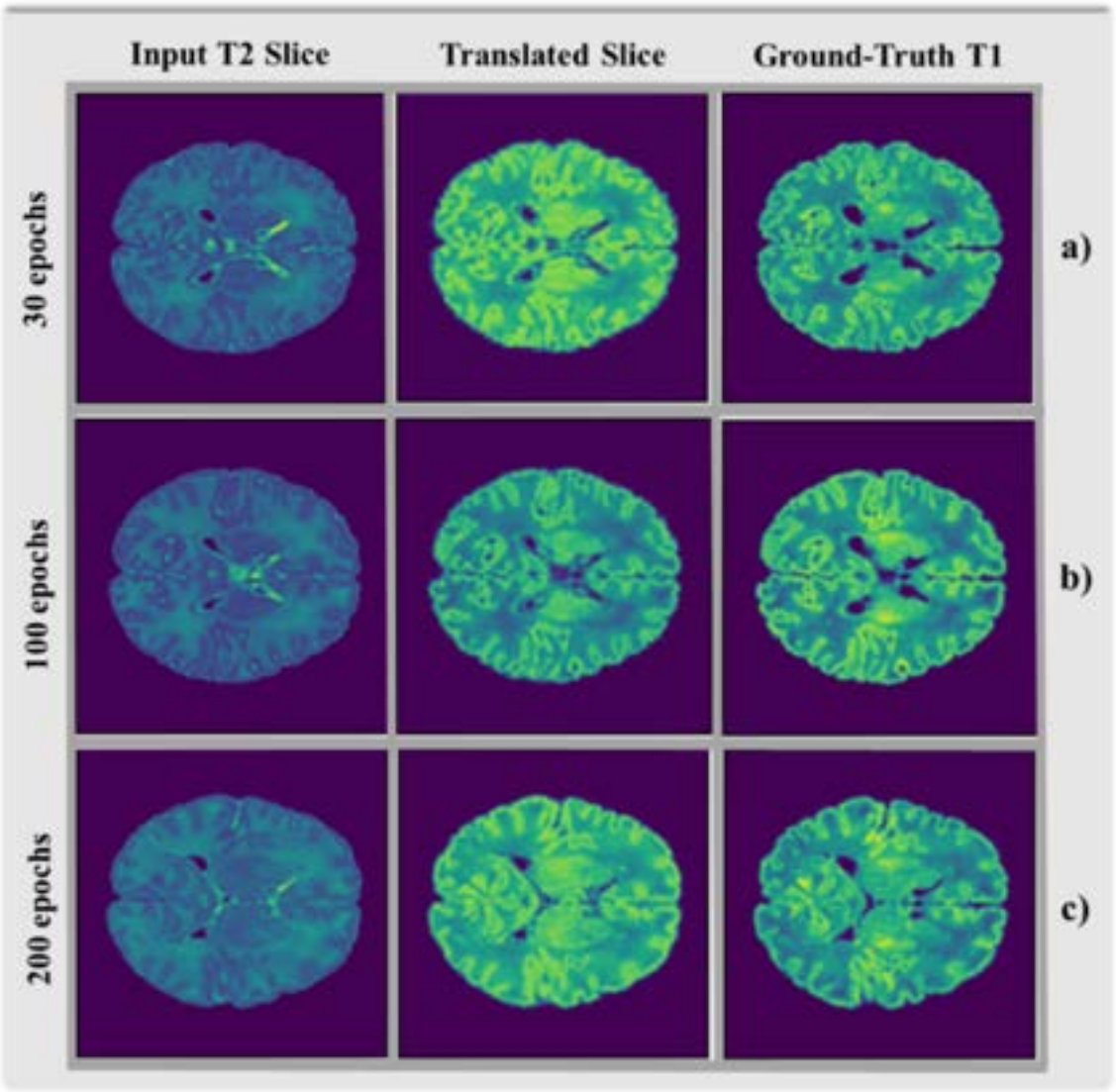

Figure 3.14: T2-to-T1 translation using three different test sample images. Example of the translation achieved by the U-net compared to the ground-truth T1 image. Input of the network was the corresponding T2 MR image. a) 30 epochs, b) 100 epochs, c) 200 epochs.

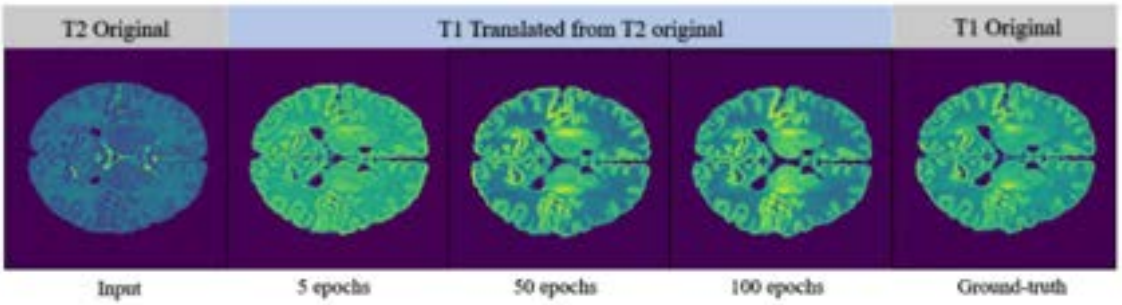

Figure 3.15: Figure 3.15: T2-to-T1 translation using a test sample image. Example of the translation achieved by the CycleGAN network when trained for up to 100 epochs, compared to the ground-truth T1 image. Input of the network was the corresponding T2 MR image.

Table 3.4 represents the values for MAE, PSNR and FID as evaluation metrics for the translated T1 images for the 2D trained networks, at test time. All results were calculated between the ground truth T1 slices and the corresponding synthesized T1 slices generated by the corresponding network.

Table 3.4: Image quality evaluation metrics - MAE, PSNR and FID - for the translated T1 images. The mean value with associated standard deviation for each metric is presented for the 2D U-net and 2D CycleGAN. Evaluation metric values correspond to 200 epochs of training for the U-net and 100 epochs of training for the CycleGAN.

	MAE	**PSNR (in dB)**	**FID score**
2D U-net	0.00771684 ± 1.15e-05	37.16 ± 6.07	23.95
2D CycleGAN	0.0145351 ± 5.21e-05	30.69 ± 0.82	14.70

ii) 3D Networks

Figures 3.16 and 3.17 illustrate the results for the 3D U-net and 3D CycleGAN applied for T2 to T1 image translation. The quality evolution of the translated T1 MR images through the several training epochs is also shown.

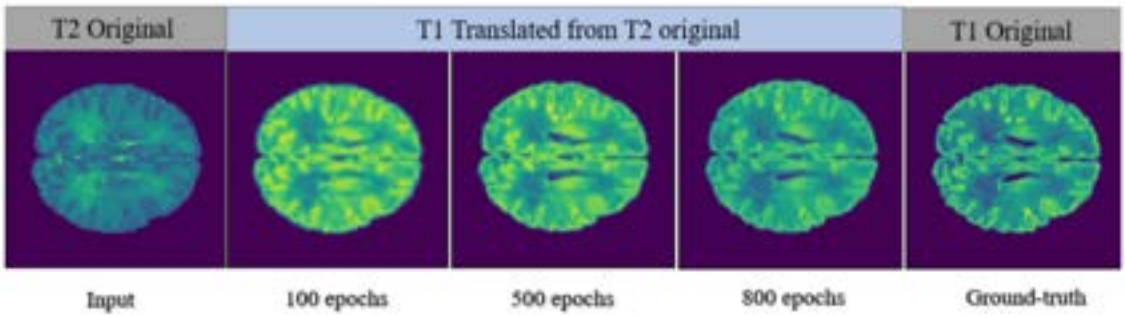

Figure 3.16: Figure 3.16: T2-to-T1 translation using a test sample image. Example of the translation achieved by the 3D U-net when trained for up to 800 epochs, compared to the ground-truth T1 image. Input of the network was the corresponding T2 MR image.

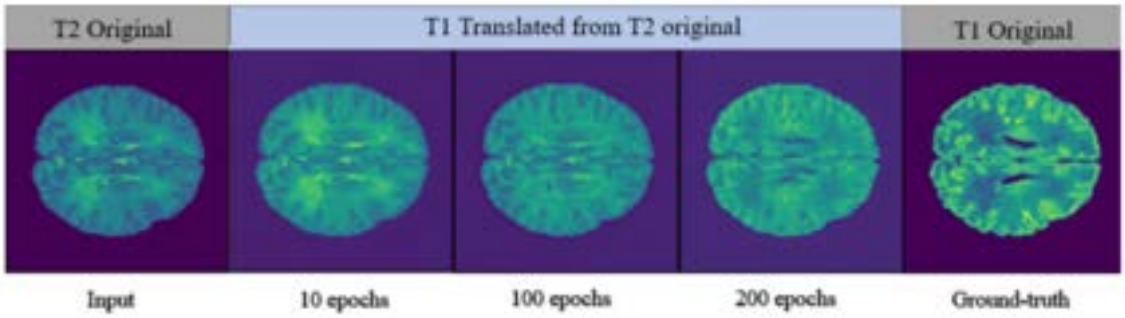

Figure 3.17: T2-to-T1 translation using a test sample image. Example of the translation achieved by the 3D CycleGAN network when trained for up to 200 epochs, compared to the ground-truth T1 image. Input of the network was the corresponding T2 MR image.

Table 3.5 represents the values for MAE, PSNR and FID as evaluation metrics for the translated T1 images for the 3D trained networks, at test time. All results were calculated between the ground truth T1 images and the corresponding T1 generated images by the corresponding network. However, to better compare both 2D and 3D evaluation metrics, only the middle slices of the 3D translated images were used to calculate the metrics between the T1 translated slice and its ground-truth slice.

Table 3.5: Image quality evaluation metrics - MAE, PSNR and FID - for the translated T1 images. The mean value of all test images, with associated standard deviation, for each metric are presented for the 3D U-net and 3D CycleGAN. Evaluation metric values correspond to 800 epochs of training for the U-net and 200 epochs of training for the CycleGAN.

	MAE	**PSNR (in dB)**	**FID score**
3D U-net	0.00532827 ± 2.56e-06	39.83 ± 2.67	25.59
3D CycleGAN	0.04356290 ± 2.91e-06	26.60 ± 1.71	56.68

3.6 Discussion

3.6.1 Image Reconstruction

From the results in Figures 3.11, 3.12 and 3.13, it is possible to compare the three different reconstructions achieved by the different networks and understand their strengths and limitations.

It is visually possible to observe that the U-net, through its ability to better reconstruct finer details in images, obtained reconstructions with more quality, illustrated in Figure 3.12, compared to the more visually blurry reconstructions obtained by the Autoencoder in Figure 3.11. The skip-connections added to the Autoencoder network therefore delivered the expected results, which accordingly match the theoretical concepts behind it (mentioned in section 2.1).

The WGAN, however, was not able to reproduce as detailed results (visibly more pixelated in Figure 3.13) as the U-net. These results were expected since WGANs' unsupervised training requires more epochs to be able to reach an equilibrium between the generator and the discriminator. The usefulness and powerfulness of these more complex GANs are expected to be emphasized in the anomaly detection task of Chapter 4. However, through this result, it was possible to analyse future ways of optimising the WGAN for image reconstruction, implemented in Chapter 4. Therefore, taking into consideration the good reconstruction results from the supervised U-net, in the next chapter, an additional supervised loss will be added to this network, to improve image reconstruction quality and promote quicker network

training. As it will be presented in the next chapter, the optimisations implemented did indeed contribute to obtaining quality image reconstructions in the 3D WGAN for anomaly detection.

One limitation of the reconstruction experiment was that no quantitative metrics were used to evaluate the image reconstruction quality, with it only being interpreted by visually comparing the networks. This is modified for the image translation task, where MAE, PSNR and FID scores are presented to help interpret the network's results.

3.6.2 Image Translation

For 2D image translation (Figures 3.14 and 3.15), the U-net shows better image quality with lower MAE and higher PSNR values compared to the CycleGAN (Table 3.4). The FID score however gets a better result in the CycleGAN, indicating a higher similarity of the ground-truth and translated T1 slices. This suggests that although the CycleGAN demonstrates overall lower image quality in translation, it has a closer image distribution to the ground-truth T1 images. In fact, it is not surprising that the CycleGAN demonstrates lower image quality, taking into consideration that it trains in an unsupervised way (meaning that there was no need to have paired T1 and T2 slices during training), unlike the U-net. Therefore, it was possible to observe the potential that both 2D networks possess in translating T2-to-T1 MR slices with good image quality, both quantitively and visually.

Shifting the focus to 3D image-translation networks, evaluation metric results (Table 3.5) and visual results (Figures 3.16 and 3.17) show the image quality degradation that can arise from the increase in complexity of training 3D images compared to the 2D networks. Although the 3D U-net matched (and even slightly improved) the 2D U-net performance in terms of MAE and PSNR values for translation, the 3D CycleGAN struggled with patched-based training, indicating worse values for MAE, PSNR and FID score. It is therefore possible to visually notice the degradation in image quality of the translations in Figure 3.17, which contains a perceptible intensity voxel shift in the background.

In fact, it has also been reported by the work of [54] that a CycleGAN model obtained worse quantitative and visual performance results compared to a supervised U-net, in an image translation task with MR and CT images. Possible reasons for this include the need for great amounts of data, especially in unsupervised methods, to improve network performance. Another factor in this case could be the use of patch-training in the CycleGAN, meaning the whole image is never fully passed through the networks, only patches belonging to it. This could explain the difficulty of the CycleGAN, an already unsupervised network, in obtaining good translations compared to the U-net.

However, the results from the supervised patched-based trained U-net showed great potential in obtaining good translation results while preserving computational memory resources – these factors motivated the use of patched-based training in Chapter 4.

Chapter 4 therefore aimed to improve the patched-based training of the CycleGAN for anomaly detection, by allowing longer training times (closely evaluated through training and validation losses) and adding more data augmentation methods. Different activation functions were also tested in the networks to try minimising the intensity differences of the voxels seen on the background of the images.

In conclusion, these results provided clarity over which network architectures and training methods should be followed, as well as which factors should be improved or combined to optimise the networks for anomaly detection in Chapter 4. The following chapter therefore takes advantage of the initial work presented here and applies these networks to create anomaly detection machine learning models using PET-MR data, based on these two different methods: image reconstruction and image translation.

Chapter 4

PET-MRI Anomaly Detection using Deep Generative Modelling

4.1 Motivation: Detection of Focal Cortical Dysplasia in Neuroimaging

FCDs are malformations of cortical development that belong to a group of rare disorders that are commonly manifested alongside developmental delay, cerebral palsy and/or seizures. [55]

The identification and classification of these malformations from neuroimaging experts can be quite challenging, since FCDs reflect small, localised errors created during the development of of the outer surface of the brain. These FCDs represent a spectrum of focal brain malformations, categorised into three subgroups (Type I, II and III), which reflect the diverse types of disordered cortical lamination [55]. Therefore, these 3 types of FCDs relate to distinct types of errors in cortical development, each with examples illustrated in MR brain scans in Figure 4.1.

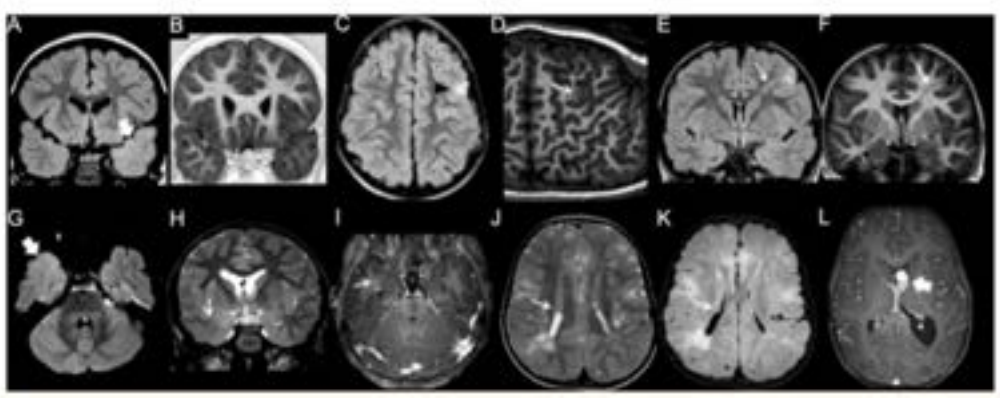

Figure 4.1: MR brain images containing visible FCDs. (A and B) Images of Type I FCD, where the left temporal pole is slightly smaller than the contralateral one and abnormal myelination (in the blurred grey-white matter junction) compared to the contralateral side (indicated by the arrow). (C and D) Images of Type IIa FCD, where the arrowhead indicates lesion in the left frontal lobe and the white arrow points to the focal blurring of the grey-white matter junction indicating another lesion. (E and F) Images of Type IIb FCD, both indicated by the arrows corresponding to regions of abnormalities: hyperintensity in FLAIR image and hypo intensity in T1 image, respectively. (G-I) Images of Type III FCD, where there is a slightly blurred grey-white matter junction (represented by the thick arrow). The thin arrow in images H and I indicate a developmental venous anomaly. (J-L) Tuberous sclerosis complex. The thin arrows show nodules that are associated with cortical tubers and white matter lesions. The thick arrow indicates a tumour [55].

As observed in Figure 4.1, FCDs are very variable in presentation and commonly present themselves as only subtle changes in MR scans. Imaging findings in MRI indicating FCDs can, therefore, be in the form of minimal blurring of cortex-white matter junction, focal changes of cortical thickness, brain folds

with abnormal size (either larger or smaller), presence of tumours, and so on. This means FCDs can either be quite visible in MR scans (as observed in Figure 4.4 – a tumour) or, at most times, not visible at all (represented in Figure 4.3 where the FCD is not visible in MRI, but it is in the PET scan in the same brain location of the patient).

When FCDs are not visible in MRI, hybrid imaging such as the combination of MRI and PET modalities becomes valuable in identifying these lesions [56]. In fact, the work of *Salamon et al.* [57] concluded that incorporating [18F] fluorodeoxyglucose - positron emission tomography (FDG-PET)/MRI coregistration into the presurgical evaluation of patients with lesions had the ability to enhance the identification of FCD in the brain, subsequently resulting in more successful surgical treatment of epileptic patients.

In FDG-PET scans, FCDs manifest themselves as focal or regional abnormal hypometabolic areas [58] (focal reductions of glucose metabolism) in the brain, which are commonly represented by more "bluish" regions, such as the example in Figure 4.3 - abnormally larger hypometabolic region (in blue) in the right temporal lobe compared to the left temporal lobe.

In an equivalent way, specialists that examine medical images, search for abnormal regions that differ from their prior experience of what a healthy scan shows [14]. Naturally, this process of labelling the regions where the lesions are present is very time-consuming and requires specialised neurologists. For this reason, automated approaches emerge as a solution to help guide experts in their diagnosis, potentially alleviating this lengthy process.

Therefore, this chapter describes a patch-basis detection approach inspired by the work of *Yaakub et. al.* [13] (analysed in section 2.1) with the same PET-MR dataset, by applying 2 different GANs for both image reconstruction and image translation methods, to detect FCDs in epilepsy patients, combining both imaging modalities. It also takes inspiration from methods used in the work of *Sun et al.* [14] - specifically in creating a personalised anomaly loss - by taking advantage of the lesion masks available for the data. The general idea of this project was therefore to create a patch-wise approach to try and identify lesions in the dataset and at the same time deconfound acquisition noise and normal cortical variabilities in PET-MR data of brains.

In this chapter, a description of the dataset used and pre-processing strategies, are presented in the following section 4.2. In section 4.3, the general training and testing methodologies applied to both WGAN and CycleGAN are described, including the data augmentation strategies implemented, architecture of the network, learning rates, loss functions and other hyperparameters. All experiments were built using the Pytorch and MONAI machine learning frameworks [37, 38], and trained using a NVIDIA Titan RTX GPU with 24 GB of RAM. Finally, results are presented in section 4.4 and their discussion in section 4.5.

4.2 Dataset and Pre-processing

4.2.1 Data Structure

The dataset was comprised of 31 MR and PET scans of patients with drug-resistant epilepsy (with dimensions of 230 x 160 x 230). The MR and PET scans were acquired on the same day using a whole-body GE Discovery 710 PET/CT system and a 3T Siemens Biograph mMR PET-MR scanner. Data acquisition for this dataset included a 15-minute [18F] FDG PET scan on the PET/CT system 30 min post-injection and a 3D T1 MP RAGE (magnetization-prepared rapid acquisition gradient echo) scan on the PET-MR system [13]. All data acquisition information was retrieved from [13], with MRI TI values and other echo values not mentioned in this paper. Information about age, gender, category, and the

region where the lesion was found is presented in Table A.7.

All subjects had their scans evaluated by two consultant Nuclear Medicine physicians and 15 subjects had their scans additionally examined by Professor Alexander Hammers, Head of King's College London Guy's and St Thomas' PET centre [59]. The patients' scans could belong to one of the following categories: MR+PET+ (lesion was visible in both MR and PET scans), MR-PET+ (lesion was not visible in the MR scan but visible in the PET scan) or MR-PET- (lesion was not visible in the MR nor the PET scans). From this visual inspection, the positive (+) scans were given labels indicating the suspected lesion location and used to create a mask of potentially pathological tissue in the scans (described in section 4.2.3). These lesion masks were then used to train the networks described in section 4.4.

From the entire dataset, the patients belonging to the MR-PET- category were excluded since the location of the lesion was unknown and therefore not useful for the intended goal of this project. The total number of patients used in this project was therefore reduced to 22 pairs of PET-MR scans.

Figures 4.2, 4.3 and 4.4 illustrate examples of 3 different lesions present in patients' scans. Figure 4.2 shows the PET and MR scans of patient mMR_BR1_050, containing abnormalities seen in both MR and PET scans, which were classified as a FCD of type I by the physicians. In Figure 4.3, the PET scan of patient mMR_BR1_022 illustrates a small abnormality (hypometabolic region) which is not visible in the MR scan. This patient is therefore under the category of MR-PET+. The patient was diagnosed with epilepsy originating in the right temporal-frontal lobe. Finally, Figure 4.4 illustrates a clearly visible lesion in both MR and PET scans, identified as a dysembryoplastic neuroepithelial tumour by the physicians. Figures 4.2-4.4 illustrate the diversity of lesions found across patients' brains in this dataset. These lesions can be either very subtle (Figure 4.2-4.3), sometimes not visible in MR scans, or visually distinct in both scans – just as Figure 4.4 where it is possible to identify a tumour.

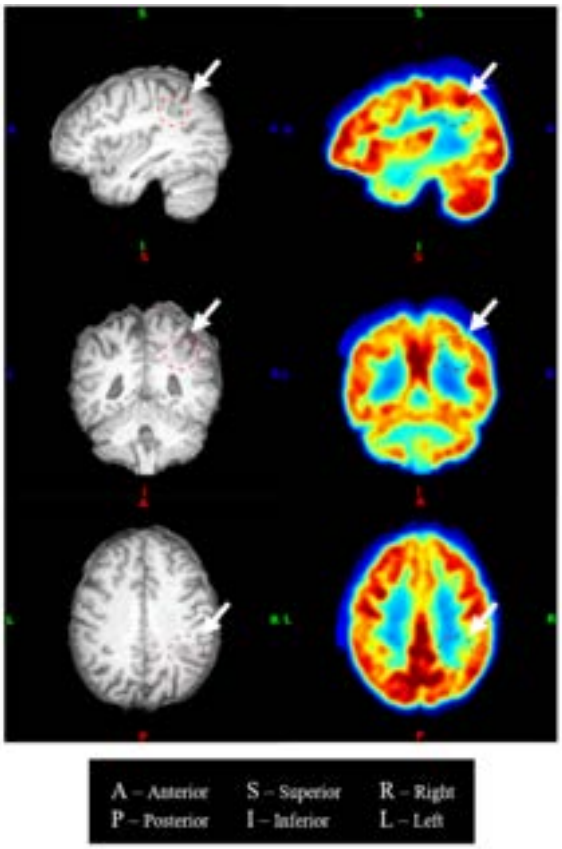

Figure 4.2: Example of 2 pairs of slices (both with sagittal, coronal, and axial schemes) of MR and PET scans of the same patient (mMR_BR1_050), showing the hypometabolic region where the lesion is located (red circle and white arrow). In this patient's case, the lesion is visible in both MR and PET scans. The FCD was classified by the physicians as type I.

There have been several studies [57, 60] showing the advantage of using both PET and MR modalities together to identify FCD, especially in cases where the lesion is visually difficult to identify in either the PET or the MR scan. In this project, the two nuclear physicians visually inspected the MR and PET scans separately unlike Professor Hammers, who examined both PET and MR scans of the patients alongside each other in order to create a clinical label of the suspected lesion location. Having this in mind, Professor Hammer's clinical labels were taken into consideration when available, over the ones noted by the two nuclear physicians - the examination was only done by Professor Hammers for 15 of the 22 patients. It is worth mentioning that the image specialists sometimes disagreed on the location of the lesions, which also occurred between the nuclear physicians and Professor Hammers in this dataset. Some of the FCDs in this dataset were also sometimes noted by the specialists as being extended and/or disperse through a lobe or several lobes of the brain, with no focal point.

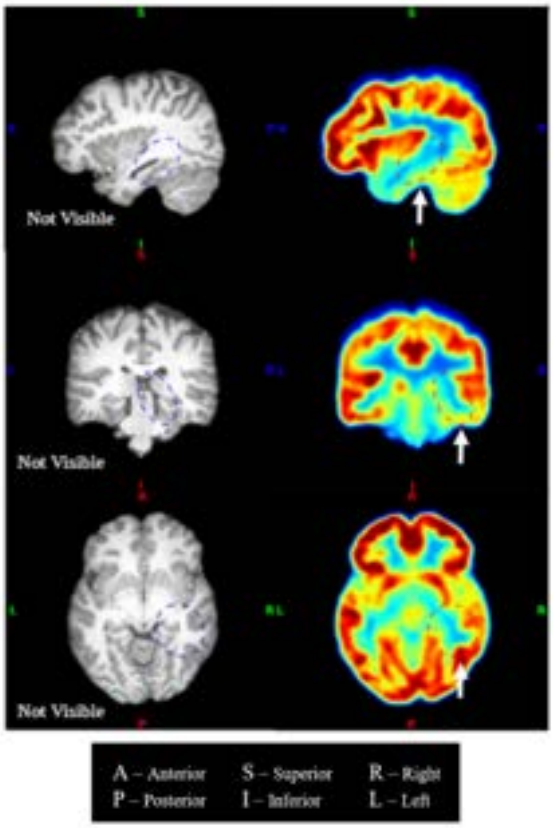

Figure 4.3: Example of 2 pairs of slices (both with sagittal, coronal, and axial schemes) of MR and PET scans of the same patient (mMR_BR1_022), showing the hypometabolic region, where the lesion is located (red circle and white arrow). In this patient's case, the lesion is only visible in the PET scan. The patient was diagnosed with suspected right temporal-frontal epilepsy.

This visual inspection of the scans is then sometimes unsurprisingly ambiguous since there is inter-subject variability in the brain and a disagreement among specialists in what is considered a "healthy looking tissue" in the brain. An example of a contributing factor to this is the fact that the temporal lobes on PET scans have characteristically lower uptake values in healthy patients [62] compared to other lobe regions, complicating the detection of abnormalities in these locations since it could simply correspond to normal inter-subject variation or in fact an abnormal hypometabolic region in the scan. The diverse types of FCD present in this dataset therefore constitute a challenge for the networks that aim to detect them, since the lesions among the patients can be heterogeneous in contrast, morphology, and size.

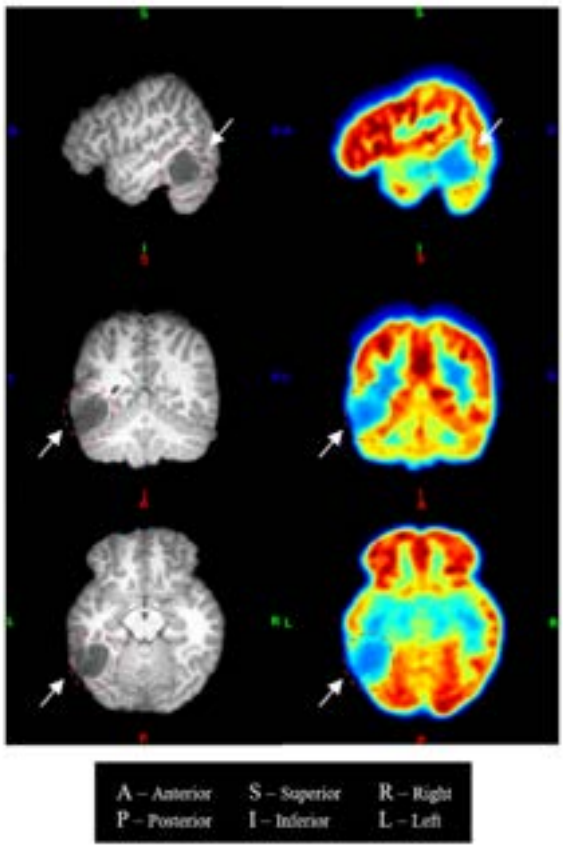

Figure 4.4: Example of 2 pairs of slices (both with sagittal, coronal, and axial schemes) of MR and PET scans of the same patient (mMR_BR1_020), showing the hypometabolic region where the lesion is located (red circle and white arrow). In this patient's case, the lesion is visible in both MR and PET scans. The lesion was classified by the physician as a dysembryoplastic neuroepithelial tumour [61].

4.2.2 Data Pre-processing

The dataset underwent several pre-processing steps before being inputted to the networks to ensure its correct display. The skull was firstly removed from the raw images in order to only keep the brain tissue in all the subjects, since this is where the networks should focus. The process of skull-stripping was therefore the first to be implemented before registering all images to a common space. Secondly, in order to better compare the same regions of the brain in every subject, all images were registered to the same (MNI) standard space. The MR and PET scans had already been previously co-registered to each other (aligned in the same subject space) but not aligned to all the subjects present in the dataset. Remasking was then applied to every scan to further eliminate skull portions that were missing in the initial skull-stripping process. Finally, the data was normalised - by scaling intensity values between 0-1 for every subject through the histogram normalisation method.

The pipeline for the data pre-processing is described in Figure 4.5. Further details associated with each step are presented next with the associated necessity to perform them.

i) Skull extraction

The BET tool [63] from FSL [64] was used to perform brain extraction for every MR and PET scan. This method segments the brain tissue from each image using an intensity-based thresholding approach. The choice of the parameters f (fractional intensity threshold) and g (threshold gradient) were manually selected for every image since different images had different optimal parameters for skull-removal.

This process was repeated for the MR and PET scans of all the patients and visually inspected after

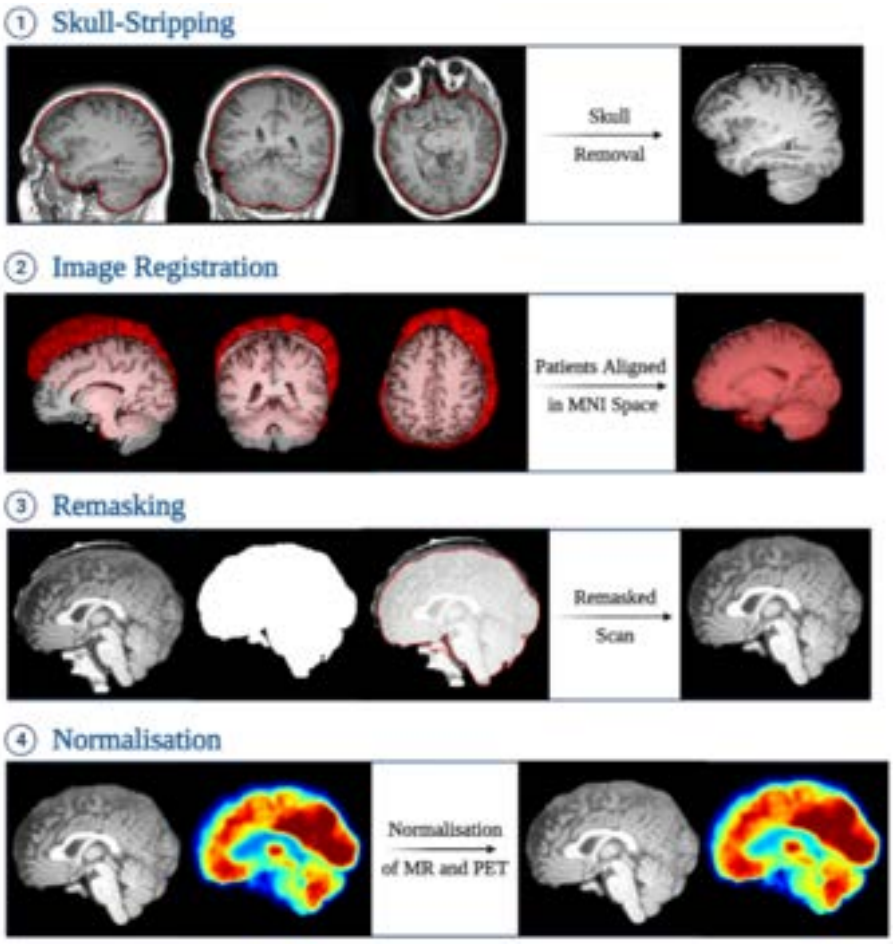

Figure 4.5: Pipeline for pre-processing the dataset. 1: Skull-stripping process with examples of sagittal, coronal, and axial MR slices of patient mMR_BR1_002 before skull removal and same sagittal slice after skull-tripping. 2: Image registration process with examples of sagittal, coronal, and axial MR slices of both patients mMR_BR1_067 (represented in red) and mMR_BR1_047 (represented in black and white) overlapped before image registration (not aligned among each other). The same sagittal slice with both patients overlapped is shown after image registration. 3: Remasking process - the first image represents the original sagittal MR slice of patient mMR_BR1_002, the second image represents the brain mask of the same slice to be applied and the third picture represents the overlay of the brain mask (with the outline in red) and the original image. The last image of the row represents the remasked sagittal slice. 4: Intensity normalisation of both MR and PET scans belonging to patient mMR_BR1_002. The first 2 images represent the scans before normalisation and the last 2 images of that row represent the scans after normalisation.

every skull-stripping operation to ensure the brain tissue was not "cut out" of the scan. If the skull removal resulted in the removal of too much of the brain tissue, changes were made to the f and g parameters for the images to include more skull - Figure 4.7 illustrates an example of an image with parameters that allow for the correct removal of skull and another example of an image with incorrect parameters that remove brain tissue. This process was optimised empirically.

In some images, the BET tool was not able to fully remove the skull without also removing important brain-tissue. In these cases, the brain-tissue and the images were left with some pieces of skull – example in Figure 4.8 – that would be further removed using a remasking method after registration took place. The remasking method will be explained later in this section.

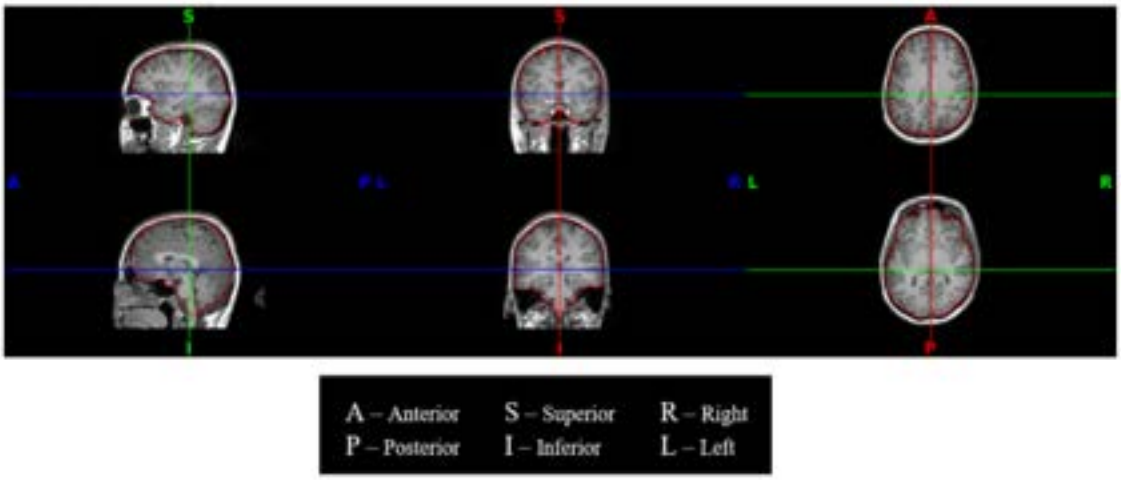

Figure 4.6: Example of 2 pairs of slices (both with sagittal, coronal, and axial schemes) of MR scans of the same patient (mMR_BR1_030), showing the outline of the brain in red, overlayed on the whole image with skull.

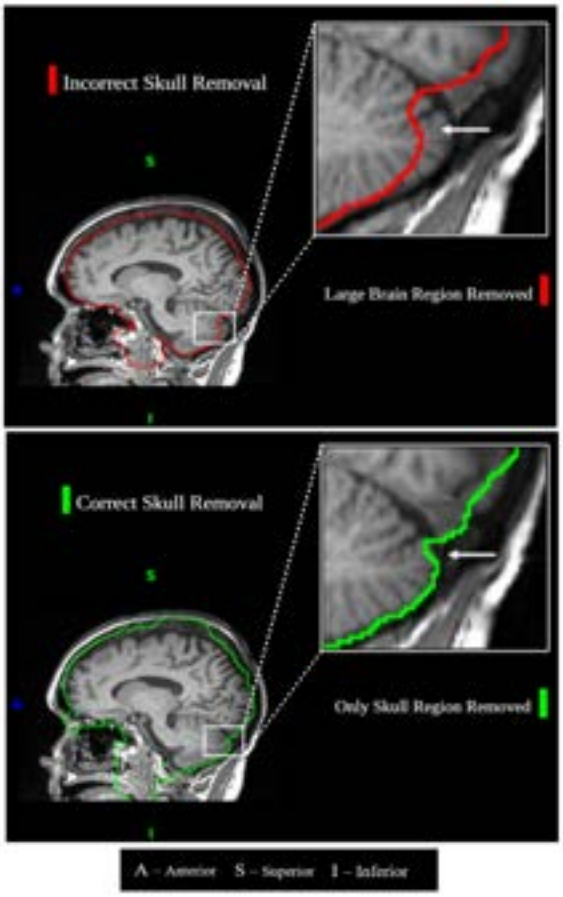

Figure 4.7: Sagittal brain MR slice of patient mMR_BR1_002 showing the incorrect way of removing skull (top scheme) and the correct way of removing skull (bottom scheme).

ii) Affine registration

After the skull-removal process was complete, it was necessary to define a common reference system to allow an anatomy comparison between the multiple subjects' scans. Image registration was therefore used to align all data to a common coordinate system - the MNI152 standard space [65].

The FLIRT algorithm [66] was used for affine registration with 12 degrees of freedom of all MR

Figure 4.8: Sagittal brain MR slice of patient mMR_BR1_002, illustrating an additional skull part that could not be removed with the BET tool without also removing important brain tissue.

scans to the MNI space with normalised mutual information cost function and tri-linear interpolation method. This computed affine transformation was then applied to the corresponding PET scan to equally align it to the MNI space. Finally, this process was repeated for every subject in the dataset to ensure the MR and PET scan alignment.

iii) Remasking

Once the image registration was complete, the images were remasked to further remove skull "left-overs" in the previous skull-stripped images and smooth the borders of the brain tissue to prevent sharp variations in contrast at the edges where the skull was eliminated. The removal of the skull ensured the networks focused on the brain tissue, where the lesions are located, thus not wasting memory or network parameters on the unnecessary parts of the images (skull and background).

For each patient it was necessary to have a personalised brain mask in order to remask the scans. The remasking was done by multiplying the mask with each subject's image, ensuring every piece of image outside the brain mask would be removed (pixel intensity turned to 0).

Each mask was created using the standard non-linear dilated brain mask available in FSL (represented as standard brain mask in Figure 4.9) and registering it back to the affine space for each subject (represented as brain mask affinely registered in Figure 4.9). To do this, all MR images were first non-linearly registered to the MNI space, using the FNIRT command in FSL. This operation outputs the non-linearly align MR scans as well as the warp fields of that operation, which can then be inverted to transform images from being non-linearly registered to affinely registered.

Therefore, the warp fields of each subject were inverted and applied to the standard non-linear brain mask of FSL, each creating an affinely registered brain mask of each subject. Each of these masks were then applied to the corresponding affine registered brain image (MR and PET) to further remove skull leftovers in the previous skull-stripped image. A schematic representation of the remasking process is presented below in Figure 4.9.

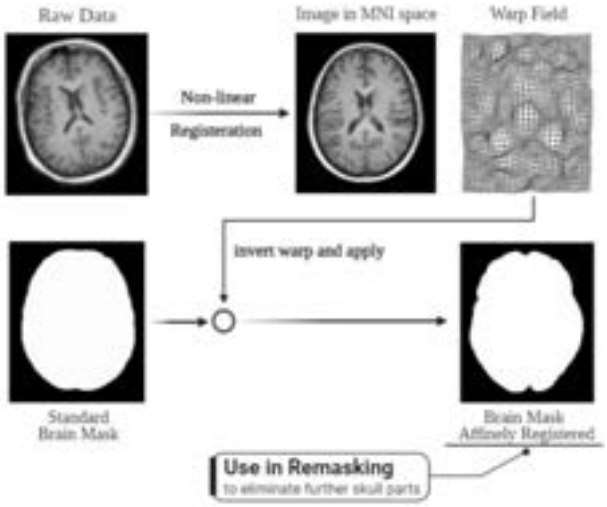

Figure 4.9: Scheme illustrating pipeline followed to obtain brain mask of patient mMR_BR1_002 and its transform back to the affine space. Warp field illustration retrieved from [67].

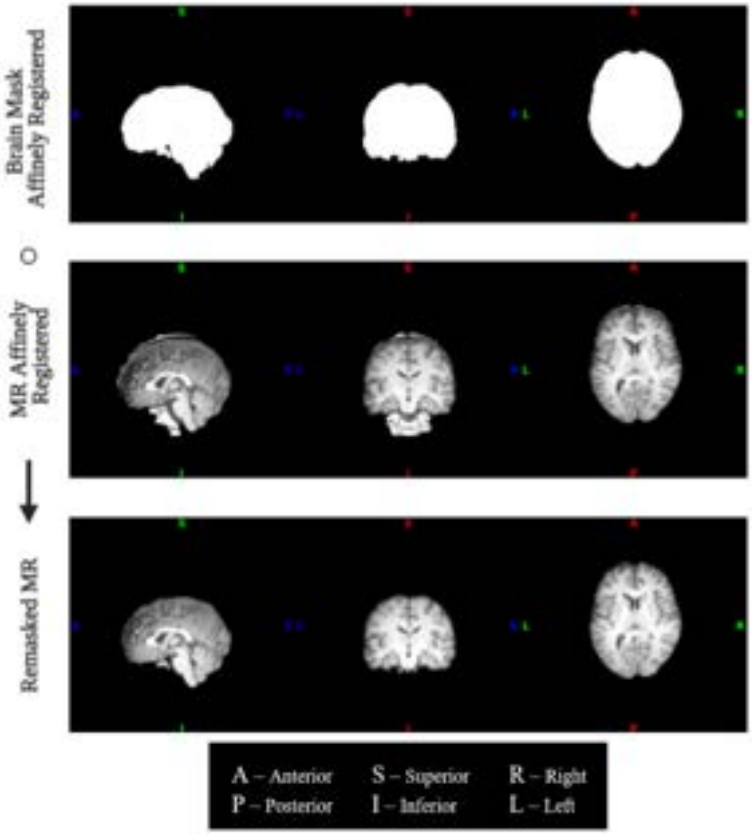

Figure 4.10: Top row represents the sagittal, coronal, and axial MR slices of brain mask of patient mMR_BR1_002 used for remasking the affine brain image for complete skull removal. Middle row represents the sagittal, coronal, and axial MR slices of the affinely registered brain image of patient mMR_BR1_002 before remasking. Bottom row represents the sagittal, coronal, and axial MR slices of the remasked brain image of patient mMR_BR1_002.

iv) Histogram Normalisation

The last step of the image pre-processing was the intensity normalisation of all the images.

Normalisation is a vital step before data analysis since it is not uncommon for images in the same dataset to have large intensity variations caused by the use of different image acquisitions parameters or even different scanners [68], which considerably affects the conclusions drawn from the image analysis.

In the specific case of FDG-PET images, the concentration of FDG in the brain was also found to be subject-dependent on the account of factors such as age, gender, or blood glucose level [62, 69]. Therefore, intensity normalisation plays an essential role, not only in MR images but also in PET images, to attenuate these variations so that it is possible to compare either voxel intensity values or ROI uptake values among patients [70].

Recently, the work of [68] has recommended the use of histogram-based normalisation methods in the harmonization of brain FDG-PET images compared to other intensity normalisation approaches. This same work [68] goes on to further illustrate that the use of inaccurate intensity normalisation methods in images can cause the wrong detection of disease-related hypometabolism regions, resulting in an increase of false positives during image analysis.

Having this in mind, the objective was to normalise the current dataset that contains several subjects to a common space. Therefore, a histogram normalisation method following the work of [71] was applied to this dataset to promote data-harmonisation among patients, rescaling image intensity distributions to match that of a standardised target distribution.

A general overview of the used histogram normalisation method is presented as follows:

- Firstly, a range of percentiles was chosen to map (10-90 range was chosen with a step of 10 between them).
- Then, the intensity values at each percentile were calculated for each of the images - commonly referred to as landmarks.
- Additional percentiles were chosen to act as the minimum and maximum of the range - 5 and 95 were chosen respectively, assuming the values below 5 and above 95 represent noise. The intensities corresponding to this range were then scaled to the target scale using interpolation.
- After this, the same scaling operation was applied to each landmark.
- Then, the landmark intensities were estimated for all images (in this case for every MR scan and then for every PET scan separately) and averaged to achieve a target set of average landmark intensities (one for each percentile in our original list).
- Finally, each image intensity was scaled to match the target (landmark) percentile scale.

As mentioned before, the intensity normalisation in this project was performed for the MR scans and PET scans separately for every patient. Figure 4.11 represents an example of the MR and PET scan of a patient before and after normalisation, where the intensity values can be visually perceived as different, specifically in some regions.

4.2.3 Lesion Masks

For each patient's data, a binary mask was created, each reflecting the brain regions where the lesions were found by the clinical labels available.

For this task, the Hammer Atlas n30r83 maximum probability map [10, 72–74] with 83 segmented brain regions in MNI space - illustrated in Figure 4.12 - was used to identify the regions accordingly to the clinical labels and a binary mask of those was created. The patient's non-linearly registered data to MNI space was used to check if the masks overlayed correctly with the brain regions they belonged to

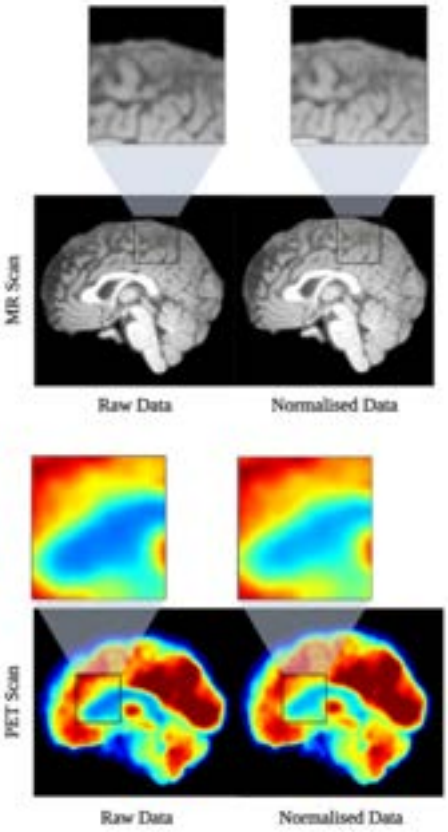

Figure 4.11: Sagittal MR scan slice (top scheme) and PET scan (bottom scheme) before and after normalisation, with emphasis on a brain region where it is visually possible to understand the difference in intensity between both images. Both MR and PET scans belong to patient mMR_BR1_002.

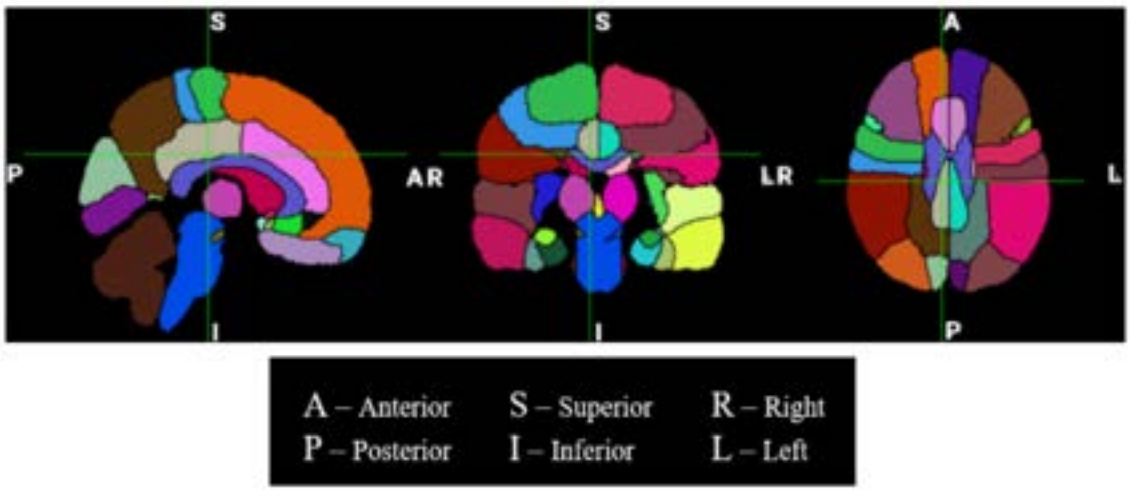

Figure 4.12: Sagittal, coronal, and axial slices of the Hammer atlas n30r83 maximum probability map visualised in FSLeyes image viewer tool [75]. Each colour represents a different segmented region of the atlas.

and were then inverted back to the affine MNI space (with the affine transforms used to register the data in an affine way to MNI space in section 4.2.2).

In this atlas, each segmented brain region had a different intensity value, with a numerical label attached to it. The specific regions were therefore selected by using the intensity tool in FSL, which allows the selection of voxels by their specific intensity.

Once the brain regions were isolated for each patient - creating a brain mask only with the lesion

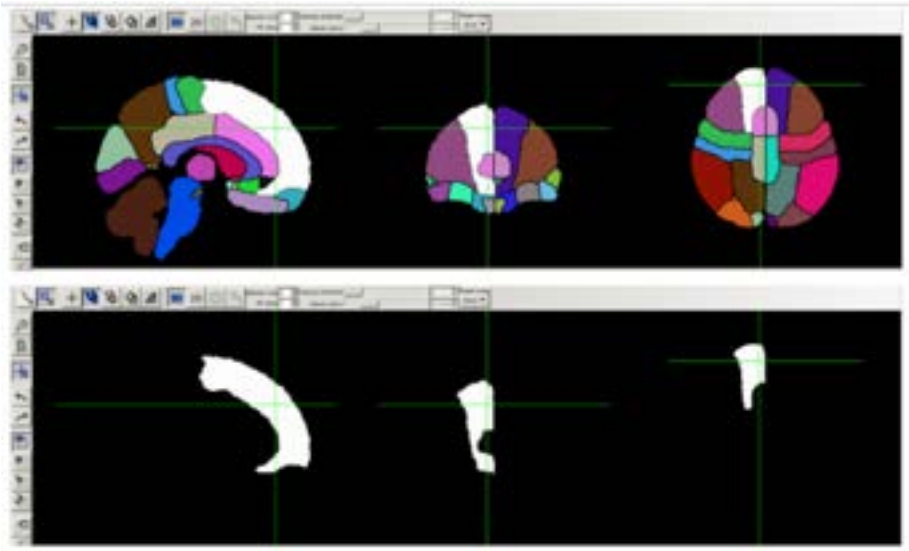

Figure 4.13: Example of how the brain regions were selected for each patient using the intensity tool of FSLeyes (top image). The selection of the right-side superior frontal gyrus (belonging to the frontal lobe) in the atlas and the isolation of that region - with now an intensity value equal to 1 (bottom image).

regions - the mask was converted back to the affine space and overlayed with the affinely registered patient's scans to visually confirm that the lesion mask was in the correct anatomical region of the brain. Figure 4.14 illustrates the overlay of the masks for different patients, confirming that the anatomical region and the atlas label correspond (are aligned) on the MR scan. Figure 4.15 shows the PET scan of a patient with a clearly visible lesion (on the crosshair location - region in blue on the temporal lobe) and the mask overlayed on top of it, confirming that the mask is aligned with the lesion area.

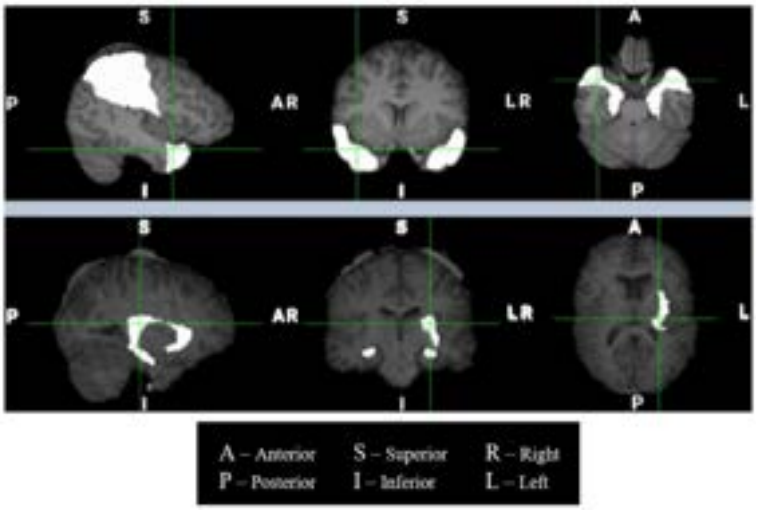

Figure 4.14: Example of lesion masks for different patients overlayed with the corresponding MR scans. Top row - sagittal, coronal, and axial slices of MR scan of patient mMR_BR1_021, with the lesion mask located on both temporal poles and the right parietal lobe. Bottom row - sagittal, coronal, and axial slices of MR scan of patient mMR_BR1_062, with the lesion mask located on the right and left hippocampus and left insula.

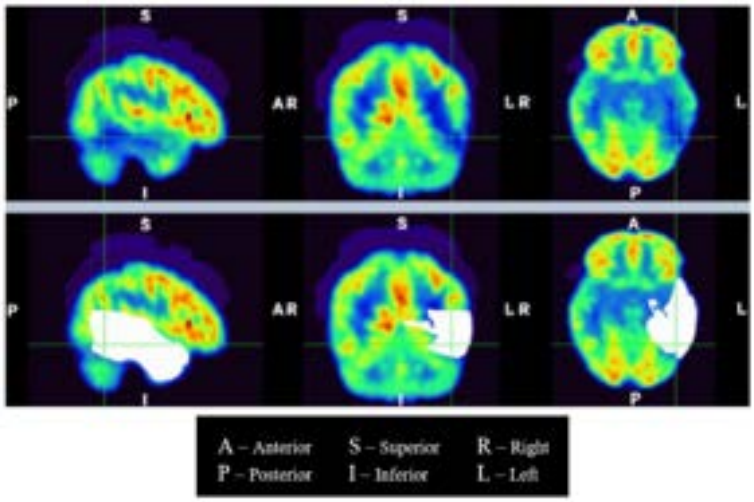

Figure 4.15: Example of PET scan (top row) and the lesion mask overlayed (bottom row). Top row - sagittal, coronal, and axial slices of PET scan of patient mMR_BR1_020. The area of the lesion can be seen where the crosshair is positioned (darker blue area on the left temporal lobe). Bottom row - sagittal, coronal, and axial slices of the PET scan of the same patient with the overlayed lesion mask, showing the lesion location and the brain mask area overlap, as expected.

4.3 Experimental Set-Up

4.3.1 Anomaly Detection Methods

In this chapter, two different anomaly detection approaches (reconstruction and translation) were implemented using two different networks (WGAN and CycleGAN), taking advantage of both PET and MR modalities present in the dataset.

i) Detection through Reconstruction

The first implemented approach involved training a 3D WGAN with only healthy patches from the patients' PET-MR scans (thus leaving out any diseased tissue), for the network to learn how to reconstruct healthy PET-MR patches. In testing, the whole PET-MR scans (with healthy and diseased tissue) were passed through the network and the diseased areas were expected to show higher reconstructed errors, therefore identifying lesion locations.

ii) Detection through Translation

This approach for anomaly detection made use of a 3D CycleGAN to perform translation between diseased and healthy patches of patients. The network was therefore trained to translate between these 2 classes (healthy and diseased), and, during testing, the whole PET-MR image was translated to a healthy version of it in which the lesions were removed, enabling to identify their location.

Therefore, this section aims to implement these two different detection approaches by combining advanced machine learning techniques to create a diagnostic tool for anomaly detection, specifically epilepsy-causing brain lesions, with a particular focus on FCD, from PET-MR data.

4.3.2 Networks Architecture and Training Details

Depending on the implemented network - WGAN or CycleGAN - different architectures, loss functions, and hyperparameters were chosen. In general, the typical loss functions for GANs were

used in both networks, apart from a personalised loss inspired by the work of [14], which was added to the CycleGAN training and will be described in this section. Therefore, both networks used different training methodologies that are enumerated and described next.

i) WGAN

The objective of the WGAN was to identify lesions through reconstruction. In this approach, training was performed using only healthy patches, with the goal of learning a generative model of healthy brain appearance. During testing, whole images (including the diseased patches where the lesion was present) are passed through the network, with the expectation that a higher reconstruction error will appear for lesion areas, since the network did not learn the diseased distribution during training.

The WGAN network (Figure 4.16) comprises a Generator and a Critic. The Generator receives ground truth healthy patches of the MR and PET and learns how to reconstruct them. The critic receives the original healthy patches and their reconstructions, with associated labels (label 0 for reconstructions and label 1 for original patches), and then gives a score to the input, predicting if it represents an original or reconstructed patch.

In terms of network losses, the WGAN was trained using a Wasserstein distance loss with gradient penalty, and a supervised L1 reconstruction loss (to penalise differences between the original and reconstructed patches). The Wasserstein loss was therefore used to measure the distance between the ground-truth data distribution and the distribution shown in reconstructed samples [27], aiming to optimise the Critic and the Generator. Additionally, a L1 loss was imposed to penalise reconstruction errors between the original and the reconstructed patch.

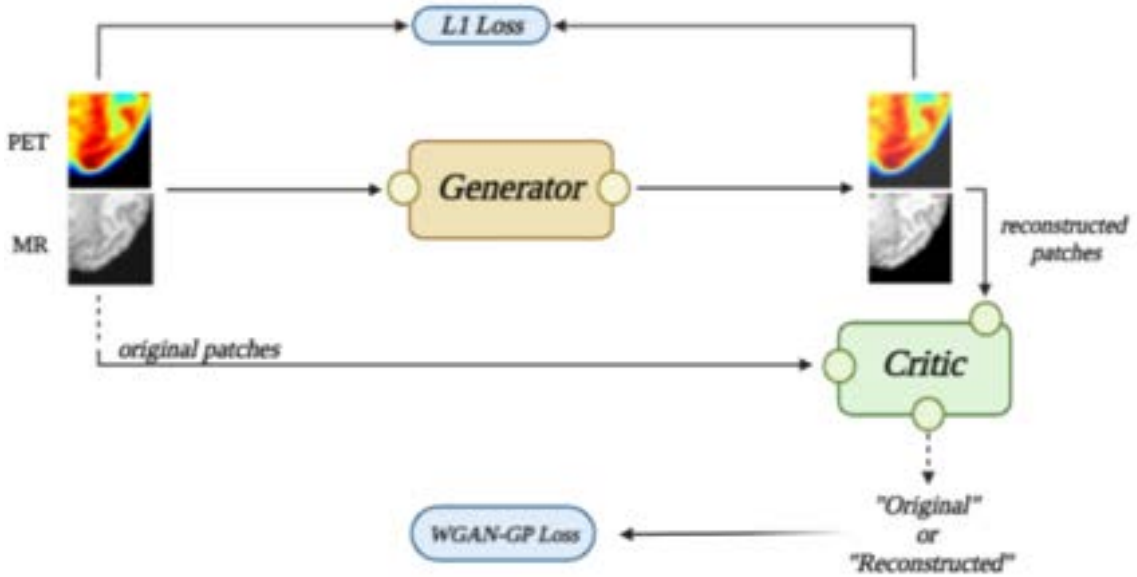

Figure 4.16: Illustration of the WGAN structure (one Generator and one Critic) and data flow with associated losses. The network input is represented by the original healthy MR and PET patches, which pass through the Generator and are reconstructed. The Critic aims to classify the patches as original or reconstructed using a WGAN-GP loss that optimises both the Generator and Critic during the training. The L1 loss between the original and reconstructed patches is used to also optimise the Generator.

The architectures for the Generator and Critic are illustrated in Figures 4.17 and 4.18, respectively. Tables A.8 and A.9 in the appendix describe the parameters used for the generator and critic, respectively. The hyperparameters used to train the WGAN were based on the recommended values from the work of [21] and are presented in Table A.10 in the appendix.

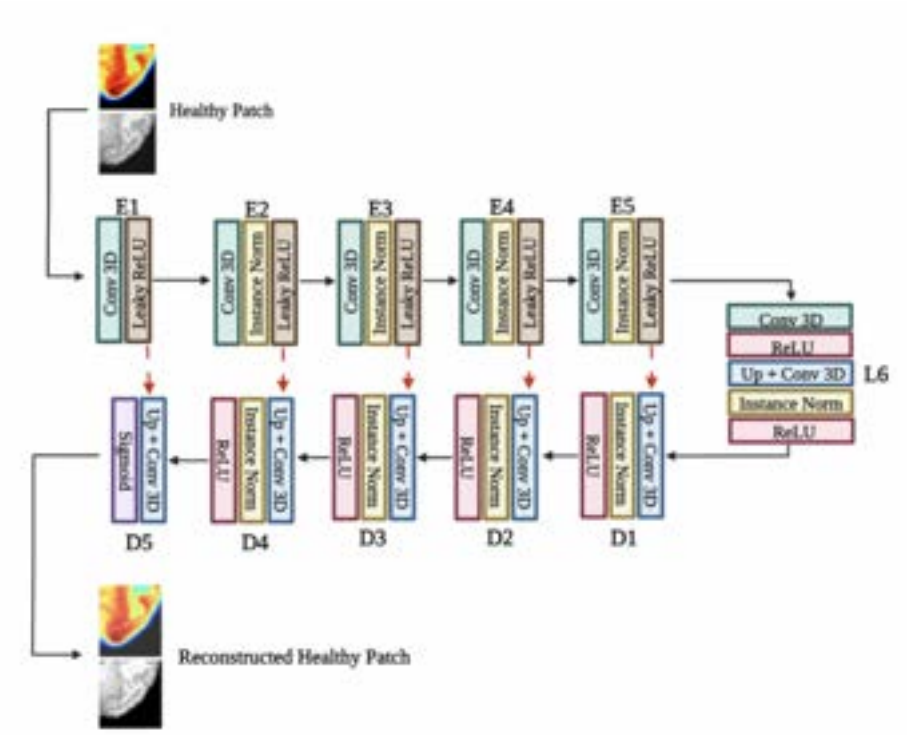

Figure 4.17: Illustration of the Generator's architecture. The Generator is based on a U-net architecture with skip connections (represented by the red arrows). This architecture included an Instance normalisation layer and a Leaky ReLU activation function (with a negative slope parameter set to 0.2), as well as a ReLu activation functions for the decoder layers. The input of the Generator was the healthy patches of size 2x64x64x64 (the 2 channels referring to both the MR and PET scans of the associated patient) and its output consisted in the reconstructed input patches with a sigmoid function as a last layer. The Generator aims to learn the mapping of the healthy patches.

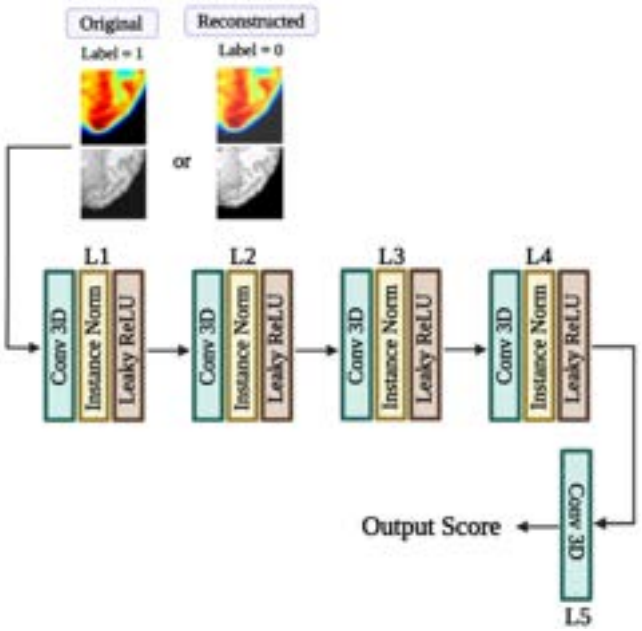

Figure 4.18: Illustration of the Critic's architecture, represents a typical down sampling network using convolution, with a distinguishing factor of not having a final sigmoid layer. This architecture included an Instance normalisation layer and a Leaky ReLU activation function (with a negative slope parameter set to 0.2). The input of the Critic was the healthy or reconstructed patches of size 2x64x64x64 (the 2 channels refer to both the MR and PET scans of each patient) and its associated labels (whether 0 or 1). The Critic output was a score given to the input image to classify it in either more probable to be an original patch or its reconstruction. The Critic aims to distinguish the original healthy patches from their own reconstruction.

ii) CycleGAN

When it comes to the CycleGAN, it uses a translation methodology where the network is trained to learn the mapping between healthy to diseased patches, and vice-versa. The typical CycleGAN uses 2 Generators and 2 Discriminators that aim to learn this mapping.

The CycleGAN was optimised using the following losses:

- Discriminator D_A loss - implements a binary cross entropy (BCE) function which outputs a predicted probability value (between 0 and 1) with 1 corresponding to the original abnormal patches, and 0 corresponding to the translated healthy patches (Figure 4.19.b with an associated label = 0).
- Discriminator D_N loss - implements a BCE function which outputs a predicted probability value (between 0 and 1) of the patches corresponding to original normal patches (Figure 4.19.a with an associated label = 1) or normal patches translated from diseased patches (Figure 4.19.e with an associated label = 0), furthermore calculating a score that penalizes the probabilities based on their distance to the expected label.
- Cycle-Consistency A2N loss - implements a L1 distance loss function between the original abnormal patches (d. in Figure 4.19) and the reconstructed abnormal patches after being translated to normal patches and translated back to abnormal patches (Figure 4.19.f).
- Cycle-Consistency N2A loss - implements a L1 distance loss function between the original normal patches (Figure 4.19.a) and the reconstructed normal patches after being translated to abnormal patches and translated back to normal patches (Figure 4.19.c).
- Anomaly mask loss - implements an L2 distance loss between the original abnormal patches (Figure 4.19.d) and its translation to normal patches (Figure 4.19.e), both multiplied by the lesion mask of the abnormal patch. Therefore, only the healthy tissue is preserved, and its reconstruction is evaluated, since the lesion regions indicated in the anomaly mask are set to a voxel intensity of 0. Equation 4.1 represents this personalised loss function.
- Identity A loss - implements a L1 distance loss function between abnormal patches (Figure 4.20.g) fed into the Generator N2A and its output, which should ideally correspond to the same abnormal patches (Figure 4.20.h). This loss is applied to further ensure that the Generator N2A will not translate abnormal patches since they already belong to the domain it should output.
- Identity N loss - implements a L1 distance loss function between the normal patches (Figure 4.20.i) fed into the Generator A2N and its output, which should ideally correspond to the same normal patches (Figure 4.20.j). This loss is applied to further ensure that the Generator A2N will not translate normal patches since they already belong to the domain it should output.

The identity losses are represented in Figure 4.20 and the remaining losses illustrated in Figure 4.19. Tables A.11 and A.12 in the appendix also describe the parameters used in the network's layers. The hyperparameters used to train the CycleGAN were based on the recommended values in [23] and are presented in Table A.13. The λ_{AM} parameter for the anomaly mask loss was used according to the work of [14].

Figure 4.19: Illustration of the CycleGAN structure and data flow with associated losses. Healthy MR and PET patches pass through the CycleGAN with the goal of learning the mapping that allows to translate between healthy and diseased patches and vice-versa. The overall structure of the CycleGAN is composed by 2 Generators (Generator A2N and N2A) and 2 Discriminators (D_A and D_N). The Generator N2A is trained to translate "normal" healthy patches (a.) to "abnormal" diseased patches (b.) and the Generator A2N is trained to translate diseased patches (d.) to healthy patches (e.) - their associated loss includes L1 losses (cycle-consistency losses) between the original (a. and d.) and reconstructed patches (c. and f.). The Discriminator D_A is trained to distinguish between these real abnormal patches (d.) and abnormal patches translated from healthy patches (output of Generator N2A – b.). The Discriminator D_N was in turned trained to distinguish between real healthy patches (a.) and healthy patches translated from diseased patches (output of Generator A2N - e.). The Discriminators losses are represented by a binary cross entropy (BCE) loss. Finally, an anomaly mask loss was added between the input of the Generator A2N (d.) and its output (e.), both multiplied by the binary lesion mask of the associated patient.

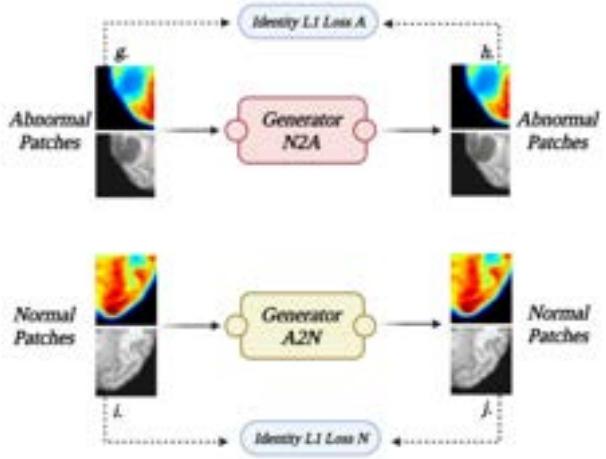

Figure 4.20: Representation of the Identity L1 losses of the CycleGAN. Identity loss A was applied between abnormal patches (g) fed into the Generator N2A and its output (h). Identity loss N was applied between the normal patches (i) fed into the Generator A2N and its output (j).

The goal of the anomaly mask loss was to ensure the healthy part of the tissue present in the diseased patch was not being modified, since the input patch was not entirely diseased. Therefore, by multiplying the lesion mask by the input and output patches of the Generator A2N, the healthy tissue remained in both images and the voxels of the diseased tissue were set to an intensity of 0. If any voxel in the healthy

tissue area was translated when passed through the network, the Generator A2N was penalised since its aim was to only translate the diseased tissue to healthy, keeping the healthy tissue of the input patch unchanged.

Equation 4.1 describes this loss function, where M represents the matrix of the binary lesion mask where the lesion location has a voxel intensity equal to 0. Both the input abnormal patch and the output patch of Generator A2N had 2 channels (corresponding to the PET and MR scans) and M had only 1 channel. Both M and the patches had the same width and height. This loss therefore measures an L2 distance between the heathy tissues in both patches, where the lesion location in the images is set to an intensity of 0 – illustrated in Figure 4.21.

It was assumed that the patches x were drawn from their corresponding distributions: $x^a \sim p_a$ and $x^n \sim p_n$, being x^a a sample from the abnormal patch distribution and x^n a sample from the normal patch distribution. The penalty $\mathscr{L}_{AM}$ can therefore be defined as:

$$\mathscr{L}_{AM} = E_{\ p_a(x)}[\|M \odot (G_{A2N}(x^a) - x^a)\|_2^2] \tag{4.1}$$

where $\odot$ represents element-wise multiplication, $\mathrm{E}_{\ (\mathrm{p}_a(x))}$ the estimator of the abnormal distribution dependent on a patch x , M the binary lesion mask, and $G_{A2N}(x^a)$ the resulting patch from the Generator A2N, that receives an abnormal patch, x^a, as its input.

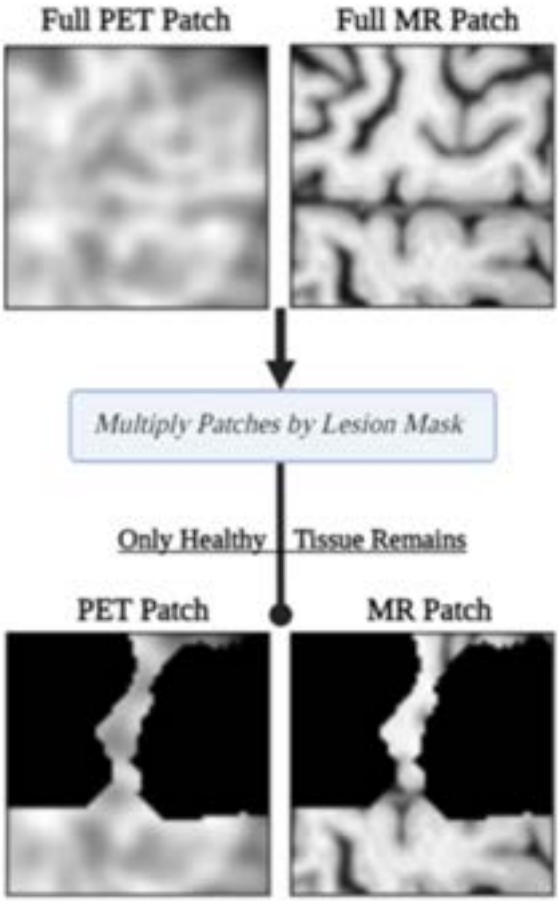

Figure 4.21: Representation of the patches evaluated by the anomaly mask loss. The full patches (corresponding to the MR and PET channels) are multiplied by the corresponding lesion mask that have the lesion regions with voxel intensity equal to 0. Consequently, the resulting patches only have healthy tissue present (with the regions that are inside the anomaly mask set to 0). The MSE loss is evaluated in this way, between the input and output patches of the Generator A2N.

The final Generator loss can therefore be described as the summation of the generator losses in Equation 4.2:

$$\mathscr{G}_{loss} = \mathscr{L}_{G_{A2N}} + \mathscr{L}_{G_{N2A}} + \lambda_{L1}\,\mathscr{L}_{Cycle_A} + \lambda_{L1}\,\mathscr{L}_{Cycle_N} + \lambda_{id}\,\lambda_{L1}\mathscr{L}_{Identity_A} + \lambda_{id}\lambda_{L1}\mathscr{L}_{Identity_N} + \lambda_{AM}\mathscr{L}_{AM} \tag{4.2}$$

where $\mathscr{L}_{G_{A2N}}$ and $\mathscr{L}_{G_{N2A}}$ represents the loss of Generator A2N and Generator N2A that resulted from the respective Discriminators, $\mathscr{L}_{Cycle_A}$ and $\mathscr{L}_{Cycle_N}$ represent the cycle-consistency losses, $\mathscr{L}_{Identity_A}$ and $\mathscr{L}_{Identity_N}$ the identity losses, and $\mathscr{L}_{AM}$ the anomaly mask loss. The different generator losses were multiplied by hyperparameters (λ_{AM} , λ_{id}, λ_{L1}) ollowing the work from [23] and [14], and their values are presented in Table A.13.

Both Discriminators D_A and D_N shared the same architecture (illustrated in Figure 4.22 and Figure 4.23, respectively), and both Generators A2N and N2A shared the same U-net structure, as presented in Figure 4.24.

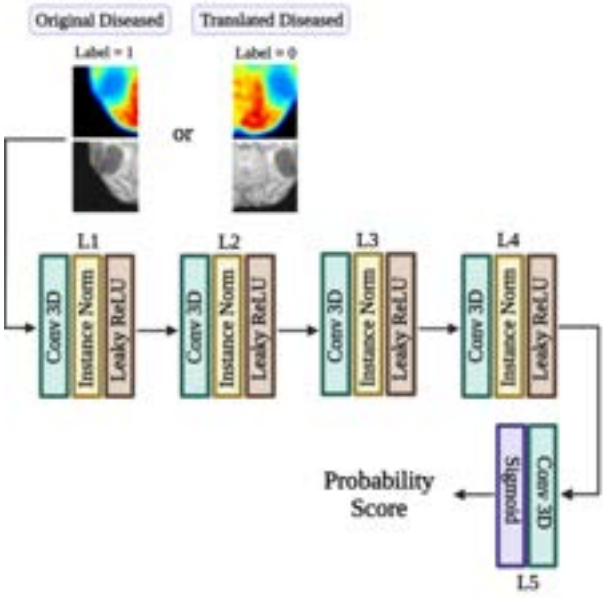

Figure 4.22: Illustration of the Discriminator D_A architecture. This architecture included an Instance normalisation layer and a Leaky ReLU activation function (with a negative slope parameter set to 0.2). The input of the Discriminator was the original diseased patches or the translated-to-diseased patches of size 2x64x64x64 (the 2 channels refer to both the MR and PET scans of the associated patient) and their associated labels (0 or 1). The Discriminator output is a probability score given to the input image depending on whether it was an original patch or a translation-to-diseased patch. The Discriminator aims to distinguish the original diseased patches from translations-to-diseased patches.

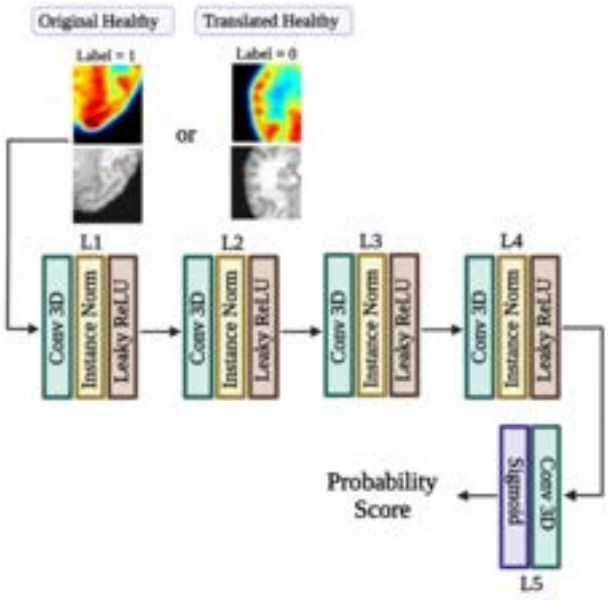

Figure 4.23: Illustration of the Discriminator D_N architecture. This architecture included an Instance normalisation layer and a Leaky ReLU activation function (with a negative slope parameter set to 0.2). The input of the Discriminator was the original healthy patches or the translated-to-healthy patches of size 2x64x64x64 (the 2 channels refer to both the MR and PET scans of the associated patient) and their associated labels (0 or 1). The Discriminator output was a probability score given to the input image depending on whether it was an original patch or a translation-to-healthy patch. The Discriminator aims to distinguish the original healthy patches from translation-to-healthy patches.

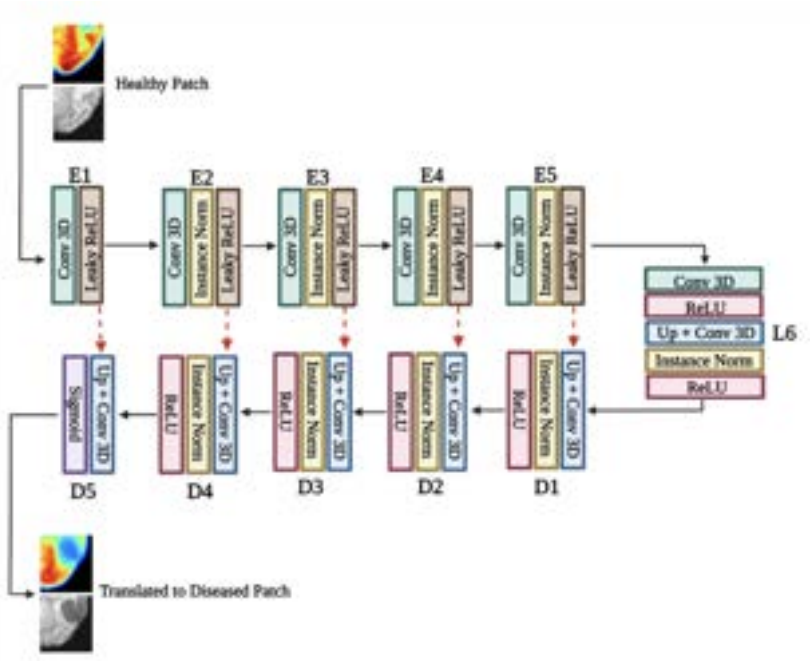

Figure 4.24: Illustration of the Generator N2A architecture (it shares the same architecture as Generator A2N but has the diseased patches as input and their translations to healthy as output). The Generators were based on a U-net architecture with skip connections (represented by the red arrows). This architecture included an instance normalisation layer and a Leaky ReLU activation function (with a negative slope parameter set to 0.2) in the encoder layers and a ReLU activation function in the decoder layers. The input of Generator N2A were the healthy patches of size 2x64x64x64 (the 2 channels refer to both the MR and PET scans of the associated patient) and its output consisted in the translation-to-diseased from the input patches, with a sigmoid function as a last layer. Generator N2A here illustrated aims to translate healthy patches to diseased patches and Generator A2N aims to translate diseased patches to healthy ones.

4.3.3 Training Methodology

i) Patch-based Training

As motivated by Chapter 3, all networks implemented for this project were trained with 3D patches instead of using the entire 3D scans in order to address the high computational cost of training large 3D images, and reduced quantity of available data. Additionally, patched-based training of the networks allows for the use of the patients' own data as healthy tissue, since the lesions are not located in the entire brain volume.

Patch selection was implemented using a 'weighted cropper' function [76], which took as input a weight map – corresponding to the lesion mask created for each image in section 4.2.3 – and cropped random regions (either inside or outside the lesion mask) depending on if a healthy patch or a diseased patch was desired. Figure 4.25 illustrates an example of a 2D axial slice of the binary weight maps used to sample random patches of either healthy or diseased patches. The regions where voxel intensity equals to 0 (represented in black) mean that the centre of the patch will not be randomly chosen in that region. In contrast, white regions (voxel intensity equals to 1) indicate that the weighted cropper function is allowed to randomly sample patches in that region.

The optimal patch size was an important factor to consider. Ideally, patch size should be optimised to be sufficiently small to save computational memory, but big enough to learn the global context of the image. In the specific case of this work, there was also the need to have a patch size that was big enough to sample most of each masked lesion area (with roughly the same order of size of a brain lobe) but, at the same time, small enough to not sample too much healthy tissue. Having this is mind, patch sizes of 128 x 128 x 128 were, considered inappropriate upon visual inspection since their size was too big considering the size of the lesions, whereas a patch size of 32 x 32 x 32 was considered too small. As a result, an intermediate patch size of 64 x 64 x 64 was chosen, and networks were trained on 3D patches randomly sampled from each whole 3D image in the training set.

Patch selection was implemented differently for each network architecture: for the WGAN, the network was trained only with patches that only contained healthy tissue. As such the random sampling of patches was parameterised to ensure that the patches had a maximum of 10% of lesion area. Therefore, patches were considered healthy if they did not have more than 10% of their volume belonging to a region marked as diseased.

For the CycleGAN training, it was needed to have both healthy and diseased classes. Therefore, while the same modified weighted cropper function (used in the WGAN) was applied to sample random healthy patches; for the diseased class, the weighted cropper function was modified to only sample patches that had at least 10% of lesion volume in the total patch.

Figure 4.26 shows an example of the random sampling of a healthy patch of size 64 x 64 x 64 voxels that also contains a portion of diseased tissue but is considered as healthy since the percentage of lesion area is less than 10% of the total volume of the patch. Figure 4.27 shows the same sampled patch of Figure 4.26, with the associated 2D and 3D visualisation of the patch in relation to the entire MR scan.

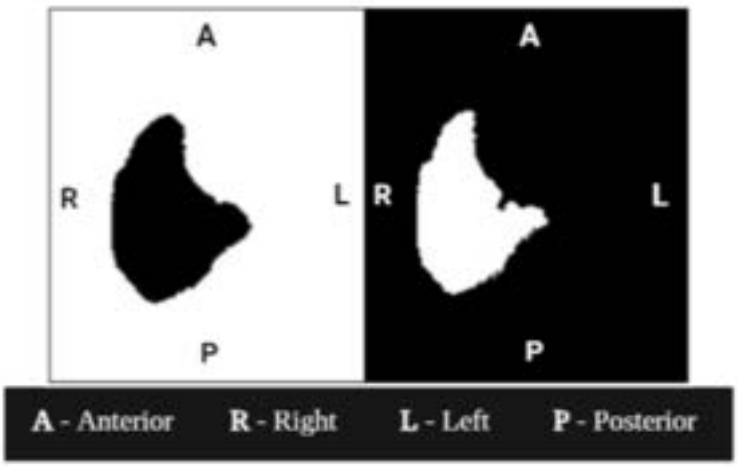

Figure 4.25: Example of axial 2D slice binary lesion mask (left temporal lobe) of patient mMR_BR1_020. The left-side figure represents the weight map used to sample healthy patches whereas the right-side figure represents the weight map used to sample diseased patches. The regions in black represent voxels with intensity equal to 0 and regions in white with intensity equal to 1.

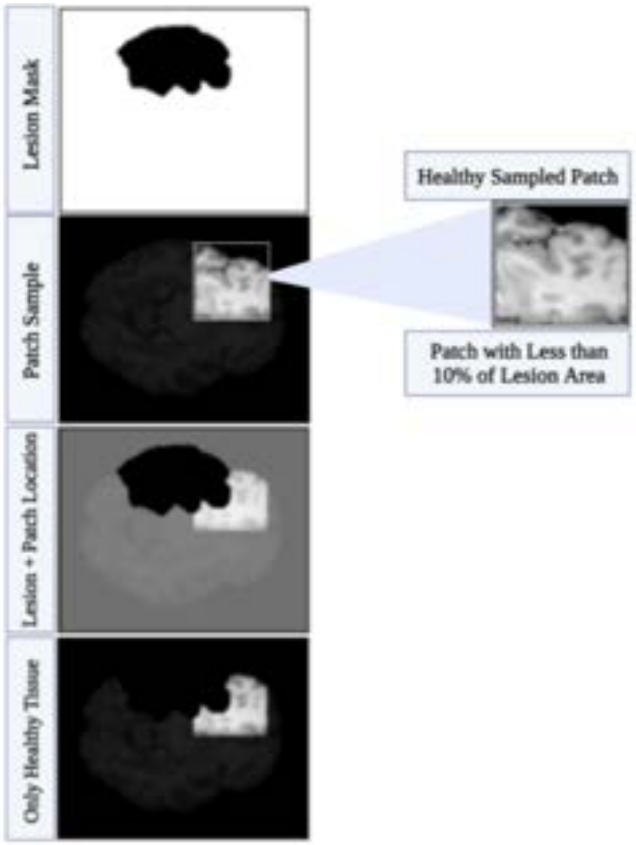

Figure 4.26: Illustration of axial slices showing a random healthy patch sampled from the MR scan of patient mMR_BR1_047, and the quantity of diseased tissue it contains – a lesion area needs to be less than 10% of the total area to be considered a healthy patch. The top image represents an axial slice of the binary lesion mask (where the observed diseased region belongs to the right temporal lobe) - this binary mask is used by the healthy patch sampler function to know in which regions it can sample patches. The second from the top image highlights the random sampled patch – this specific patch was sampled close to the lesion mask but contains less than 10% of lesion area and is, therefore, considered as a healthy patch. The third image illustrates the overlay of the lesion mask and the highlighted sampled patch, showing that the patch contains a portion of diseased tissue. The bottom image represents only the healthy tissue in the patch, with the intensity of the diseased tissue voxels set to 0 for better visualisation.

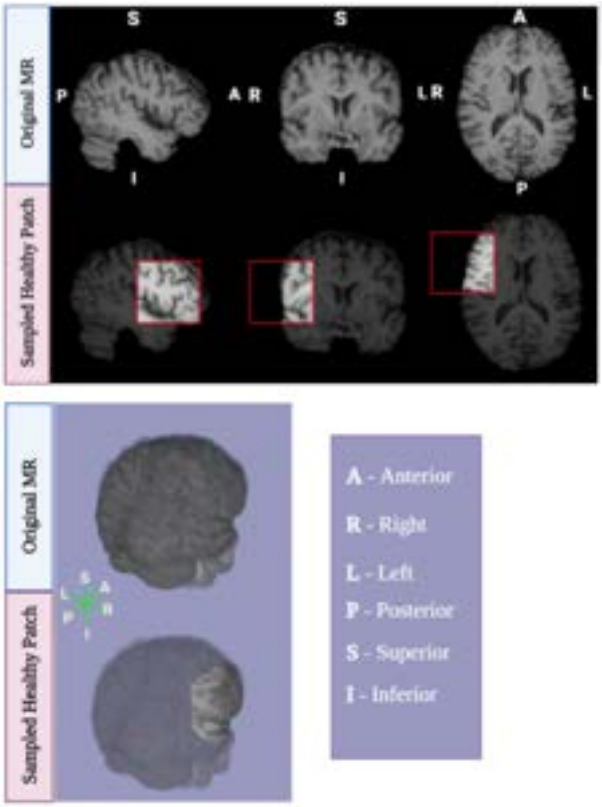

Figure 4.27: 2D and 3D visualisation of the same patch sampled in Figure 4.26 in relation to the entire MR scan of patient mMR_BR1_047.

ii) Multimodal Channel versus Single Channel Training

Additionally, a comparison was made between training with both PET and MR modalities and with a single channel (either MR or PET modality individually). Consequently, each patch for the multimodal training had a shape of 1 x 2 x 64 x 64 x 64 – where 1 corresponds to the batch-size (only one patch in this case), 2 to the multimodal channels (PET and MR) and 64 x 64 x 64 to the patch dimension. Each patch for the single channel training had a shape of 1 x 1 x 64 x 64 x 64 – where the single channel would now only be either PET or a MR modality.

As a result, the WGAN and CycleGAN networks were trained with the following data inputs:

- WGAN with multimodal channel (PET and MR data)
- WGAN with single channel (PET data)
- WGAN with single channel (MR data)
- CycleGAN with multimodal channel (PET and MR data)
- CycleGAN with single channel (PET data)
- CycleGAN with single channel (MR data)

This comparison was motivated by several papers [60, 77, 78] that note the many challenges when training with multimodal data, especially when imaging data is obtained from different scanners or research centres. Some of these challenges and future ways to tackle them will be discussed in chapter 5.

iii) Data Augmentation

In addition to randomised patch selection, random left-right (horizontal) patch flipping [79] was also introduced in training as a data augmentation strategy. The random flips had the intention of allowing the network to learn how both hemispheres of a healthy brain should look like. In the WGAN, for example, this would allow that a patient that has the right temporal lobe diseased (which makes this region not to be passed through the network in the training) could have a representation of that left side (imitating what a healthy right temporal lobe should look like) if the patch is flipped.

4.3.4 Testing Methodology

i) Patient Selection

For testing, both PET and MR scans of the patient needed to have a well-defined lesion location to analyse the network performance in detecting the abnormality in that same region. Having this is mind, patients (mMR_BR1_050 and mMR_BR1_020) with lesions of the category MR+PET+ were chosen. The lesions of both patients are represented in Figures 4.2 and 4.4, respectively.

Patient mMR_BR1_020 had a very visible FCD on both PET and MR scans, which resembles the lesions present on the MR data used in the work of [14], replicated in Figure 2.4(a). Since the networks tested in this chapter take inspiration from the work of [14], it was desired to replicate the performance of this paper with the networks built in this chapter, evaluating if they could equally identify the noticeable anomaly. Conversely, patient mMR_BR1_050 was chosen since it had a subtle FCD on both PET and MR scans, which aimed to test the networks' ability to detect these more subtle lesions, more difficult to identify.

ii) Whole-image Testing

All networks in this chapter used patched-based training (described in section 4.3.3.1) but were tested using the entire 3D images to identify lesions in the whole brain volume, and not exclusively on patches.

During inference, the whole 3D image of the test data is passed through the model using the sliding window function described in section 3.3.2 and its process is illustrated in Figure 3.10. This enables the model to receive the patches that form the entire image, output their reconstruction (for the WGAN) or translation (for the CycleGAN), and construct recursively the entire 3D image again with these predictions. It is this 3D whole-image that results from all the patch predictions of the networks that will be used to evaluate the networks performance in detecting the lesions, through difference maps calculate between the input and output images of the networks, for the two models: WGAN and CycleGAN.

iii) Difference Maps

For both detection experiments, the detection of lesions was completed by using absolute difference maps, which indicated the regions in the images with more differences than the potential lesions.

Therefore, for the WGAN model at test time, a difference map was computed between the input of the model and its output (reconstruction). The map should highlight the regions which have intensity distributions the WGAN has not learned how to reconstruct (diseased tissue) – therefore indicating lesions in said area. Higher reconstruction error indicates the most probable lesion location.

In the CycleGAN experiment, at test time, the difference map is calculated between the input image (considered diseased) and its output (a healthy translation). The model is expected to only translate to healthy the regions it has identified as diseased, leaving the already healthy tissue untouched. This will result in higher differences in the diseased regions of the image, corresponding to the lesion location.

4.4 Results

All networks in this section were trained until both the training and validation losses decreased until reaching a point of stabilisation. As a result, the WGAN (either multimodal or single-channel network) trained for 10 000 epochs (approximately 5 days) and the CycleGAN (either multimodal or single-channel network) for 15 000 epochs (approximately 7 days). Therefore, the results presented for the WGAN and CycleGAN correspond to the tests performed with the test patients using the fully trained networks.

4.4.1 WGAN

i) Multimodal Data

Figure 4.28 corresponds to the original, reconstructed and difference map of patient mMR_BR1_020 obtained for the WGAN, using multimodal data for training. These figures show both MRI (first set of images) and PET (second set of images) channels. The axial, sagittal and coronal slices illustrated in the figures correspond to the same region, where the lesion was located.

Figure 4.30 corresponds to the original, reconstructed and difference map of patient mMR_BR1_050 obtained for the WGAN, using multimodal data for training. These figures show both MRI (first set of images) and PET (second set of images) channels. The axial, sagittal and coronal slices illustrated in the figures correspond to the same region, where the lesion was located.

ii) PET Data

Figure 4.31 corresponds to the original, reconstructed and difference map of patient mMR_BR1_020 obtained for the WGAN, using a single image modality – PET. The axial, sagittal and coronal slices illustrated in the figures correspond to the same region, where the lesion was located.

Figure 4.33 correspond to the original, reconstructed and difference map of patient mMR_BR1_050 obtained for the WGAN, using a single image modality – PET. The axial, sagittal and coronal slices illustrated in the figures correspond to the same region, where the lesion location.

iii) MR data

Figures 4.34, correspond to the original, reconstructed and difference map of patient mMR_BR1_020 obtained for the WGAN, using a single image modality – MRI. The axial, sagittal and coronal slices illustrated in the figures correspond to the same region, where is possible to visualise the lesion location.

Figures 4.36 correspond to the original, reconstructed and difference map of patient mMR_BR1_050 obtained for the WGAN, using a single image modality – MR. The axial, sagittal and coronal slices illustrated in the figures correspond to the same region, where the lesion is located.

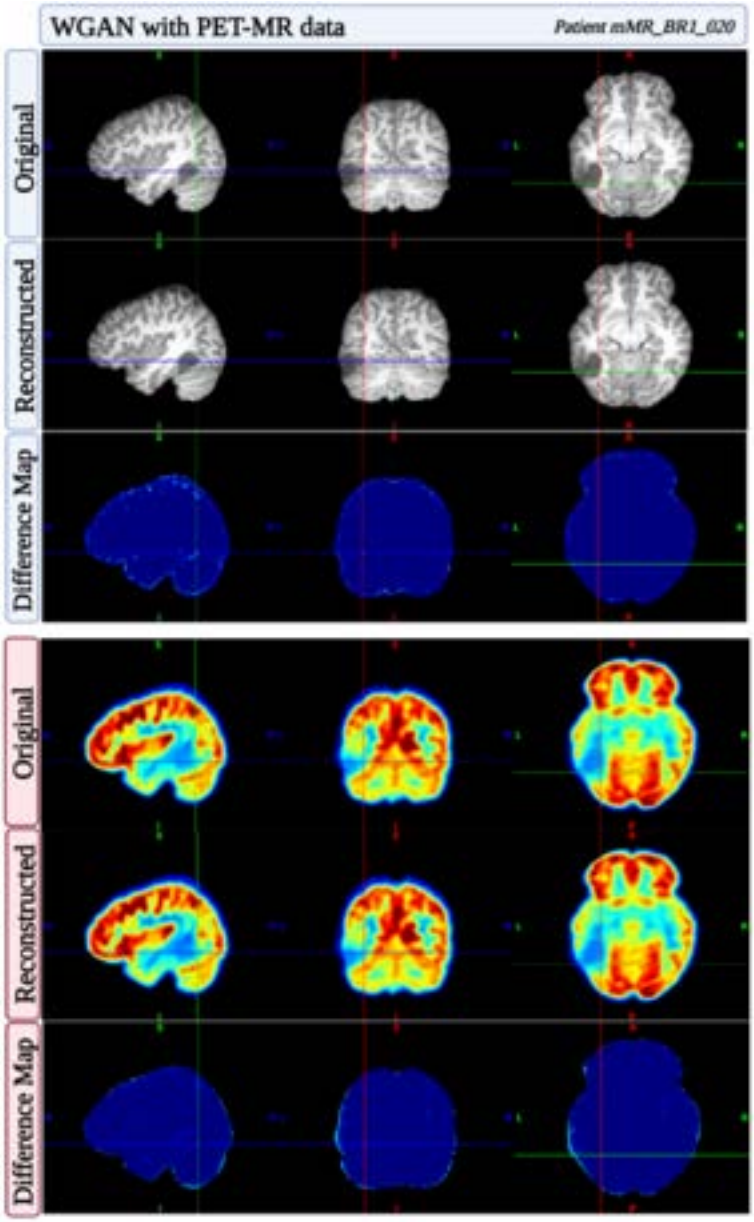

Figure 4.28: Original, Reconstructed and Difference maps images of both MRI (first set of three images) and PET channels (second set of three images), for patient mMR_BR1_020, using the WGAN.

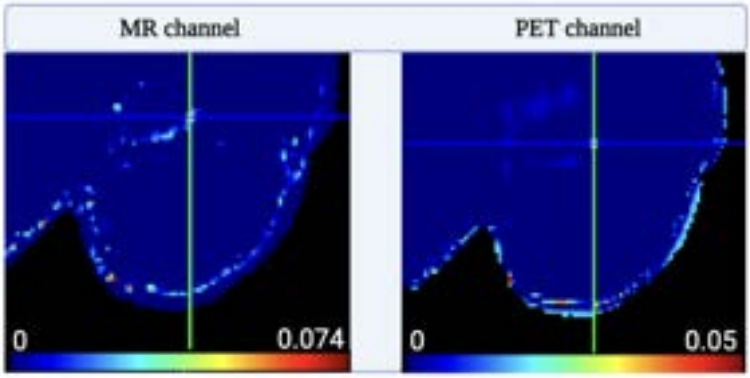

Figure 4.29: Magnification of the difference maps of MR and PET channels on the region where the lesion should be identified. A slightly higher intensity is visible in the lesion area.

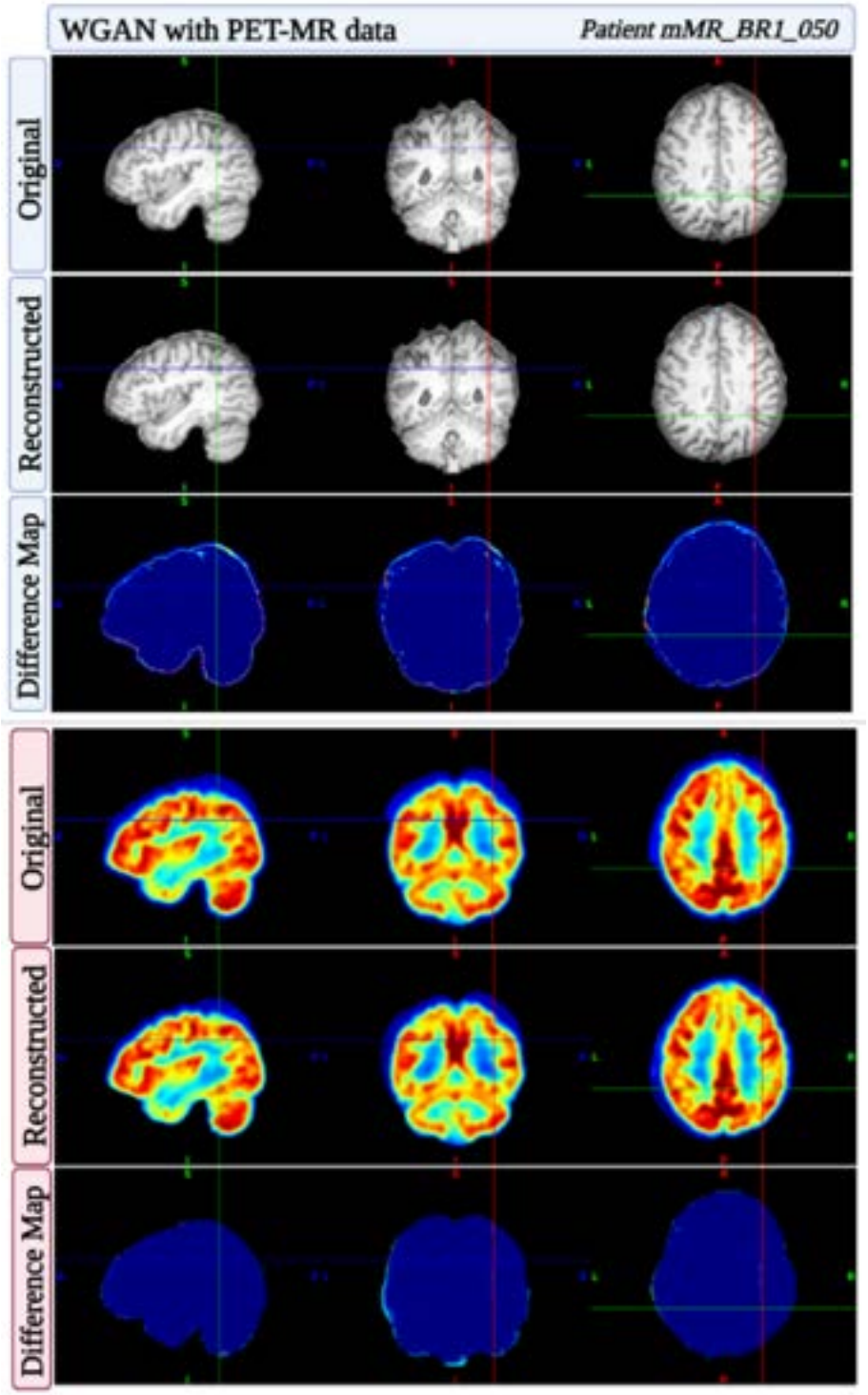

Figure 4.30: Original, Reconstructed and Difference maps images of both MRI (first set of three images) and PET channels (second set of three images), for patient mMR_BR1_050, using the WGAN.

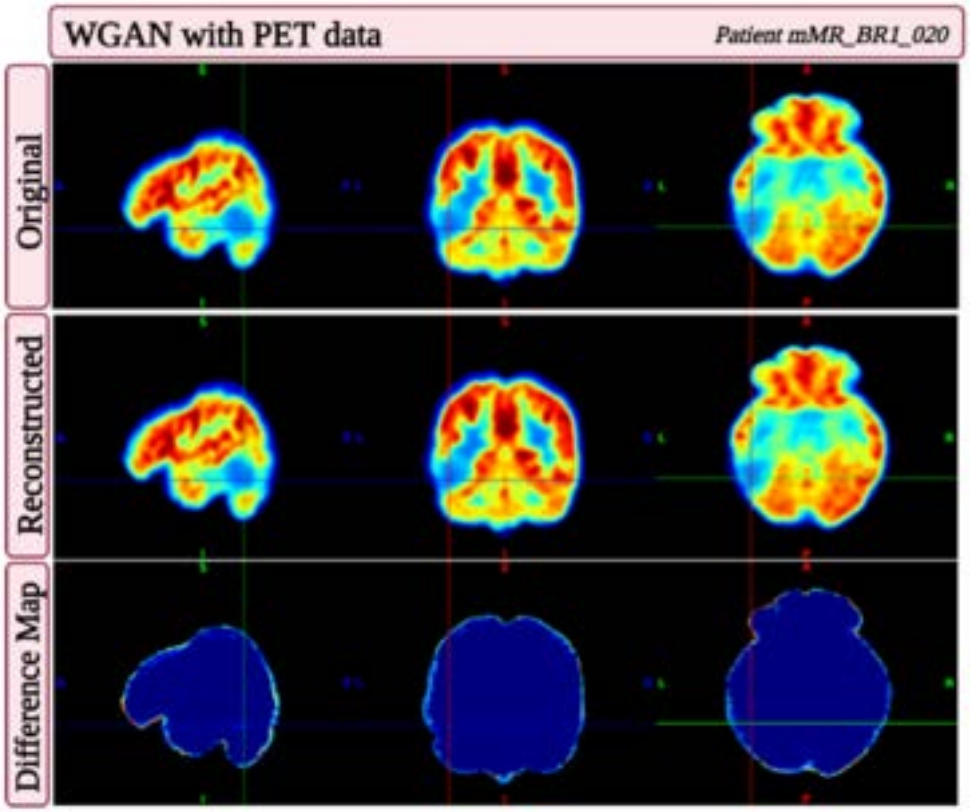

Figure 4.31: Original, Reconstructed and Difference maps images of PET scans, for patient mMR_BR1_020, using the WGAN only with PET modality.

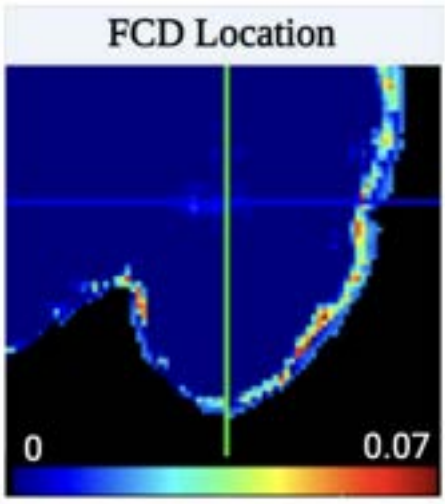

Figure 4.32: Magnification of the PET difference map in the region where the lesion should be identified. A slightly higher intensity is visible in the lesion area.

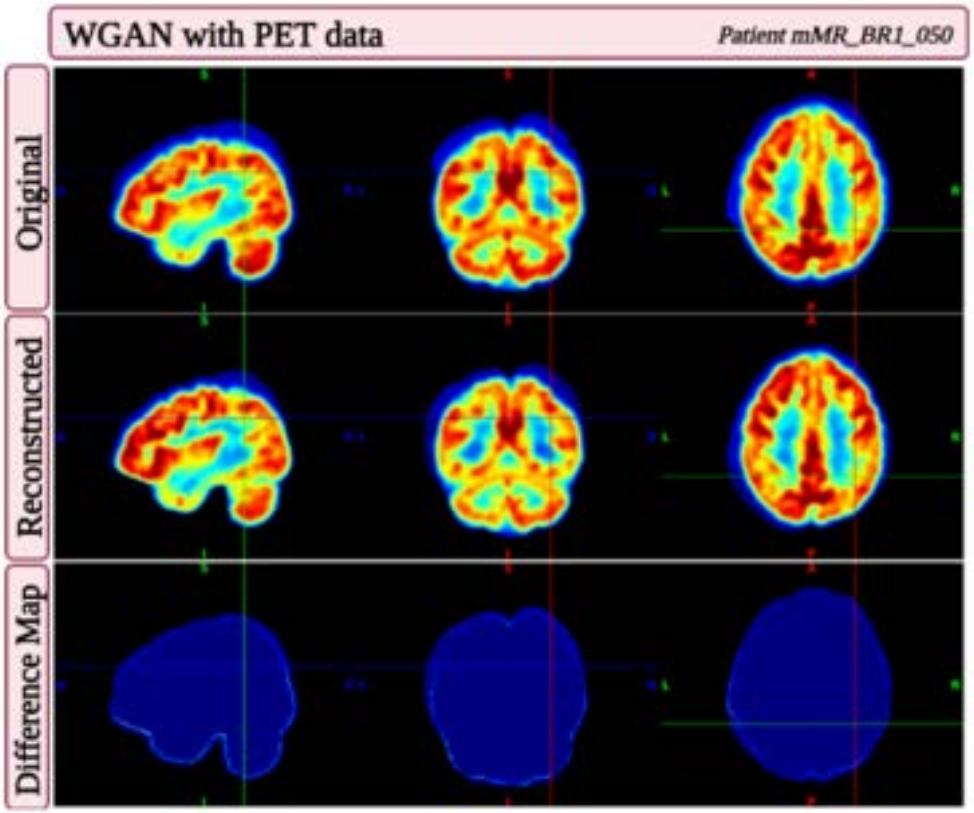

Figure 4.33: Original, Reconstructed and Difference maps images of PET scans, for patient mMR_BR1_050, using the WGAN only with PET modality.

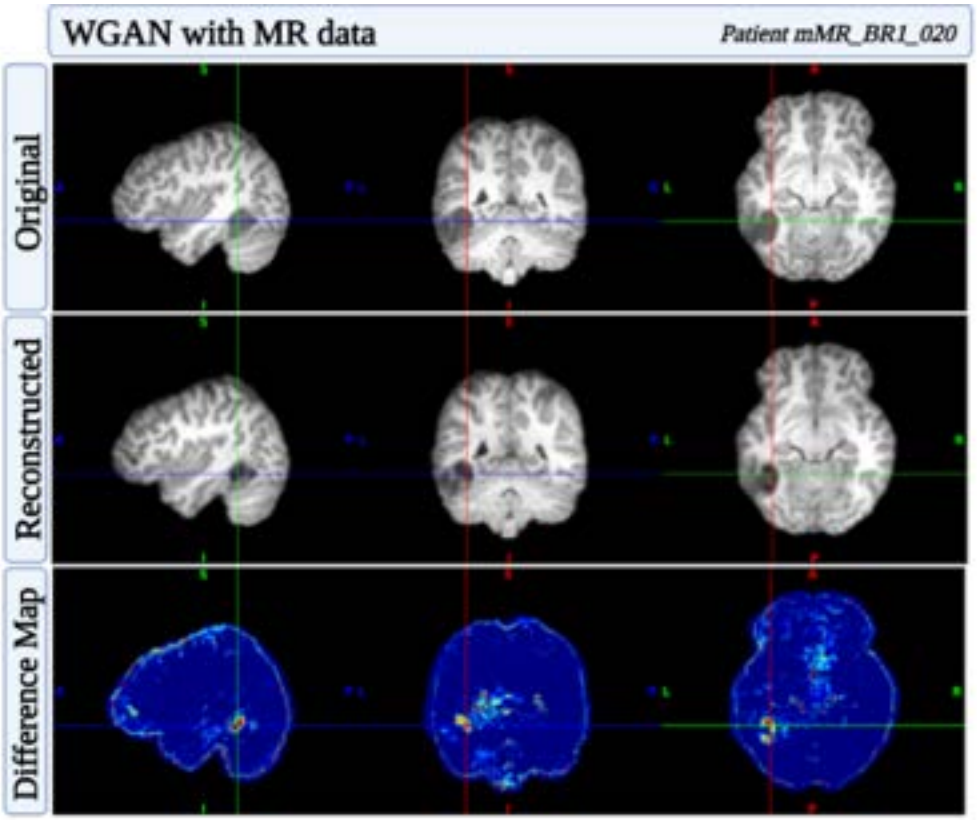

Figure 4.34: Original, Reconstructed and Difference maps images of MR scans, for patient mMR_BR1_020, using the WGAN only with MR modality.

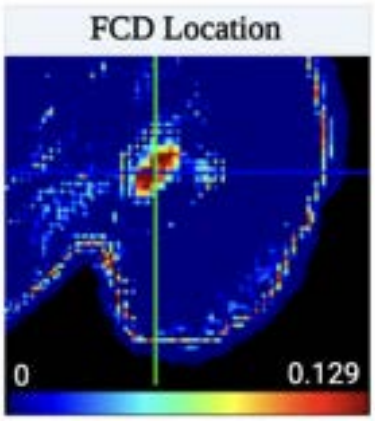

Figure 4.35: Magnification of the region of the MR difference map where the lesion should be identified. A cluster with higher intensity is visible in the lesion area.

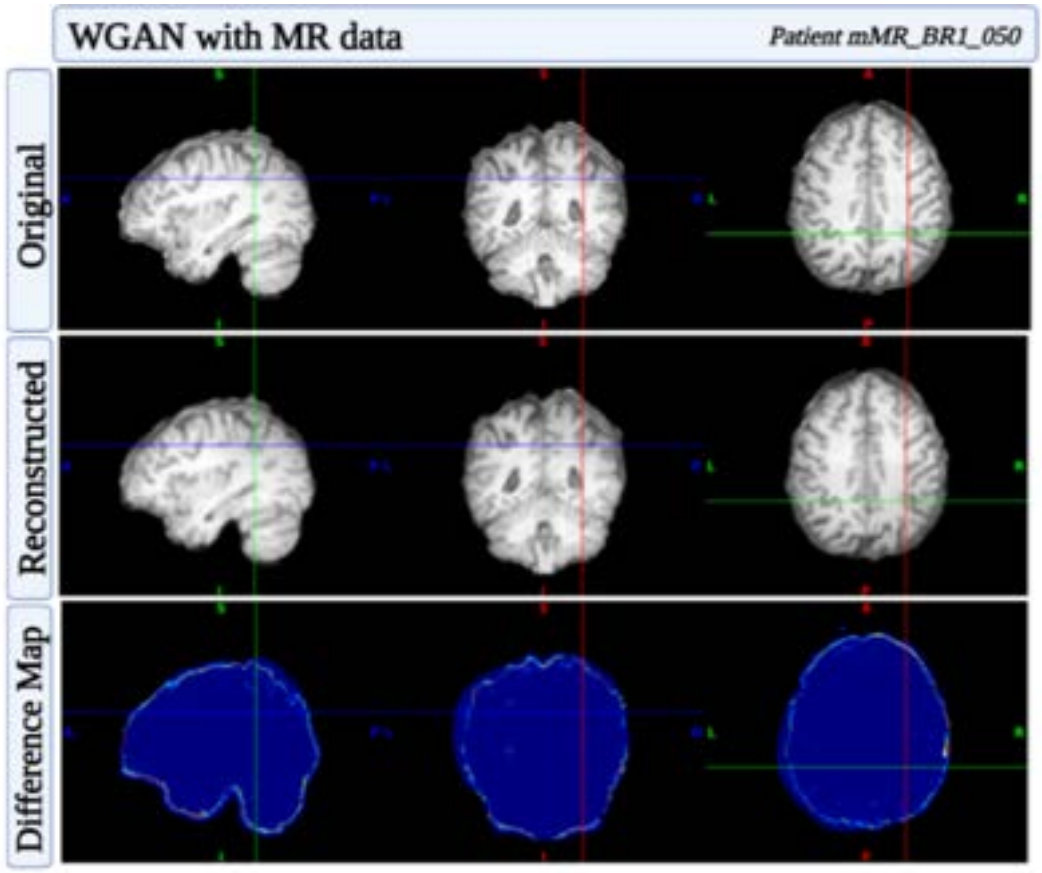

Figure 4.36: Original, Reconstructed and Difference maps images of MR scans, for patient mMR_BR1_050, using the WGAN only with MR modality.

4.4.2 CycleGAN

i) Multimodal data

Figure 4.37 corresponds to the original, translation to healthy, and difference map of patient mMR_BR1_020 obtained for the CycleGAN, using multimodal data for training. This figure show both MRI and PET channels. The axial, sagittal and coronal slices illustrated in the figure corresponds to the same region, where the lesion is located.

Figure 4.38 corresponds to the original, translation to healthy, and difference map of patient mMR_BR1_050 obtained for the CycleGAN, using multimodal data for training. This figure shows both MRI and PET channels. The axial, sagittal and coronal slices illustrated in the figure correspond to the same region, where the lesion is located.

ii) PET data

Figure 4.40 corresponds to the original, translation to healthy, and difference map of patient mMR_BR1_020 obtained for the CycleGAN, using a single image modality – PET. The axial, sagittal and coronal slices illustrated in the figure correspond to the same region, where the lesion is located.

Figure 4.41 corresponds to the original, translation to healthy, and difference map of patient mMR_BR1_050 obtained for the CycleGAN, using a single image modality – PET. The axial, sagittal and coronal slices illustrated in the figure correspond to the same region, where the lesion is located.

iii) MR data

Figure 4.42 corresponds to the original, translation to healthy, and difference map of patient mMR_BR1_020 obtained for the WGAN, using a single image modality – MRI. The axial, sagittal and coronal slices illustrated in the figure correspond to the same region, where the lesion is located.

Figure 4.43 corresponds to the original, translation to healthy, and difference map of patient mMR_BR1_050 obtained for the CycleGAN, using a single image modality – MRI. The axial, sagittal and coronal slices illustrated in the figure correspond to the same region, where the lesion is located.

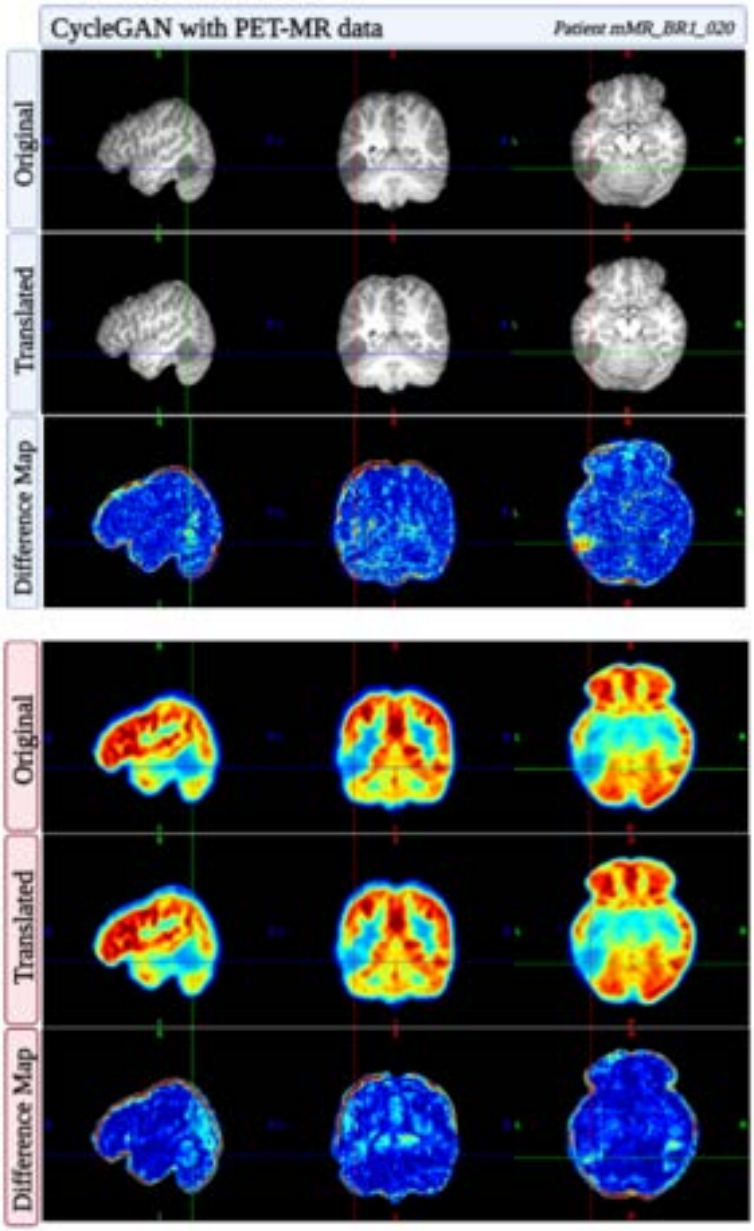

Figure 4.37: Original, Translated and Difference maps images of both MRI (first set of three images) and PET channels (second set of three images), for patient mMR_BR1_020, using the CycleGAN.

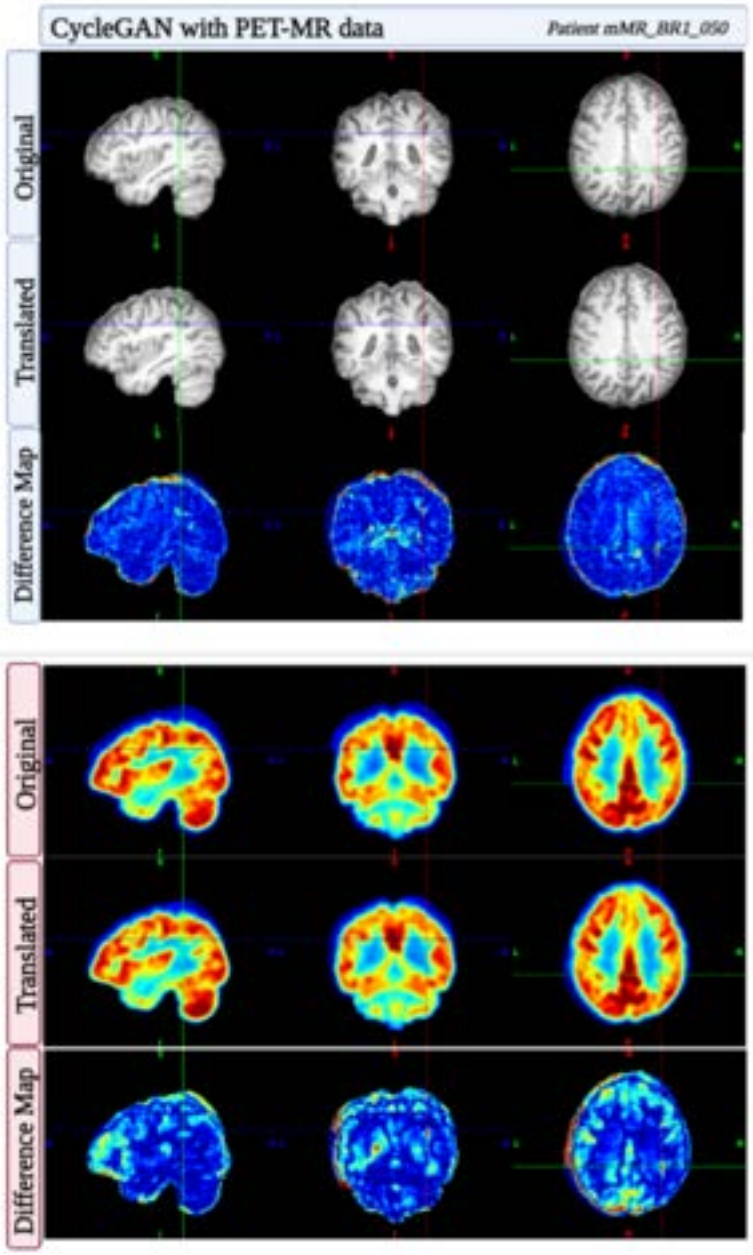

Figure 4.38: Original, Translated and Difference maps images of both MRI (first set of three images) and PET channels (second set of three images), for patient mMR_BR1_050, using the CycleGAN.

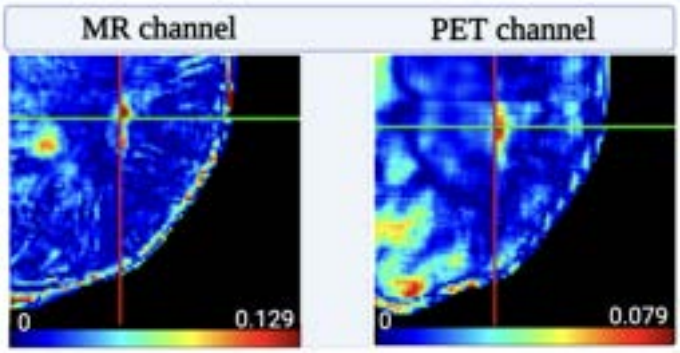

Figure 4.39: Magnification of the difference maps resulting from MR and PET channels in the region where the lesion should be identified. A cluster with higher intensity is visible in the lesion area.

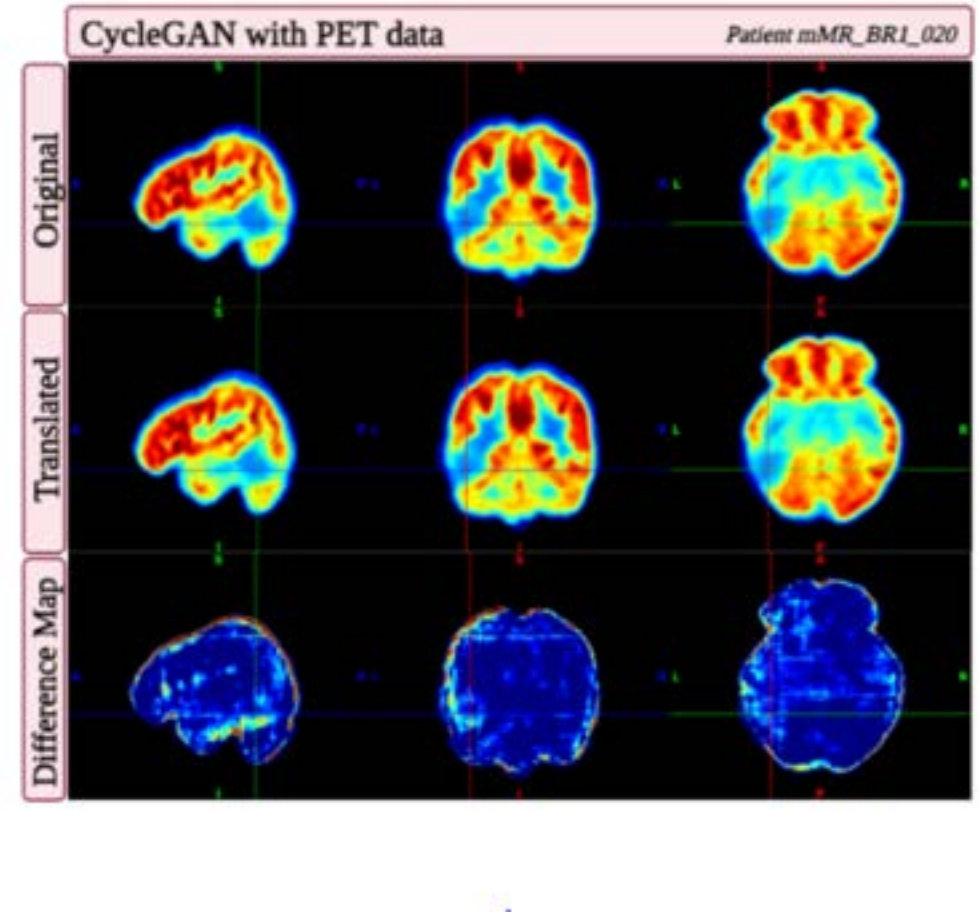

Figure 4.40: Original, Translated and Difference maps images of MRI, for patient mMR_BR1_020, using the CycleGAN only with PET data.

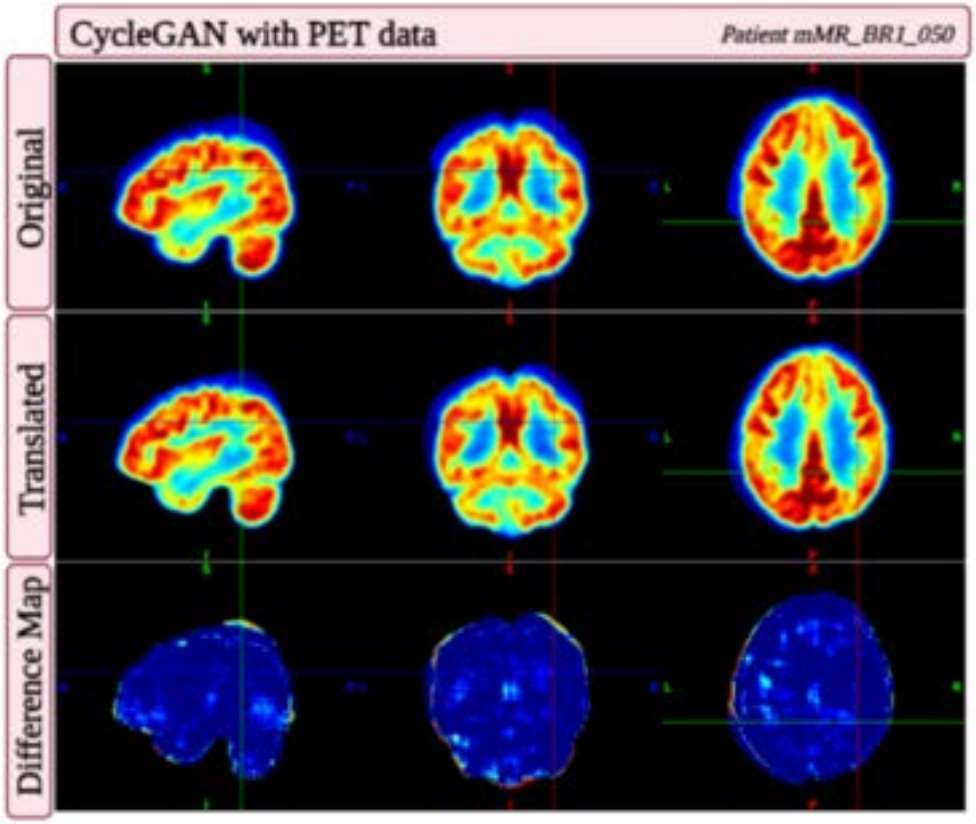

Figure 4.41: Original, Translated and Difference maps images of MRI, for patient mMR_BR1_050, using the CycleGAN only with PET data.

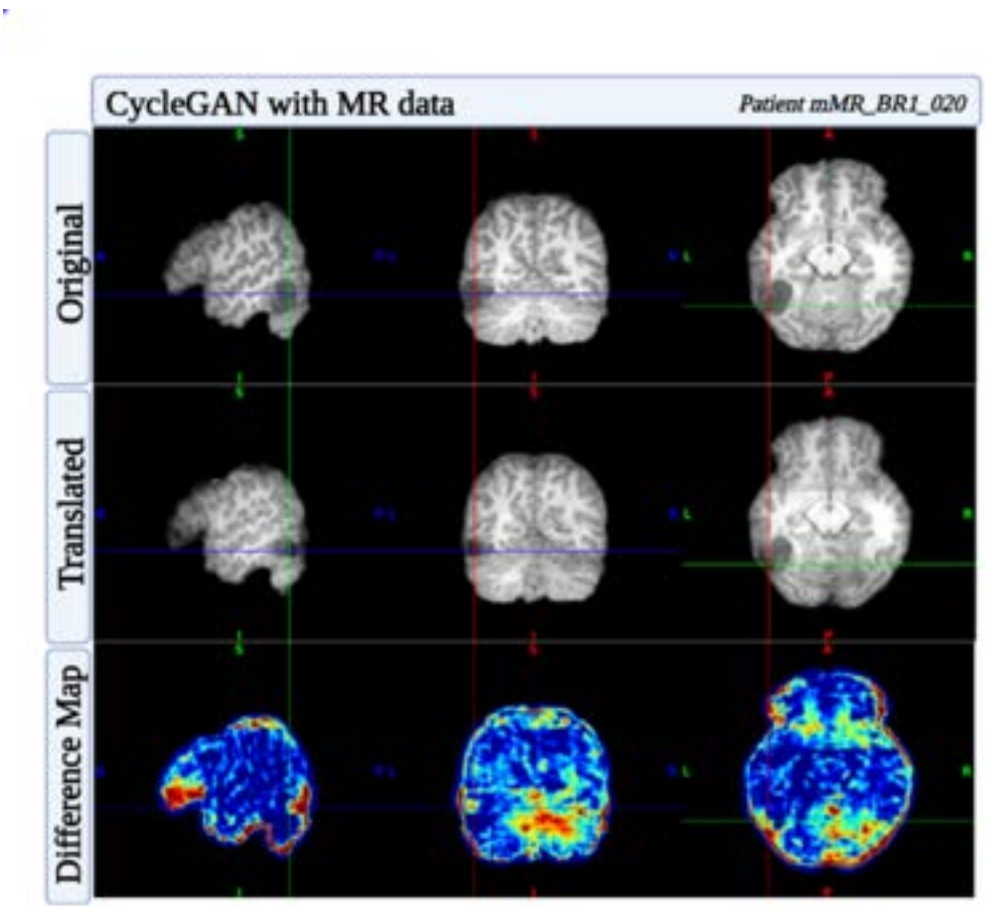

Figure 4.42: Original, Translated and Difference maps images of MRI, for patient mMR_BR1_020, using the CycleGAN only with MR data.

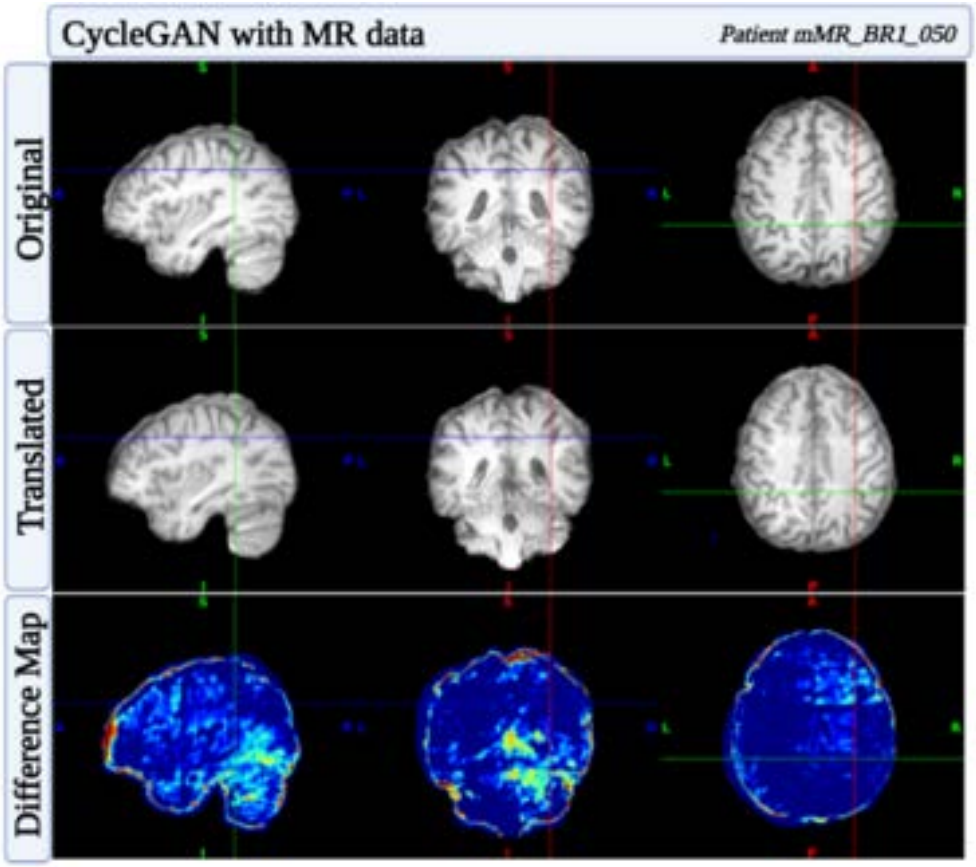

Figure 4.43: Original, Translated and Difference maps images of MRI, for patient mMR_BR1_050, using the CycleGAN only with MR data.

4.5 Discussion

4.5.1 Multimodal Data Training

i) WGAN

The result for lesion detection (difference maps) for patient mMR_BR1_020, using the WGAN, is presented in Figure 4.28. It shows the regions in the brain with the highest differences in intensity between the original images and its reconstructions (for both PET and MR channels of the input).

It can be observed that the WGAN, in both modality channels, was only able to detect a very subtle difference between the images (seen in more detail in Figure 4.29) in the location where the lesion was identified. This is indicated by the appearance of higher voxel intensity (in white/yellow in the image) located in the region that matches the lesion. However, for patient mMR_BR1_020 with a very visible lesion in both imaging modalities, it was desirable for the network to be able to identify this area in a more precise manner, forming a cluster that could be easily differentiated by eye such as in the work of *Yaakub et al.* [13] (see Figure 2.8).

Figure 4.30 shows the difference maps (of both the MR and PET channels) of patient mMR_BR1_050 for the multimodal WGAN. Both difference maps indicate that the WGAN cannot identify the subtle lesion of this patient in both neuroimaging modalities since it shows no clusters in the brain region. Therefore, the network only showed higher levels of pixel intensity outside the brain, which is a common occurrence in all difference maps that should be ignored since the brain tissue is the only area of interest to analyse. A similar method as the one applied by [13], which remasked the results to ignore any clusters in skull regions, could have also been performed to only show these reconstruction errors in the brain tissue.

The WGAN implemented in this work differentiates itself from the previous work of [13] by not using "ground-truth" control data (from healthy patients) as healthy tissue to train the networks. This complicates the networks' task since it uses its own data as a "ground-truth" healthy tissue (the areas the imaging experts did not consider diseased). However, as mentioned in section 4.2, neuroimaging experts can often have contradictory opinions on lesion location, especially when it comes to the visual identification of FCDs. Therefore, this can be a possible source of noise in the networks. This also resulted in a reduced dataset (in comparison to [13]) for training the networks, which was one of the reasons that motivated the use of patched-based training.

In summary, the multimodal WGAN showed potential in identifying anomalies in multiple modalities, especially in visible lesions such as tumours, but with clear room for improvement in future works.

ii) CycleGAN

The difference maps (for both PET and MR channels of the input) for patient mMR_BR1_020, using the CycleGAN, are in Figure 4.37, showing the difference between the original and translated image (to healthy), which illustrates the regions most modified by the CycleGAN. These regions indicate the areas the network considered as diseased, therefore modifying them to get a healthy version of the image, which will appear with a higher intensity in the difference maps.

For patient mMR_BR1_020, the difference maps of both channels show a subtle increase in voxel intensity in the lesion location, especially in the MR channel. However, it is possible to observe other high intensity areas on the difference maps, although not as intense when compared to the area where the lesion is located (this is more evident in the MR channel). In the PET channel, the model tries to modify the diseased hypometabolic area to a more "typically healthy" region (in this case to a less "bluish" region), which is observed in the translated image by a slight decrease in intensity. This shift is identified

in the difference map by a higher voxel intensity (represented by the colour yellow/green in Figure 4.37).

Observing the results (Figure 4.38) for the CycleGAN of patient mMR_BR1_050, the difference maps for MR and PET show a very small cluster with higher voxel intensities (seen in more detail in Figure 4.39). This cluster matches the lesion location on the MR scan; however, it is also possible to notice other background high intensity clusters (although of smaller dimensions) in the difference maps. It is also possible to see a region with higher intensity on the frontal lobe of the PET difference map (marked in yellow/green), although not in the form of a cluster and with lower intensity compared to the lesion cluster.

Overall, the CycleGAN presents more "noisy" difference maps (also observed in patient mMR_BR1_020), compared to the WGAN, where not only one cluster or high intensity region can be identified. However, the fact that the cluster matches the precise lesion location in both difference maps is a motivating factor to further improve this approach for better anomaly detection.

In both patients, the difference maps indicate that the network is changing regions outside the diseased areas. The anomaly mask loss was implemented in the CycleGAN to help alleviate this effect, aiming to focus the network's attention on the diseased areas by penalising the model for translating already healthy tissue. However, the large size of anomaly masks available (not with a very precise location but with a considerable extended area) could be contributing for the translation of some healthy regions by the CycleGAN. The anomaly mask loss therefore seems to not be able to entirely shift the attention of the network to only change diseased regions.

Overall, both networks using multimodal data showed challenges in training. The validation and training losses were monitored throughout the training and showed a difficulty in reaching an equilibrium between generator and discriminator – one of many known challenges of GANs. It is also worth mentioning that for the multimodal networks, one conditioning factor was that the number of random 3D patches sampled per batch was limited to five because of computational memory issues, expressing how computational costly these networks can be.

Having these problems in mind, several approaches were taken to improve training stability in the multimodality networks.

One of the approaches taken included adding noise to the discriminator networks (for both WGAN and CycleGAN). This consisted in randomly flipping the labels (0 or 1) in epochs that went through the discriminator alongside the "real" (input images in the WGAN for example) or "fake" images (reconstructions in the WGAN for example). This technique is commonly used to add noise to the discriminator to prevent it from improving very rapidly compared to the generator, which consequently prevents the generator from receiving positive feedback to be able to improve.

Another implementation used in this work was the replacement of transposed convolutions for upsampling [80] followed by a convolution operation, to prevent checkerboard artifacts related to transposed convolutions (discussed in [81]). This technique did show improvement in image quality overall (less patch artifacts were present in the reconstructions and translations of the WGAN and CycleGAN, respectively), but did not influence the performance of the networks in detecting lesions.

Although these applied techniques improved training for multimodal networks, they did not increase the performance of the networks in detecting the lesions. The multimodal training was then thought to be the source of training instability, which motivated the implementation of single-channel networks.

4.5.2 Single-Channel Data Training

i) WGAN with MR Data

For patient mMR_BR1_020, it can be observed that the WGAN was able to detect the lesion - represented by the cluster visible in the region where the lesion is in both the PET and MR scans - this indicates that the reconstructed image had a higher error in the locations where a possible anomaly was identified.

By looking at the reconstructed image (in Figure 4.34), it is clearly visible that the WGAN is not able to reconstruct the diseased area as well as the healthy tissue, resulting in the cluster in the difference map (Figure 4.34). The difference map shows the clearly higher voxel intensity that forms the cluster, corresponding to the lesion location (shown in more detail in the Figure 4.35). The difference map also shows some other high voxel intensity areas near the lesion location but are not as intense or as big clusters as the one identified.

However, for patient mMR_BR1_050, the network is still not able to identify the lesion location as seen in Figure 4.36.

ii) WGAN with PET Data

For patient mMR_BR1_020, the difference maps illustrated in Figure 4.31 show that the WGAN identified a slightly higher reconstruction error on the lesion location (presented in more detail in Figure 4.32) - represented by the higher voxel intensity in the region where the lesion is in.

For patient mMR_BR1_050, the difference maps illustrated in Figure 4.33 show that the WGAN could not identify the lesion location, similarly to the WGAN with only MR data and the WGAN with multimodal data.

Although the single-channel networks still showed difficulty in identifying the most subtle FCDs, they revealed some improvements in relation to the multimodal networks. For example, for both the single-channel WGAN networks, the training was much more stable compared to the multimodal WGAN. The training and validation losses showed their gradual decrease as expected in a GAN model and the network performance (image reconstruction quality) improved throughout the epochs as expected.

Specifically, the WGAN using only MR data identified a much higher intensity cluster (Figure 4.35) compared to the WGAN using MR and PET data.

iii) CycleGAN with MR Data

For patient mMR_BR1_020, Figure 4.42 shows the original, translated and difference map for the CycleGAN using only MR data. The difference map shows a slightly higher voxel intensity in the region where the lesion is in. Observing the translated image in Figure 4.42, it is possible to notice that the CycleGAN visibly modified the lesion location slightly (resulting in the high intensity value in the difference map). However, the difference map observed also has background noise, since it also shows other regions with high intensity values. Nevertheless, the cluster associated with the lesion appears to be the most prominent in the difference map.

For patient mMR_BR1_050, Figure 4.43 shows the original, translated and difference map for the CycleGAN using only MR data. The difference map does not show a high intensity value in the lesion location, indicating that the CycleGAN could not identify the subtle lesion, just as the single-channel CycleGAN using only PET data.

iv) CycleGAN with PET only

For patient mMR_BR1_020, Figure 4.40 shows the original, translated and difference map for the CycleGAN using only PET data. The difference map shows a high intensity region around the area where the lesion is located, however also showing some other background high intensities.

For patient mMR_BR1_050, Figure 4.41 shows the original, translated and difference map for the CycleGAN using only PET data. The difference map does not identify any relevant high intensity regions in the lesion location. However, it is possible to observe in the translated image of Figure 4.41, the shift in intensity (predominantly in hypometabolic regions) in the image compared to its original form. This indicates that the network is modifying the regions it considers diseased to a "healthy version" in the translated image (although not including the lesion area).

The single-channel networks still showed difficulty in identifying FCDs but also revealed some improvements in relation to the multimodal networks. For example, for both the single-channel CycleGAN networks, the training was much more stable compared to the multimodal CycleGAN. The monitorisation of the training and validation losses showed their gradual decrease as expected in a GAN model and the network performance (image translation quality) improved throughout the epochs as expected. The translated image also showed signs to be improving each epoch by changing the diseased area more than the healthy tissue, which was desired.

In general, the CycleGAN model showed more difficulty in training (either multimodal or single-channel), which is expected since it is a more complex network with a more challenging training set-up than the WGAN. However, it still showed signs to be able to detect visible anomalies such as tumours (in patient mMR_BR1_020). A further step of this work should also introduce a quantitative evaluation metric to better interpret the visual results obtained and identify false positives in difference maps, enabling to quantitatively compare the performance of models.

Overall, the multimodal networks show potential to be improved and applied for lesion detection since they can identify anomalies in regions that match the location of the lesions. The single-channel networks showed improvements in training balance but did not show a significant increase in performance compared to the multimodal networks when detecting the lesions, therefore suggesting that more improvements should be made not only in the multimodal data fusion method, but also in training methodology as well.

Further suggestions to improve and modify the multimodal networks in this dissertation are found in chapter 5.

Chapter 5

Conclusion and Future Work

Throughout this dissertation, various GAN models have been explored for different applications including image reconstruction, image translation and the ultimate goal of anomaly detection.

Image reconstruction and image translation tasks implemented in chapter 3 achieved high image quality using 2D data in both reconstruction and translation tasks. Training in 3D was more challenging when using a CycleGAN for T2-to-T1 MR image translation, nonetheless, showing GANs capabilities to learn image domain mappings.

Chapter 4 described and presented different methods for detecting lesions from PET and MR data - using reconstruction or translation approaches - and by initially combining both modalities for training. Since joint training with WGAN and CycleGAN showed difficulties in training stability, finding it hard to balance the discriminator and generator losses, single modality networks were tested additionally. These networks revealed a more stable training and presented more anomaly detection difference maps identifying lesions in the case of patient mMR_MR1_020. However, the very subtle FCD seen in patient mMR_BR1_050, proved to be undetectable by the networks.

Since combining modalities appeared to be the source of difficulty in training, some suggestions to improve this include recent papers such as [82], describing a GAN model entitled TarGAN (target-aware generative adversarial network) that could be modified for generation or translation tasks for this anomaly detection purpose. This model uses unpaired data to translate from one image modality to another by giving special focus on a translated target area within the image. Its architecture also follows inspiration from a CycleGAN (has a cycle-consistency loss) and a StarGAN [83], outperforming other state-of-the-art methods in segmentation tasks. Additionally, the TarGAN introduces a novel loss denominated crossing loss, which allows the generator to focus on the target area when performing the translation. Therefore, the CycleGAN implemented in this dissertation could be modified to support some of the techniques or losses from the TarGAN model.

Another recent work [84] proposed the use of separate normalization layers for each image modality, in a 3D U-net for segmentation to help dealing with the differences of intensity distributions in the image modalities. This method proved to outperform typical image modality fusion methods (as implemented in this dissertation) and individual modality training in networks for segmentation. Therefore, similar methods described in [84] will be implemented in future work. A review of the dataset's labels used could also be helpful by getting more accurate anomaly masks that impact patch-sampling regions for training the networks. More precise anomaly masks will translate in more diverse and better quality training data. Future work will therefore be focused on trying new ways of combining PET and MR data as well as improving network architecture and adding new losses specific for anomaly detection, with the hope of improving training stability and network performance.

Bibliography

[1] J. Kabat et al. “Focal cortical dysplasia - review”. In: *Polish Journal of Radiology* 77.2 (2012), pp. 35–43.

[2] *Overview | Epilepsies: diagnosis and management | Guidance | NICE*. URL: `https://www.nice.org.uk/guidance/cg137` (visited on 12/05/2021).

[3] J. O’Muircheartaigh et al. “Focal structural changes and cognitive dysfunction in juvenile myoclonic epilepsy”. In: *Neurology* 76.1 (2011), pp. 34–40.

[4] A. Hammers et al. “Neocortical abnormalities of [11C]-flumazenil PET in mesial temporal lobe epilepsy”. In: *Neurology* 56.7 (2001), pp. 897–906.

[5] M. Koepp et al. “11C-flumazenil PET in patients with refractory temporal lobe epilepsy and normal MRI”. In: *Neurology* 54.2 (2000), pp. 332–332.

[6] J. O’Muircheartaigh et al. “Abnormal thalamocortical structural and functional connectivity in juvenile myoclonic epilepsy”. In: *Brain* 135.12 (2012), pp. 3635–3644.

[7] E. Robinson et al. “MSM: a new flexible framework for Multimodal Surface Matching”. In: *Neuroimage* 100 (2014), pp. 414–426.

[8] M. Glasser et al. “A multi-modal parcellation of human cerebral cortex”. In: *Nature* 536.7615 (2016), pp. 171–178.

[9] E. Robinson et al. “Multimodal surface matching with higher-order smoothness constraints”. In: *Neuroimage* 167 (2018), pp. 453–465.

[10] A. Hammers et al. “Three-dimensional maximum probability atlas of the human brain, with particular reference to the temporal lobe”. In: *Human Brain Mapping* 19.4 (2003), pp. 224–247.

[11] Y. Tan et al. “Quantitative surface analysis of combined MRI and PET enhances detection of focal cortical dysplasias”. In: *Neuroimage* 166 (2018), pp. 10–18.

[12] S. Jayalakshmi et al. “Focal cortical dysplasia and refractory epilepsy: role of multimodality imaging and outcome of surgery”. In: *American Journal of Neuroradiology* 40.5 (2019), pp. 892–898.

[13] S. Yaakub et al. “Pseudo-normal PET synthesis with generative adversarial networks for localising hypometabolism in epilepsies”. In: *International Workshop on Simulation and Synthesis in Medical Imaging*. Shenzhen, China, 2019, pp. 42–51.

[14] L. Sun et al. “An adversarial learning approach to medical image synthesis for lesion detection”. In: *IEEE Journal of Biomedical and Health Informatics* 24.8 (2020), pp. 2303–2314.

[15] I. Goodfellow et al. “Generative adversarial nets”. In: *Advances in Neural Information Processing Systems*. Vol. 27. Quebec, Canada: Curran Associates, Inc., 2014, pp. 2672–2680.

[16] J. Langr and V. Bok. *GANs in action: deep learning with generative adversarial networks*. Manning Publications, 2019.

[17] M. Mirza and S. Osindero. "Conditional generative adversarial nets". In: *arXiv:1411.1784* (2014).

[18] A. Creswell et al. "Generative adversarial networks: an overview". In: *IEEE Signal Processing Magazine* 35.1 (2018), pp. 53–65.

[19] J. Kalin. *Generative Adversarial Networks Cookbook*. Packt Publishing, 2018.

[20] O. Ronneberger et al. "U-Net: Convolutional networks for biomedical image segmentation". In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI*. Munich, Germany, 2015, pp. 234–241.

[21] I. Gulrajani et al. "Improved training of wasserstein gans". In: *arXiv:1704.00028* (2017).

[22] S. Kazeminia et al. *GANs for medical image analysis*. Elsevier, 2020.

[23] J. Zhu et al. "Unpaired image-to-image translation using cycle-consistent adversarial networks". In: *Proceedings of the IEEE international Conference on Computer Vision*. Venice, Italy, 2017, pp. 2223–2232.

[24] P. Isola et al. "Image-to-image translation with conditional adversarial networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Honolulu, Hawaii, USA, 2017, pp. 1125–1134.

[25] A. Alotaibi. "Deep generative adversarial networks for image-to-image translation: A review". In: *Symmetry* 12.10 (2020), pp. 17–25.

[26] K. Armanious et al. "Unsupervised medical image translation using Cycle-MedGAN". In: *27th European Signal Processing Conference (EUSIPCO)*. IEEE. A Coruña, Spain, 2019, pp. 1–5.

[27] V. Chandola et al. "Anomaly detection: a survey". In: *Association for Computing Machinery Computing Surveys.* 41.3 (2009), pp. 1–58.

[28] V. Hodge et al. "A survey of outlier detection methodologies". In: *Artificial Intelligence Review* 22.2 (2004), pp. 85–126.

[29] J. O'Muircheartaigh et al. "Modelling brain development to detect white matter injury in term and preterm born neonates". In: *Brain* 143.2 (2020), pp. 467–479.

[30] S. Adler et al. "Novel surface features for automated detection of focal cortical dysplasias in paediatric epilepsy". In: *NeuroImage: Clinical* 14 (2017), pp. 18–27.

[31] W. Wei et al. "Predicting PET-derived demyelination from multimodal MRI using sketcher-refiner adversarial training for multiple sclerosis". In: *Medical Image Analysis* 58 (2019), pp. 101–115.

[32] K. Van Hespen et al. "An anomaly detection approach to identify chronic brain infarcts on MRI". In: *Scientific Reports* 11.1 (2021), pp. 1–10.

[33] T. Schlegl et al. "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery". In: *International Conference on Information Processing in Medical Imaging*. Boone, USA, 2017, pp. 146–157.

[34] V. Alex et al. "Generative adversarial networks for brain lesion detection". In: *Medical Imaging: Image Processing* 10133.2 (2017), pp. 13–30.

[35] X. Chen et al. "Unsupervised detection of lesions in brain MRI using constrained adversarial auto-encoders". In: *arXiv:1806.04972* (2018).

[36] A. Munawar et al. “Limiting the reconstruction capability of generative neural network using negative learning”. In: *IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP)*. Tokyo, Japan, 2017, pp. 1–6.

[37] *PyTorch*. URL: https://pytorch.org/ (visited on 10/26/2021).

[38] *MONAI - Home*. URL: https://monai.io/ (visited on 10/26/2021).

[39] *Developing Human Connectome Project (dHCP) | The Developing Human Connectome Project*. URL: http://www.developingconnectome.org/project/ (visited on 02/20/2021).

[40] E. Hughes et al. “A dedicated neonatal brain imaging system”. In: *Magnetic Resonance in Medicine* 78.2 (2017), pp. 794–804.

[41] A. Makropoulos et al. “The developing human connectome project: a minimal processing pipeline for neonatal cortical surface reconstruction”. In: *bioRxiv* (2018), pp. 88–112.

[42] R. Chalapathy and S. Chawla. “Deep learning for anomaly detection: A survey”. In: *arXiv: 1901.03407* (2019).

[43] M. Kuklisova-Murgasova et al. “Reconstruction of fetal brain MRI with intensity matching and complete outlier removal”. In: *Medical Image Analysis* 16.8 (2012), pp. 1550–1564.

[44] *Convolution Autoencoder - Pytorch | Kaggle*. URL: https://www.kaggle.com/ljlbarbosa/convolution-autoencoder-pytorch (visited on 01/22/2021).

[45] D. Ulyanov et al. “Instance normalization: the missing ingredient for fast stylization”. In: *arXiv: 1607.08022* (2017).

[46] C. Baumgartner et al. “Visual feature attribution using wasserstein gans”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Salt Lake City, UT, USA, 2018, pp. 8309–8319.

[47] *MSELoss — PyTorch 1.10.1 documentation*. URL: https://pytorch.org/docs/stable/generated/torch.nn.MSELoss.html (visited on 03/15/2021).

[48] *Adam — PyTorch master documentation*. URL: https://pytorch.org/docs/master/generated/torch.optim.Adam.html#adam (visited on 03/15/2021).

[49] *How to Implement Convolutional Autoencoder in PyTorch with CUDA*. URL: https://analyticsindiamag.com/how-to-implement-convolutional-autoencoder-in-pytorch-with-cuda/ (visited on 01/22/2021).

[50] *arnab39/cycleGAN-PyTorch: A clean and lucid implementation of cycleGAN using PyTorch*. URL: https://github.com/arnab39/cycleGAN-PyTorch (visited on 04/29/2021).

[51] *Transforms — MONAI 0.8.0 Documentation*. URL: https://docs.monai.io/en/stable/transforms.html#dictionary-transforms (visited on 05/10/2021).

[52] *Inference methods — MONAI 0.8.0 Documentation*. URL: https://docs.monai.io/en/latest/inferers.html#sliding-window-inference (visited on 05/10/2021).

[53] *Modules Overview — MONAI 0.8.0 Documentation*. URL: https://docs.monai.io/en/latest/highlights.html (visited on 05/10/2021).

[54] Y. Li et al. “Comparison of supervised and unsupervised deep learning methods for medical image synthesis between computed tomography and magnetic resonance images”. In: *BioMed Research International* (2020).

[55] M. Severino et al. “Definitions and classification of malformations of cortical development: practical guidelines”. In: *Brain* 143.10 (2020), pp. 2874–2894.

[56] E. Knight et al. “Pre-operative evaluation in pediatric patients with cortical dysplasia”. In: *Child’s Nervous System* 31.12 (2015), pp. 2225–2233.

[57] N. Salamon et al. “FDG-PET/MRI coregistration improves detection of cortical dysplasia in patients with epilepsy”. In: *Neurology* 71.20 (2008), pp. 1594–1601.

[58] S. Desarnaud et al. “18F-FDG PET in drug-resistant epilepsy due to focal cortical dysplasia type 2: additional value of electroclinical data and coregistration with MRI”. In: *European Journal of Nuclear Medicine and Molecular Imaging* 45.8 (2018), pp. 1449–1460.

[59] *The PET Centre*. URL: `http://www.sthpetcentre.org.uk/` (visited on 09/27/2021).

[60] R. Robinson et al. “Image-level harmonization of multi-site data using image-and-spatial transformer networks”. In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2020*. Lima, Peru, 2020, pp. 710–719.

[61] M. Rossi. “Focal cortical dysplasia-associated tumors: resecting beyond the lesion.” In: *Epilepsy Currents* 14.5 (2014), pp. 115–122.

[62] V. Berti et al. “Brain: normal variations and benign findings in fluorodeoxyglucose-PET/computed tomography imaging”. In: *PET Clinics* 9.2 (2014), pp. 129–140.

[63] S. Smith. “Fast robust automated brain extraction”. In: *Human Brain Mapping* 17.3 (2002), pp. 143–155.

[64] M. Jenkinson et al. “FSL”. In: *NeuroImage* 62.2 (2012), pp. 782–790.

[65] *Atlases - FslWiki*. URL: `https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/Atlases` (visited on 06/29/2021).

[66] M. Jenkinson et al. “A global optimisation method for robust affine registration of brain images”. In: *Medical Image Analysis* 5.2 (2001), pp. 143–156.

[67] *FNIRT/UserGuide*. URL: `http://ftp.nmr.mgh.harvard.edu/pub/dist/freesurfer/tutorial_packages/centos6/fsl_507/doc/wiki/FNIRT(2f)UserGuide.html` (visited on 05/20/2021).

[68] F. López-González et al. “Intensity normalization methods in brain FDG-PET quantification”. In: *NeuroImage* 222 (2020), pp. 117–122.

[69] C. Sprinz et al. “Effects of blood glucose level on 18F-FDG uptake for PET/CT in normal organs: A systematic review”. In: *PLOS ONE* 13.2 (2018), pp. 13–27.

[70] K. Mortensen et al. “Impact of global mean normalization on regional glucose metabolism in the human brain”. In: *Neural Plasticity* (2018), pp. 1–16.

[71] L. Nyúl et al. “New variants of a method of MRI scale standardization”. In: *IEEE Transactions on Medical Imaging* 19.2 (2000), pp. 143–150.

[72] I. Gousias et al. “Automatic segmentation of brain MRIs of 2-year-olds into 83 regions of interest”. In: *NeuroImage* 40.2 (2008), pp. 672–684.

[73] I. Faillenot et al. “Macroanatomy and 3D probabilistic atlas of the human insula”. In: *NeuroImage* 150 (2017), pp. 88–98.

[74] *Brain Development*. URL: `http://brain-development.org/` (visited on 01/29/2021).

[75] *FSL / fsleyes / fsleyes · GitLab*. URL: https://git.fmrib.ox.ac.uk/fsl/fsleyes/fsleyes/ (visited on 05/29/2021).

[76] *Transforms — MONAI 0.8.0 Documentation*. URL: https://docs.monai.io/en/latest/transforms.html#randweightedcropd (visited on 05/12/2021).

[77] Y. Dong Zhang et al. "Advances in multimodal data fusion in neuroimaging: Overview, challenges, and novel orientation". In: *An International Journal on Information Fusion* 64 (2020), pp. 149–160.

[78] D. Ramachandram et al. "Deep multimodal learning: A survey on recent advances and trends". In: *IEEE Signal Processing Magazine* 34.6 (2017), pp. 96–108.

[79] *Transforms — MONAI 0.8.0 Documentation*. URL: https://docs.monai.io/en/latest/transforms.html#randflipd (visited on 05/12/2021).

[80] *Upsample — PyTorch 1.10.1 documentation*. URL: https://pytorch.org/docs/stable/generated/torch.nn.Upsample.html (visited on 10/26/2021).

[81] *Deconvolution and Checkerboard Artifacts*. URL: https://distill.pub/2016/deconv-checkerboard/?utm_source=researcher_app&utm_medium=referral&utm_campaign=RESR_MRKT_Researcher_inbound (visited on 10/27/2021).

[82] J. Chen et al. "TarGAN: target-aware generative adversarial networks for multi-modality medical image translation". In: *Medical Image Computing and Computer Assisted Intervention – MICCAI*. Strasbourg, France, 2021, pp. 24–33.

[83] Y. Choi et al. "StarGAN: unified generative adversarial networks for multi-domain image-to-image translation". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Salt Lake City, UT, USA, 2018, pp. 8789–8797.

[84] Q. Dou et al. "Unpaired Multi-Modal Segmentation via Knowledge Distillation". In: *IEEE Transactions on Medical Imaging* 39.7 (2020), pp. 2415–2425.

Appendix A

Appendix

The tables presented in the appendix report the layer parameters and the hyperparameters used for training all the networks in this dissertation. Additionally, Table A.7 includes the description of the dataset used in chapter 4.

Table A.1: Illustration of the layer parameters of the 2D Autoencoder represented in Figure 3.3. Layers E1-E3 downsample (through the convolution operation) the input and layers D1-D3 upsample (through the transposed convolution operation) the input. The information described in the table corresponds to the parameters used for the 2D convolutional operation, in the E1-E3 layers, and the 2D transposed convolution operation, in the D1-D3 layers. These parameters consist in the number of input channels, number of output channels, filter size, stride, and padding – the parameters with only one value indicate that it is applied for all dimensions.

2D Autoencoder	
E1	In channels= 1 ; Out channels= 16 ; Filter= [2, 2] ; Stride= 2 ; Padding= 0
E2	In channels= 16 ; Out channels= 32 ; Filter= [4, 3] ; Stride= 2 ; Padding= 0
E3	In channels= 32 ; Out channels= 64 ; Filter= [2,3] ; Stride= 2 ; Padding= 0
D1	In channels= 64 ; Out channels= 32 ; Filter= [2, 3] ; Stride= 2 ; Padding= 0
D2	In channels= 32 ; Out channels= 16 ; Filter= [4, 3] ; Stride= 2 ; Padding= 0
D3	In channels= 16 ; Out channels= 1 ; Filter= [2, 2] ; Stride= 2 ; Padding= 0

Table A.2: Illustration of the layer parameters of the 2D U-net represented in Figure 3.4. Layers E1-E3 downsample (through the convolution operation) the input and layers D1-D3 upsample (through the transposed convolution operation) the input. The information described in the table corresponds to the parameters used for the 2D convolutional operation, in the E1-E3 layers, and the 2D transposed convolution operation, in the D1-D3 layers. These parameters consist in the number of input channels, number of output channels, filter size, stride, and padding – the parameters with only one value indicate that it is applied for all dimensions. Layers D2 and D3 have channels multiplied by 2 because of the skip connections present in the network.

2D U-net	
E1	In channels= 1 ; Out channels= 16 ; Filter= [2, 2] ; Stride= 2 ; Padding= 0
E2	In channels= 16 ; Out channels= 32 ; Filter= [4, 3] ; Stride= 2 ; Padding= 0
E3	In channels= 32 ; Out channels= 64 ; Filter= [2,3] ; Stride= 2 ; Padding= 0
D1	In channels= 64 ; Out channels= 32 ; Filter= [2, 3] ; Stride= 2 ; Padding= 0
D2	In channels= 32*2 ; Out channels= 16 ; Filter= [4, 3] ; Stride= 2 ; Padding= 0
D3	In channels= 16*2 ; Out channels= 1 ; Filter= [2, 2] ; Stride= 2 ; Padding= 0

Table A.3: Illustration of the layer parameters of the 3D U-net represented in Figure 3.5. Layers E1-E6 downsample (through the convolution operation) the input and layers D1-D6 upsample (through the transposed convolution operation) the input. The information described in the table corresponds to the parameters used for the 3D convolutional operation, in the E1-E6 layers, and the 3D transposed convolution operation, in the D1-D6 layers. These parameters consist in the number of input channels, number of output channels, filter size, stride, and padding – the parameters with only one value indicate that it is applied for all dimensions. Layers D2 to D6 have channels multiplied by 2 because of the skip connections present in the network.

3D UNET	
E1	In channels= 1 ; Out channels= 16 ; Filter= 4 ; Stride= 2 ; Padding= 1
E2	In channels= 16 ; Out channels= 32 ; Filter= 4 ; Stride= 2 ; Padding= 1
E3	In channels= 32 ; Out channels= 64 ; Filter= 4 ; Stride= 2 ; Padding= 1
E4	In channels= 64 ; Out channels= 128 ; Filter= 4 ; Stride= 2 ; Padding= 1
E5	In channels= 128 ; Out channels= 256 ; Filter= 4 ; Stride= 2 ; Padding= 1
E6	In channels= 256; Out channels= 512 ; Filter= 4 ; Stride= 2 ; Padding= 1
D1	In channels= 512 ; Out channels= 256 ; Filter= 4 ; Stride= 2 ; Padding= 1
D2	In channels= 256*2 ; Out channels= 128 ; Filter= 4 ; Stride= 2 ; Padding= 1
D3	In channels= 128*2 ; Out channels= 64 ; Filter= 4 ; Stride= 2 ; Padding= 1
D4	In channels= 64*2 ; Out channels= 32 ; Filter= 4 ; Stride= 2 ; Padding= 1
D5	In channels= 32*2 ; Out channels= 16 ; Filter= 4 ; Stride= 2 ; Padding= 1
D6	In channels= 16*2 ; Out channels= 1 ; Filter= 4 ; Stride= 2 ; Padding= 1

Table A.4: Illustration of the layer parameters of the Critic of the 2D WGAN represented in Figure 3.6. Layers L1-L7 downsample (through the convolution operation) the input to obtain a classification score. The information described in the table corresponds to the parameters used for the 2D convolutional operation, in the L1-L7 layers. These parameters consist in the number of input channels, number of output channels, filter size, stride, and padding – the parameters with only one value indicate that it is applied for all dimensions.

Critic WGAN	
L1	In channels= 1 ; Out channels= 16 ; Filter= 3 ; Stride= 2 ; Padding= 0
L2	In channels= 16 ; Out channels= 32 ; Filter= 3 ; Stride= 2 ; Padding= 0
L3	In channels= 32 ; Out channels= 64 ; Filter= 3 ; Stride= 2 ; Padding= 0
L4	In channels= 64 ; Out channels= 128 ; Filter= 3 ; Stride= 2 ; Padding= 0
L5	In channels= 128 ; Out channels= 256 ; Filter= 3 ; Stride= 2 ; Padding= 0
L6	In channels= 256; Out channels= 256 ; Filter= 3 ; Stride= 2 ; Padding= 0
L7	In channels= 256 ; Out channels= 1 ; Filter= 1 ; Stride= 1 ; Padding= 0

Table A.5: Layer parameters for both the Generators of the CycleGAN illustrated in Figure 3.7. The information described in the table corresponds to the parameters used for the convolutional operation (in E1-L6 layers) and for the transposed convolutional operation (in L6-D5 layers). These parameters consist in the number of input channels, number of output channels, the filter size, stride, and padding (in which filter size, stride and padding have one value that is applied for all dimensions). Layers E1 to E5 represent encoding layers, as well as the convolutional operation in L6, which downsample the image by a factor of 2. Layer L6 represents the innermost layer of the network – formed by a convolutional operation, a ReLU activation function, a transposed convolutional operation, an instance normalisation layer and a final ReLU activation function. The transposed convolution operation in L6, as well as layers D1 to D5 represent decoding layers that upsample the image by a factor of 2. Layers D1 to D5 have in channels multiplied by 2 because of the skip connections present in the network.

Generator – 2D and 3D CycleGAN	
E1	In channels= 2 ; Out channels= 64 ; Filter= 4 ; Stride= 2 ; Padding= 1
E2	In channels= 64 ; Out channels= 128 ; Filter= 4 ; Stride= 2 ; Padding= 1
E3	In channels= 128 ; Out channels= 256 ; Filter= 4 ; Stride= 2 ; Padding= 1
E4	In channels= 256 ; Out channels= 512 ; Filter= 4 ; Stride= 2 ; Padding= 1
E5	In channels= 512 ; Out channels= 512 ; Filter= 4 ; Stride= 2 ; Padding= 1
L6	Conv3D: In channels= 512 ; Out channels= 512 ; Filter= 4 ; Stride= 2 ; Padding = 1 TransConv3D: In channels= 512 ; Out channels= 512 ; Filter= 4 ; Stride= 2 ; Padding= 1
D1	In channels= 512*2 ; Out channels= 512 ; Filter= 4 ; Stride= 2 ; Padding= 1
D2	In channels= 512*2 ; Out channels= 256 ; Filter= 4 ; Stride= 2 ; Padding= 1
D3	In channels= 256*2 ; Out channels= 128 ; Filter= 4 ; Stride= 2 ; Padding= 1
D4	In channels= 128*2 ; Out channels= 64 ; Filter= 4 ; Stride= 2 ; Padding= 1
D5	In channels= 64*2 ; Out channels= 2 ; Filter= 4 ; Stride= 2 ; Padding= 1

Table A.6: Illustration of the layer parameters of the Discriminator of the 2D and 3D CycleGAN represented in Figure 3.8. Layers L1-L5 downsample (through the convolution operation) the input to obtain a classification score. The information described in the table corresponds to the parameters used for the convolutional operation, in the L1-L7 layers. These parameters consist in the number of input channels, number of output channels, filter size, stride, and padding – the parameters with only one value indicate that it is applied for all dimensions.

Discriminator - 2D and 3D CycleGAN	
L1	In channels= 1 ; Out channels= 64 ; Filter= 4 ; Stride= 2 ; Padding= 1
L2	In channels= 64 ; Out channels= 128 ; Filter= 4 ; Stride= 2 ; Padding= 1
L3	In channels= 128 ; Out channels= 256 ; Filter= 4 ; Stride= 2 ; Padding= 1
L4	In channels= 256 ; Out channels= 512 ; Filter= 4 ; Stride= 2 ; Padding= 1
L5	In channels= 512 ; Out channels= 1 ; Filter= 4 ; Stride= 1 ; Padding= 1

Table A.7: Information about the subjects used in this project including: age, gender, category of the lesion and the location the region was found.

Subject ID	Age	Gender	Category	Lesion Location
mMR_BR1_002	37	F	MR-PET+	Bil TL, Bil SFG, Bil PL
mMR_BR1 _004	50	F	MR-PET+	L Insula
mMR_BR1_008	16	M	MR-PET+	L TL
mMR_BR1_014	18	F	MR+PET+	L superior PL, L TL
mMR_BR1_019	26	M	MR-PET+	Bil TL, L OFC
mMR_BR1_020	14	F	MR+PET+	L TL
mMR_BR1_021	30	F	MR+PET+	Bil Temporal Poles, Bil Medial TL, R PL
mMR_BR1_022	15	F	MR-PET+	R TL
mMR_BR1_026	20	M	MR+PET+	L TL
mMR_BR1_030	17	F	MR+PET+	R superior TL
mMR_BR1_033	40	M	MR-PET+	L Insula
mMR_BR1_035	13	M	MR-PET+	R STG, R Temporal Pole
mMR_BR1_044	32	M	MR-PET+	R PL
mMR_BR1_046	25	F	MR+PET+	L TL, L Insula
mMR_BR1_047	45	F	MR-PET+	R TL, R insula
mMR_BR1_049	16	F	MR-PET+	R Pre-Central Gyrus, R Post-Central Gyrus, Precuneus
mMR_BR1_050	70	F	MR+PET+	R PL
mMR_BR1_055	59	F	MR-PET+	Bil TL, Bil FL, R Central Regions
mMR_BR1_058	56	F	MR+PET+	L TL, L PL
mMR_BR1_059	14	M	MR+PET+	R TL
mMR_BR1_062	17	M	MR-PET+	L Insula, Bil Hippocampus
mMR_BR1_067	27	M	MR-PET+	L TL, L Temporal Pole, L Insular

L - Left ; R - Right ; Bil - Bilateral ; TL - Temporal Lobe ; FL - Frontal Lobe ; PL - Parietal ; OFC - Orbitofrontal Cortex ; SFG - Superior Frontal Gyrus ; STG - Superior Temporal Gyrus

Table A.8: Illustration of the Generator's architecture. The information in the table corresponds to the parameters used for the 3D convolutional or transposed convolution operation, per layer. These parameters consist of the number of input channels, number of output channels, filter size, stride, and padding (in which filter size, stride and padding have one value that is applied for all dimensions). Layers E1 to E6 represent encoding layers (illustrated in Figure 4.17) that down sample the image (through the convolution operation) by a factor of 2. Layers D1 to D6 represent decoding layers (illustrated also in Figure 4.17) that up sample the image (through the transposed convolution operation) by a factor of 2. Layers D2 to D6 have channels multiplied by 2 because of the skip connections present in the network.

Generator – WGAN	
E1	In channels= 2 ; Out channels= 16 ; Filter= 4 ; Stride= 2 ; Padding= 1
E2	In channels= 16 ; Out channels= 32 ; Filter= 4 ; Stride= 2 ; Padding= 1
E3	In channels= 32 ; Out channels= 64 ; Filter= 4 ; Stride= 2 ; Padding= 1
E4	In channels= 64 ; Out channels= 128 ; Filter= 4 ; Stride= 2 ; Padding= 1
E5	In channels= 128 ; Out channels= 256 ; Filter= 4 ; Stride= 2 ; Padding= 1
E6	In channels= 256 ; Out channels= 512 ; Filter= 4 ; Stride= 2 ; Padding= 1
D1	In channels= 512 ; Out channels= 256 ; Filter= 4 ; Stride= 2 ; Padding= 1
D2	In channels= 256*2 ; Out channels= 128 ; Filter= 4 ; Stride= 2 ; Padding= 1
D3	In channels= 128*2 ; Out channels= 64 ; Filter= 4 ; Stride= 2 ; Padding= 1
D4	In channels= 64*2 ; Out channels= 32 ; Filter= 4 ; Stride= 2 ; Padding= 1
D5	In channels= 32*2 ; Out channels= 16 ; Filter= 4 ; Stride= 2 ; Padding= 1
D6	In channels= 16*2 ; Out channels= 2 ; Filter= 4 ; Stride= 2 ; Padding= 1

Table A.9: Illustration of the Critic's architecture. Layers L1 to L5 (illustrated in Figure 4.18) down sample the image (through the convolution operation). The information in the table correspond to the parameters used in the 3D convolutional operation, in each layer. These parameters consist of the number of input channels, number of output channels, filter size, stride, and padding (in which filter size, stride and padding have one value that is applied for all dimensions).

Critic – WGAN	
L1	In channels= 2 ; Out channels= 64 ; Filter= 4 ; Stride= 2 ; Padding= 1
L2	In channels= 64 ; Out channels= 128 ; Filter= 4 ; Stride= 2 ; Padding= 1
L3	In channels= 128 ; Out channels= 256 ; Filter= 4 ; Stride= 2 ; Padding= 1
L4	In channels= 256 ; Out channels= 512 ; Filter= 4 ; Stride= 2 ; Padding= 1
L5	In channels= 512 ; Out channels= 1 ; Filter= 4 ; Stride= 1 ; Padding= 0

Table A.10: Hyperparameters chosen to train the WGAN. These include batch-size, patch-size (size of the 3D patches to be sampled in the whole image), learning rate, the β parameter of the Adam optimiser chosen (for both the Generator and Discriminator), the critic iterations (the number of iterations of the critic per generator iterations), and the λ values applied to the L1 loss and gradient penalty.

Hyperparameters - WGAN	
Batch-size	5
Patch-size	64 x 64 x 64
Learning Rate	1e-04
Adam Optimiser	β = (0, 0.9)
Critic Iterations	5
λ L1 loss	0.001
λ gradient penalty	10

Table A.11: Illustration of the Discriminator's architecture (same architecture used for D_N and D_A) for the 3D CycleGAN, illustrated in Figures 4.22 and 4.23. Layers L1 to L5 downsample the image (through the convolution operation). The information described in the table corresponds to the parameters used for the 3D convolutional operation, in each layer. These parameters consist in the number of input channels, number of output channels, filter size, stride, and padding (in which filter size, stride and padding have one value that is applied for all dimensions).

Discriminator CycleGAN	
L1	In channels= 2 ; Out channels= 64 ; Filter= 4 ; Stride= 2 ; Padding= 1
L2	In channels= 64 ; Out channels= 128 ; Filter= 4 ; Stride= 2 ; Padding= 1
L3	In channels= 128 ; Out channels= 256 ; Filter= 4 ; Stride= 2 ; Padding= 1
L4	In channels= 256 ; Out channels= 512 ; Filter= 4 ; Stride= 2 ; Padding= 1
L5	In channels= 512 ; Out channels= 1 ; Filter= 4 ; Stride= 1 ; Padding= 0

Table A.12: Illustration of the Generator's architecture (same for both the A2N and N2A Generators) of the 3D CycleGAN in Figure 4.24. The information described in the table corresponds to the parameters used for the 3D convolutional operation, in each layer. These parameters consist in the number of input channels, number of output channels, the filter size, stride, and padding (in which filter size, stride and padding have one value that is applied for all dimensions). Layers E1 to E5 represent encoding layers, as well as the convolutional operation in L6, which downsample the image (through the convolution operation) by a factor of 2. Layer L6 represents the innermost layer of the network – formed by a convolutional operation, a ReLU activation function, an upsample operation followed by a convolutional operation, an instance normalisation layer and a final ReLU activation function. The upsample and convolution operation in L6, as well as layers D1 to D5 represent decoding layers that upsample the image (using a default k-nearest neighbour algorithm) by a factor of 2 and pass through a convolutional operation after. Layers D1 to D5 have in channels multiplied by 2 because of the skip connections present in the network.

Generator – CycleGAN	
E1	In channels= 2 ; Out channels= 64 ; Filter= 4 ; Stride= 2 ; Padding= 1
E2	In channels= 64 ; Out channels= 128 ; Filter= 4 ; Stride= 2 ; Padding= 1
E3	In channels= 128 ; Out channels= 256 ; Filter= 4 ; Stride= 2 ; Padding= 1
E4	In channels= 256 ; Out channels= 512 ; Filter= 4 ; Stride= 2 ; Padding= 1
E5	In channels= 512 ; Out channels= 512 ; Filter= 4 ; Stride= 2 ; Padding= 1
L6	Conv3D: In channels= 512 ; Out channels= 512 ; Filter= 4 ; Stride= 2 ; Padding = 1 Up sample: factor = 2 ; Conv3D: In channels= 512 ; Out channels= 512 ; Filter= 3 ; Stride= 1 ; Padding= 1
D1	In channels= 512*2 ; Out channels= 512 ; Filter= 3 ; Stride= 1 ; Padding= 1
D2	In channels= 512*2 ; Out channels= 256 ; Filter= 3 ; Stride= 1 ; Padding= 1
D3	In channels= 256*2 ; Out channels= 128 ; Filter= 3 ; Stride= 1 ; Padding= 1
D4	In channels= 128*2 ; Out channels= 64 ; Filter= 3 ; Stride= 1 ; Padding= 1
D5	In channels= 64*2 ; Out channels= 2 ; Filter= 3 ; Stride= 1 ; Padding= 1

Table A.13: Hyperparameters chosen to train the CycleGAN. Includes: batch-size, patch-size (size of the 3D patches to be sampled in the whole-image), initial learning rate, epoch decay (after how many epochs the learning rate starts to decay linearly to 0), the β parameter of the Adam optimiser chosen (for both the Generators and Discriminators), the λ values applied to the L1 loss, identity loss and anomaly loss.

Hyperparameters - CycleGAN	
Batch-Size	5
Patch-Size	64 x 64 x 64
Learning Rate	0.0002
Epoch Decay	1000
Adam Optimiser	β = (0.5, 0.999)
λ **L1 Loss**	0.001
λ **AM**	10
λ **Id**	0.5