FACULTY OF SCIENCE AND TECHNOLOGY

MASTER THESIS

Study programme / specialisation:
Computer Science

The spring semester, 2022

Open / ~~Confidential~~

Author: Håvard Godal

.....

Supervisor(s): Ketil Oppedal and Álvaro Fernández Quílez

Thesis title:
A Multi-class Dementia Classification Assessment Utilizing GAN to Convert MRI Scans from the 1.5T Domain to the 3.0T Domain

Credits (ECTS): 30

Keywords: Classification, CNN, Deep Learning, GAN, Generative Adversarial Network, Image-to-Image, MRI, Neural Networks, Pix2Pix.

Pages: 72

+ appendix: 14

Stavanger, June 14, 2022

University of Stavanger

Faculty of Science and Technology
Department of Electrical Engineering and Computer Science

A Multi-class Dementia Classification Assessment Utilizing GAN to Convert MRI Scans from the 1.5T Domain to the 3.0T Domain

Master's Thesis in Computer Science
by

Håvard Godal

Internal Supervisors

Ketil Oppedal

Álvaro Fernández Quílez

June 14, 2022

Abstract

Dementia is the seventh leading cause of death among all diseases and increases rapidly. With 10 million new cases every year, research is crucial for finding a treatment to cure dementia in the future. Magnetic resonance imaging (MRI) examination enables qualified professionals to analyze and detect discrepancies and anomalies in the brain. The quality and the signal-to-noise ratio (SNR) of MRI scans are directly proportional to the magnetic field strength used. For example, machines using a magnetic field strength of 3.0 Tesla (T) can generate scans with a higher SNR than magnetic field strengths of 1.5T and 0.5T but require more facilitation on the premises and considerable financial resources.

This thesis will explore how generative adversarial networks can improve the level of detail and SNR of MRI scans from the 1.5T domain to approach that of the 3.0T domain. GANs have proven to perform satisfactorily in similar scenarios, but only in binary classification tasks. This thesis investigates how the Pix2Pix GAN can be modified to use three-dimensional images. Furthermore, this thesis evaluates the performance through a multi-class convolutional neural network (CNN), classifying cognitively normal, mild cognitive impairment, and Alzheimer's disease.

An average performance measure of 0.84 and an average AUC score of 0.949 was achieved by classifying the generated 3.0T* MRI images, improving the evaluation of the 1.5T domain from 0.80 and 0.710, respectively. However, the small dataset size and the short training duration of the GAN could be limitations for the GAN performance. Nevertheless, this work presents a clear potential for increasing the SNR ratio for the 1.5T domain, which could be expanded to the 0.5T domain.

Acknowledgements

This thesis marks the end of my master's degree in Computer Science at the Department of Electrical Engineering and Computer Science at the University of Stavanger.

I would like to give a special thanks to my head supervisor Ketil Oppedal and my co-supervisor, Álvaro Fernández Quílez. They have provided me with valuable information and feedback throughout my work, something I am very grateful for.

Finally, I want to thank all lecturers and co-students for five years filled with joy and new knowledge. I would also like to thank ISI (the student organization for MSc students at the Department of Electrical Engineering and Computer Science) for organizing social activities and providing an excellent place for both knowledge sharing and fun.

Contents

Abbreviations

AD	**A**lzheimer's **D**isease
ADAM	**ADA**prive **M**oment estimation
ADNI	**A**lzheimer's **D**isease **N**euroimaging **I**nitiative
ANN	**A**rtificial **N**eural **N**etwork
AUC	**A**rea **U**nder the **C**urve
BCE	**B**inary **C**ross-**E**ntropy
BET	**B**rain **E**xtraction **T**ool
CDF	**C**umulative **D**istribution **F**unction
CGAN	**C**onditional **G**enerative **A**dversarial **N**etwork
CN	**C**ognitively **N**ormal
CNN	**C**onvolutional **N**eural **N**etwork
CT	**C**omputed **T**omography
DCGAN	**D**eep **C**onvolutional **G**enerative **A**dversarial **N**etwork
FID	**F**réchet **I**nception **D**istance
FLIRT	**F**MRIB's **L**inear **I**mage **R**egistration **T**ool
GAN	**G**enerative **A**dversarial **N**etwork
GPU	**G**raphics **P**rocessing **U**nit
MAE	**M**ean **A**bsolute **E**rror
MCI	**M**ild **C**ognitive **I**mpairment
MRI	**M**agnetic **R**esonance **I**maging
MSE	**M**ean **S**quared **E**rror
PET	**P**ositron **E**mission **T**omography
ROC	**R**eceiver **O**perating **C**haracteristic
SAR	**S**pecific **A**bsorption **R**ate
SGD	**S**tochastic **G**radient **D**escent
SSIM	**S**tructural **SIM**ilarity
T	**T**esla

Chapter 1

Introduction

1.1 Motivation

Dementia is an umbrella term describing a gradual decline in memory, thinking, and social abilities that is severe enough to interfere with daily functioning. Some dementia cases could also result in death if not treated. Alzheimer's disease (AD) accounts for 80 percent of all dementia cases, and the number of new AD cases is estimated to be around 360 000 per year [1]. Magnetic resonance imaging (MRI) can provide detailed three-dimensional brain scans of patients. MRI scans allow doctors and other qualified professionals to analyze the brain and detect discrepancies and anomalies compared to normal cognitive brains. The signal-to-noise ratio of a clinical MRI scan is directly proportional to the magnetic field strength used. Therefore, high-quality brain scans with a magnetic field strength of 3.0 Tesla(T) could improve dementia detection compared to lower magnetic field strengths.

Hospitals and institutions must facilitate the use of a 3.0T magnetic field strength. Strong magnetic fields require large areas to be allocated. Such machines also need patients to be transported to the scanner's location, which is not always feasible. The higher price of these machines compared to machines utilizing a lower magnetic field strength results in 3.0T MRI scanners not always being viable. Despite the downsides, a magnetic field strength of 3.0T has the potential to provide MRI data with an improved signal-to-noise ratio. Better SNR can lead to enhanced radiological assessment. A solution to avoid the downsides of utilizing a magnetic field strength of 3.0T is portable MRI scanners. Such portable scanners often use a weak magnetic field strength of 0.5T. Still, they are becoming increasingly popular because of their versatile use and low-cost [2]. Having a magnetic field strength of 0.5T compared to 3.0T reduces the SNR of MRI images and makes radiological assessment more difficult. This leads to the motivation of using deep

learning and generative adversarial networks (GAN) [3] to convert MRI scans from a lower domain to the 3.0T domain.

The Alzheimer's Disease Neuroimaging Initiative (ADNI) [4] has a database containing MRI scans of the same patient on the same visit for both the 1.5T domain and the 3.0T domain. As a result, the 1.5T domain is used instead of the 0.5T domain commonly found in portable scanners. Though different, the 1.5T domain serves the same purpose as the 0.5T in creating MRI images with a lower SNR and overall image quality, making radiological assessment more difficult.

1.2 Problem Definition

This thesis explores the limitations of assessing 1.5T MRI scans compared to 3.0T. The proposed solution involves using generative adversarial networks (GAN) [3] to improve the SNR, enabling a more accurate brain pathology and healthy tissue detection. GANs are a deep learning approach to generative modeling, using a generative and discriminative network to generate data. This thesis will conduct experiments using a dataset containing pairwise 1.5 Tesla and 3.0 Tesla scans from patients. The patients in the dataset are diagnosed with either cognitively normal brain functioning (CN), mild cognitive impairment (MCI), or dementia. The dataset consists of scans generated from several MRI machines and obtained through The Alzheimer's Disease Neuroimaging Initiative (ADNI) [4].

1.3 Outline

Chapter 1 - Introduction

Chapter one describes the outline of the thesis and a brief introduction to the structure of the work.

Chapter 2 - Background

The content of chapter two explains the workings of MRI and how brain imaging is assessed. The chapter also covers the dataset's different types of brain functionality and which tools and methods the thesis uses. Chapter 2 also explains how convolutional neural networks (CNN) and GANs work.

Chapter 3 - Solution Approach (Materials and Methods)

Chapter three revolves around the analysis and pre-processing of the dataset, which uses parts of FSL [5]. The chapter also covers configurations and adjustments regarding the GAN and the CNN classifier.

Chapter 4 - Experimental Evaluation (Experiments and Results)

Chapter four provides insight into the results obtained throughout the progress of the thesis. This includes both visual results and metric results from multi-class classification.

Chapter 5 - Discussion

The content of chapter five contains a discussion of the results and evaluations obtained throughout the thesis.

Chapter 6 - Conclusion and Future Directions

Chapter six concludes the thesis. The chapter also discusses limitations throughout the work and how future research in similar settings can overcome similar problems.

Chapter 2

Background

2.1 Dementia

Dementia is a syndrome used to describe several neurological conditions affecting the brain. Dementia is characterized by a gradual loss of cognitive and emotional abilities severe enough to interfere with daily functioning. The term is one of the most significant causes of dependency and disability among older people. Around 1 percent of all people aged 60 are diagnosed with dementia, and the percentage doubles every five years. Between 30 and 50 percent of people at the age of 85 are diagnosed with dementia [6]. Worldwide, WHO estimates that around 50 million people have some form of dementia, with 10 million new cases every year [7]. The cost of dementia-related causes is calculated to represent 1.1% of aggregated global GDP [8].

Despite the significant prevalence of cases, dementia is not a normal part of natural aging. An average person will have a prolonged memory decline, starting at an old age. Dementia diseases such as Alzheimer's disease have a more rapid cognitive decline, often starting earlier in life. There is currently no treatment available to cure dementia or reverse the progression of the syndrome [9]. Research and dementia diagnosis and detection are crucial for developing cures and treatments in the future. Ideas to aid this type of research include utilizing MRI scans with a high magnetic field strength of 3.0T or higher to provide images with a better signal-to-noise ratio and enable more people to get diagnosed by utilizing portable scanners.

The dataset used in this thesis consists of cognitively normal patients (CN), an early phase of dementia known as mild cognitive impairment (MCI), and Alzheimer's disease (AD).

2.1.1 Cognitively Normal State

Patients with no signs of dementia are included in the dataset. Including CN is necessary when performing multi-class classification and is useful when assessing the results of the generative adversarial network.

2.1.2 Mild Cognitive Impairment

Mild cognitive impairment (MCI) is commonly referred to as the stage between standard cognitive functionality and dementia diseases. Aging causes a natural cognitive decline, and dementia diseases such as Alzheimer's disease speeds up this process. MCI is the stage between these declines and is characterized by problems with language and reasoning. People with MCI are often aware that they have the disease, and the cognitive changes are not severe enough to interfere with everyday activities significantly. MCI may evolve to become Alzheimer's disease but could also stabilize and not change [10].

2.1.3 Alzheimer's Disease

Alzheimer's disease (AD) is a degenerative brain disease that causes atrophy in the brain and brain cells to die. The disease is the most common type of dementia, accounting for around 80% of all dementia cases. AD is characterized by a progressive decline in several cognitive domains, including language, personality, executive and visuospatial function, memory, and behavior [11].

Diagnosing a patient with Alzheimer's disease is based upon a clinical presentation based on fluid and imaging biomarkers and fulfilling several criteria. These criteria include beta-amyloid plaque deposition, and neurofibrillary tangles of hyperphosphorylated tau [11]. Another common trait revolves around the loss of connections between neurons in the brain.

There is currently no cure for Alzheimer's disease, though the research for treatments and development has had a significant process. Medications may temporarily slow the progression but not heal or remove the Alzheimer's disease symptoms. Some treatments revolve around drugs that regulate chemicals that transmit messages between neurons and help with behavioral issues but do not change the underlying disease problem.

2.2 Magnetic Resonance Imaging

Magnetic Resonance Imaging, also known as MRI, is a tool used to create three-dimensional anatomical images. MRI is a non-invasive method of mapping the internal structures within the body. It is often used to analyze and detect diseases, make a diagnosis, or monitor treatments.

MRI uses magnets that produce a strong magnetic field around a body part, such as the brain. This magnetic field forces protons in the brain to align with the field. A radiofrequency current can then be passed through the brain to make the protons spin out of equilibrium. The magnetic field is carefully controlled to produce cross-sectional images. The amount of time used before the protons are aligned with the magnetic field, and the energy released from the alignment is measured as signals [12]. Finally, a mathematical process known as a Fourier transformation is used to convert the signals from the frequency domain to the spatial domain, forming a three-dimensional brain model.

Another standard tool to create anatomical images is computed tomography (CT). Compared to CT, MRI is well suited for imaging soft tissues and body parts not containing bones. Also, MRI does not use damaging ionizing radiation of x-rays, which CT uses.

MRI can still be harmful, despite not using x-rays. This is because MRI uses a magnetic field that can affect magnetizable metals. This is important for people with iron implants, such as brain aneurysm clips, cardiac defibrillators, and pacemakers. In some cases, the MRI scan has to be performed with a lower magnetic field strength than desired, resulting in images with a worse signal-to-noise ratio. This is an important reason to develop software able to increase the signal-to-noise ratio of MRI scans to create an image with a higher magnetic field strength artificially.

2.2.1 Magnetic Field Strengths

Two common magnetic field strengths in clinical MRI scanning are 1.5 and 3.0 Telsa. The strength of the magnetic field used is directly proportional to the amount of signal received from the subject during a scan. This means that scans created using 3.0T will have a better signal-to-noise ratio, thus providing a more accurate and detailed analysis of the brain. The opposing sides of having a stronger magnetic field include pricing and safety. 3.0T imaging with stronger magnets can cause implants that would be safe in a 1.5T magnetic field to be unsafe, limiting the diversity of patients. Exposure to a strong magnetic field also increases the specific absorption rate (SAR), the estimated energy

rate absorbed by the body during an MRI scan. Increased SAR can be harmful as the heating of implants, and surrounding tissue may result in burns [13]. Machines capable of creating a magnetic field strength of 3.0T are often large, requiring hospitals and institutes to allocate large areas and considerable financial resources. This also requires patients to be transported to the location, which is not always feasible.

2.2.2 Portable MRI Scanners

Research in MRI technology has traditionally been around improving techniques and instrumentation to enable MRI scanners to generate more accurate images with a better signal-to-noise ratio. This type of research brings several favorable and disadvantageous features. Introducing low-cost and truly portable scanners could solve some of the disadvantages and enable new point-of-care and monitoring applications not feasible for large, expensive, and centralized 3.0T MRI machines [2]. General portable MRI scanners have seen advances in both speed and patient comfort. Some limitations regarding specialty scanners such as portable brain MRI scanners are hardware and computational technology. These limitations might diminish because of recent advances in those fields [2]. Another limiting factor is that smaller, portable machine produces a weaker 0.5T magnetic field strength than large stationary machines. This leads to scans with a comparatively lower signal-to-noise ratio, making dementia diagnosis a more complicated and inaccurate task.

This thesis uses a dataset with a magnetic field strength of 1.5T instead of 0.5T because ADNI does not provide 0.5T MRI scans with corresponding 3.0T MRI scans. Converting MRI scans from the 1.5T domain to the 3.0T domain utilizing generative adversarial networks can be further developed when portable MRI scanners become more popular.

2.3 Preprocessing

Preprocessing MRI images is critical in ensuring the success of any quantitative analysis pipeline. Such preprocessing can be composed of various operations to improve image quality or standardize geometric and intensity patterns. Clinical MRI scans of the brain contain some irrelevant information which can cause problems when using generative adversarial networks. For example, body parts such as the eyes and the neck of a patient do not provide useful information when evaluating the brain's structure and should therefore be removed. Machine learning algorithms are dependent on such preprocessing steps to detect patterns and information of interest. It is essential to balance too little and too much preprocessing. Too little will cause machine learning implementations to

learn patterns that do not contribute to the desired result. GANs risk learning the general shape and orientation of the MRI scan of a subject instead of the finer details inside the brain. At the same time, CNN classifiers might be biased if some head orientations belong in one class and another orientation belongs in another class. Too much preprocessing can cause important information inside the brain to be removed and lost. This can cause GANs to create too general images, not reflecting patterns that could be used to classify the MRI scan. Too much preprocessing can therefore lead to worse results than expected.

Removing unnecessary data from the MRI dataset is performed by extracting the brain from the clinical MRI scans. To avoid the GAN from learning false or undesired patterns, the preprocessing also includes steps to correct the MRI scans' bias field and register each image to a brain template, ensuring uniform rotation, position, and shear.

2.3.1 Brain Extraction

Extracting the brain from a clinical MRI scan consists of multiple complex operations performed several times to create a brain mask in the MRI scan. The mask is then used to extract the brain from the rest of the head, resulting in an MRI scan without skin, eyes, neck, and other non-interesting body parts.

2.3.2 Bias Field Correction

The bias field (also known as the intensity in-homogeneity) of MRI scans consists of artifacts arising from improper image acquisition. The condition of each scanner used to create the dataset is unknown. This can lead to discrepancies between some scanners [14]. Therefore, it is crucial to do a bias-field analysis and correct all images to ensure field similarities. Field dissimilarities could be an unwanted pattern machine learning systems could learn.

2.3.3 Image Registration

Increasing the quality of an MRI scan to match another scan requires two corresponding images to be as geometrically similar as possible. In some instances, the similarity is satisfactory, but the geometric position does not match in most cases. This includes 1.5T and 3.0T MRI scans of the same patient at the same visit. One factor responsible for this dissimilarity is that MRI scans belonging to the same visit could have been created multiple days apart. Having numerous days between two scans can result in different head orientations, and some movement might be present in one of the images.

A solution to the dissimilarity problem is to perform image registration on all MRI scans. Image registration is a method used to ensure the spatial correspondence of anatomy across images. In the preprocessing step, all MRI scans are registered to the MNI152 template as it is the preferred template choice when working with brain MRI scans [15]. This ensures that the orientation and scale are equal by performing affine transformations on all MRI scans.

2.3.4 Histogram Matching

Histogram matching is used to match the voxel intensity of an image to that of another image, resulting in the same intensity distribution. Minor intensity artifacts might arise from the preprocessing steps described above. Histogram matching is conducted to ensure that the voxel intensity distribution of the 1.5T and the 3.0T MRI scans does not cause any problems, thus preventing these artifacts from interfering with the GAN training.

2.3.5 Augmentations

Each MRI image contains a lot of dark voxels surrounding the head. These voxels result from the air between the MRI machine and the head of the patient and do not provide any information regarding the brain. Still, they can not be removed without removing valuable information by cutting parts of the brain. Therefore, all MRI scans have a similar structure with dark voxels around a region of bright voxels (the brain). This can cause the GAN to focus on the more apparent features of the scans, limiting the capabilities for learning the more intricate parts of the brain. This problem is circumvented by applying transformations to each set of MRI scans, thus varying how each image is structured. The transformations used include flipping the MRI scans randomly in all directions and transforming the MRI scans in a random direction.

2.4 Software

2.4.1 FSL

The Functional Magnetic Resonance Imaging of the Brain (FMRIB) Software Library, abbreviated FSL, is a software library containing tools for MRI brain imaging analysis and modification. FSL provides both graphical GUI and command-line interactions, allowing FSL to be integrated into other procedures and tasks [5].

2.4.2 Python

Python is an interpreted, object-oriented, high-level programming language with access to a vast library ecosystem for machine learning and image visualization. The experiments conducted in this thesis utilized multiple libraries. Some of the most important ones are listed below.

Nibabel

Nibabel is a Python package that provides read and write access to some standard medical and neuroimaging file formats such as NIfTI (Neuroimaging Informatics Technology Initiative). In addition, Nibabel stores both the MRI scan matrix and the corresponding metadata, making the package helpful in creating datasets.

Nipype

Nipype is an open-source initiative that provides an interface to existing neuroimaging software such as FSL. This package provides a streamlined pipeline for external software processing MRI images.

PyTorch

PyTorch is an open-source machine learning framework built on Torch's scientific computing framework. PyTorch provides tensor computing on graphics processing units (GPU) and is optimized for deep learning.

Wandb

Wandb, short for Weights & Biases, consists of lightweight, interoperable tools to track experiments, evaluate model performance, and visualize results.

2.5 Neural Networks

Artificial neural networks (ANN) are a subset of machine learning and are central to deep learning. The ideology is inspired by the human brain, mimicking how the brain operates. Neurons in artificial neural networks operate the same way the brain works by

propagating a signal throughout the network [16]. ANNs are composed of nodes clustered inside an input layer, an output layer, and some hidden layers.

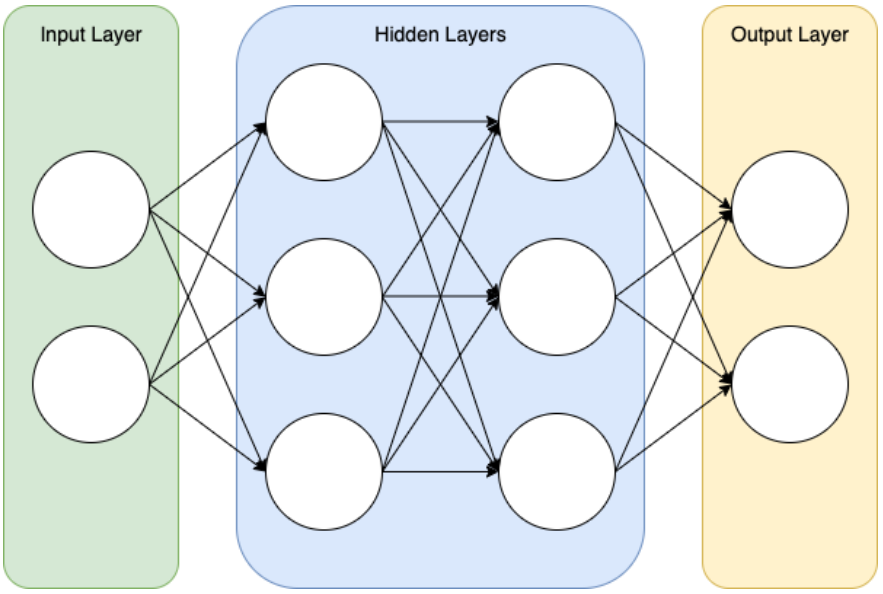

Figure 2.1: Simplified structure of a fully connected ANN.

Each node, or artificial neuron, has a linear regression model composed of some assigned weight matrix and bias to the node. Each node uses the input data matrix combined with the weight and bias to compute the output value of that node. This output is passed through a non-linear activation function. This is important because the whole neural network could be condensed into a single linear function without the activation function. The activation function determines how the node should be activated.

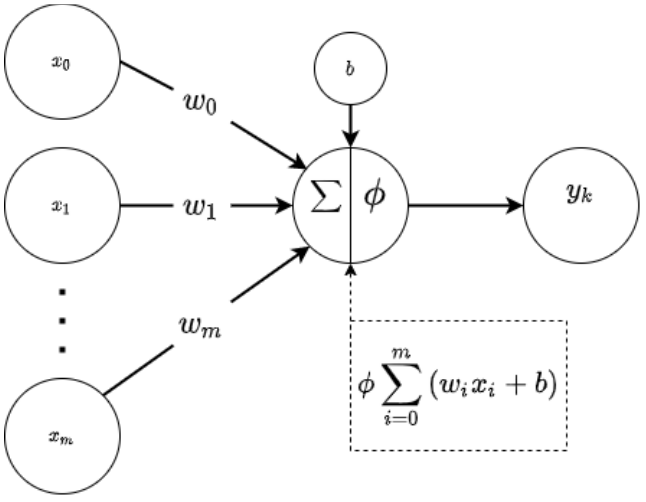

Figure 2.2: A simple structure of an artificial neuron with a linear regression model.

ANNs are often initiated with some random values as weights to the nodes. Forward propagation uses these weights to create an output of the neural network, while back propagation learns from the results of the forward pass and adjusts the weights to improve

the neural network. This is done by the backward propagation calculating a value known as loss through a loss function 2.5.2. The loss is interpreted by an optimizer function 2.5.3 that updates the nodes in the network with optimized weights.

2.5.1 Activation Functions

Activation functions transform the weighted sum of the inputs in a neural node into an output ranging between some values. Most of such activation functions are non-linear. This is important to avoid being able to collapse a neural network into a single linear regression problem.

Sigmoid

The sigmoid activation function takes any value as input and returns a value between 0 and 1. It is often used for classification as the output resembles the probability space.

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2.1}$$

Tanh

The tanh activation function resembles the shape of the sigmoid activation function but has an output range between -1 and 1. Such activation functions are helpful inside neural nets as the mean of the output will be close to 0, which helps keep the data normalized.

$$f(x) = \frac{(e^{x} - e^{-x})}{(e^{x} + e^{-x})} \tag{2.2}$$

ReLU

The rectified linear unit (ReLU) differs from the other activation functions by being constructed of two linear functions. The output from the ReLU ranges between 0 and ∞. ReLU is a typical activation function to use inside neural networks because of the simple gradient computation. ReLU does not need any multiplication, no exponential, and no multiplication and division. One of the properties of ReLU is that some neurons can die, meaning that a gradient update weights the neuron so that the output of the activation function is always zero. This can lead to undesired results in some cases. ReLU also combats a neural network problem known as the vanishing gradient problem. The

vanishing gradient problem is that derivatives approach zero as the number of repeated uses approaches infinity. ReLU avoids this issue as the derivative for any positive input value is 1.

$$f(x) = max\,(0, x) \tag{2.3}$$

LeakyReLU

Another commonly used version of ReLU is the LeakyReLU, which solves the dead neuron problem that can arise in the normal ReLU activation function. LeakyReLU modifies ReLU by introducing a non-zero linear equation for negative and positive input values. This results in an output range between $-\infty$ and ∞.

$$f(x) = max\,(0.1x, x) \tag{2.4}$$

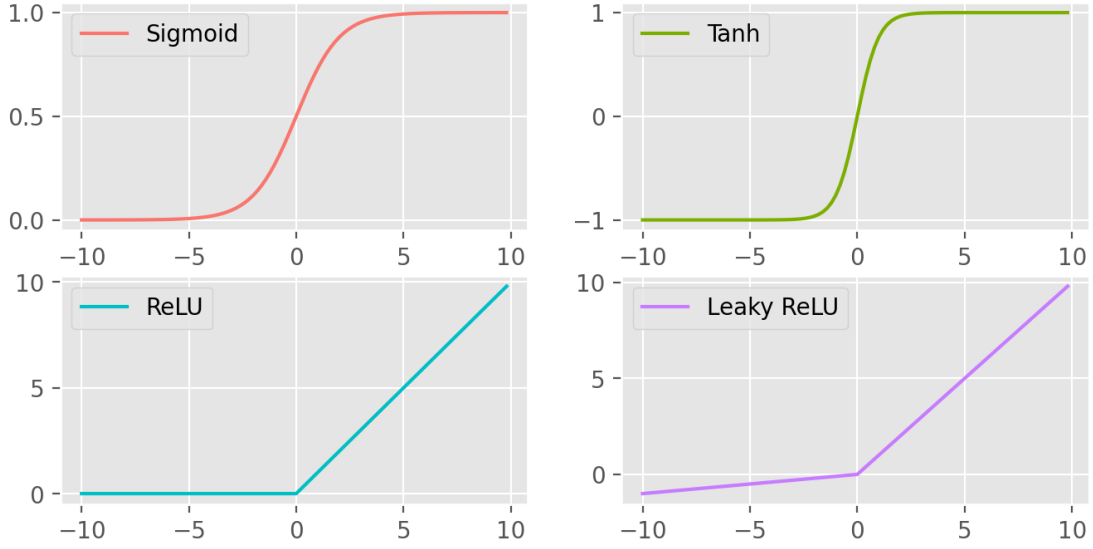

Figure 2.3: Activation functions.

2.5.2 Loss Function

A loss function is responsible for creating a score of the expected outcome compared to the produced outcome of a neural network. Such a loss function score can further be used in an optimizer 2.5.3 to calculate the gradients required to train the neural network through back propagation. Different loss functions suit different types of neural networks. Some of these include Mean Absolute Error (MAE), Mean Squared Error (MSE), and Binary Cross-Entropy (BCE).

MAE

MAE uses the L_1 norm to calculate the difference between each element between the expected outcome y and the produced outcome $\hat{y}$. This type of loss function weights all error values equally.

$$\mathcal{L}_{MAE}(y, \hat{y}) = \frac{1}{N} \sum_{i=1}^{N} |y_i - \hat{y}_i| \tag{2.5}$$

MSE

The MSE criterion measures the squared L_2 norm between the expected and the produced outcome. MSE can be used when the data has a Gaussian distribution centered around some value and when it is vital to penalize outliers. By squaring the errors, significant errors will be weighted more.

$$\mathcal{L}_{MSE}(y, \hat{y}) = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \tag{2.6}$$

BCE

Binary cross-entropy (BCE) is based on entropy, which measures the uncertainty associated with a given distribution. Compared to categorical cross-entropy, BCE is binary and implies that the data must be categorized into two categories. BCE uses a probability mass function in its core and computes the log probability of a sample belonging to each of the two distributions. The log probability is used because it is a monotonically increasing/decreasing function. This makes maximizing/minimizing less computationally heavy. BCE assumes a binomial distribution and is often used in binary classification problems, while cross-entropy is used in multi-class classification problems.

$$\mathcal{L}_{BCE}(y, \hat{y}) = -\frac{1}{N} \sum_{i=1}^{N} (y_i \log \hat{y}_i + (1 - y_i) \log (1 - \hat{y}_i)) \tag{2.7}$$

2.5.3 Optimizer

It is difficult to calculate the perfect weight used in a neural network because of the complexity and non-linearity. Instead, finding the best weights becomes an optimization problem. This includes some algorithm that searches the space of possible weights the

model can use to increase the model's performance. This optimization search is often known as gradient descent. The gradient descent algorithm changes the weights used in the model based on the error of the model's predictions. The algorithm wants to reduce the error, resulting in the optimization algorithm navigating down the error gradient.

The two most common optimizers used in deep neural networks are the Stochastic Gradient Descent (SGD) and the Adaptive Moment Estimation (ADAM).

SGD

SGD moves in the opposite direction of the steepest ascent in the error space to minimize the error. The optimizer depends on the loss function's derivatives to find a minimum. The optimizer updates the model weights after each calculation, separating SGD from regular gradient descent [17].

$$\theta = \theta - \eta * \nabla_\theta J\left(\theta; x^{(i)}; y^{(i)}\right) \tag{2.8}$$

Here J denotes the loss function, and η denotes the learning rate. The learning rate is a hyperparameter defining how much the computed gradient affects the weights. A low learning rate can yield a lengthy training process that could get stuck in sub-optimal weights. Conversely, a significant learning rate can cause the model to struggle to converge to a minimum.

ADAM

The ADAM optimizer operates with first and second-order momentum values. The optimizer computes the adaptive learning rates for each parameter. ADAM stores both an exponentially decaying average of past squared gradient (v_t) and an exponentially decaying average of past gradients (m_t) [17]. This reduces the oscillations often created by other optimizers such as the SGD.

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1)\, g_t \tag{2.9}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2)\, g_t^2 \tag{2.10}$$

Both v_t and m_t are initiated with a value of zero. This causes the optimizer to be biassed towards zero. Computing bias-corrected first and second-moment estimates avoid this.

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \tag{2.11}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \tag{2.12}$$

This yields the ADAM update formula:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon}\hat{m}_t \tag{2.13}$$

2.5.4 Instance and Batch Normalization

Instance normalization is a technique used to normalize the values of a neural network. Normalization layers prevent instance-specific mean and covariance shifts, simplifying the learning process. Furthermore, instance normalization normalizes each batch element independently, i.e., across spacial locations. This causes the distribution of each sample to be more Gaussian, but not jointly. On the other hand, batch normalization normalizes activations across a batch of values. Batch normalization is standard among CNN and fully connected neural networks, but instance normalization has been shown to improve the performance of specific deep neural networks for image generation [18].

2.5.5 Dropout

Neural networks tend to overfit if the training dataset consists of few samples. Overfitting occurs when a neural network learns too many details of a specific dataset. This results in the network learning both information and noise in the dataset, which leads to inferior results on other datasets. A technique to limit overfitting is to introduce dropout layers to the network. Dropout randomly prevents a signal from passing through a node. This dynamically alters the network structure, thus improving network regularization.

2.5.6 Evaluation

There exist multiple ways of evaluating the performance of a neural network. Calculating the receiver operating characteristics (ROC) curve is a common performance measurement for binary classification problems, which can be adapted to a multi-class problem by analyzing each class in a "one versus the rest" manner. The ROC curve is constructed by using the true positive rate and the false positive rate. This corresponds to a trade-off

analysis between the sensitivity and the recall. The area under the ROC curve (AUC) can be calculated together with the ROC curve and provides an overall value of how well the classifier distinguishes between classes.

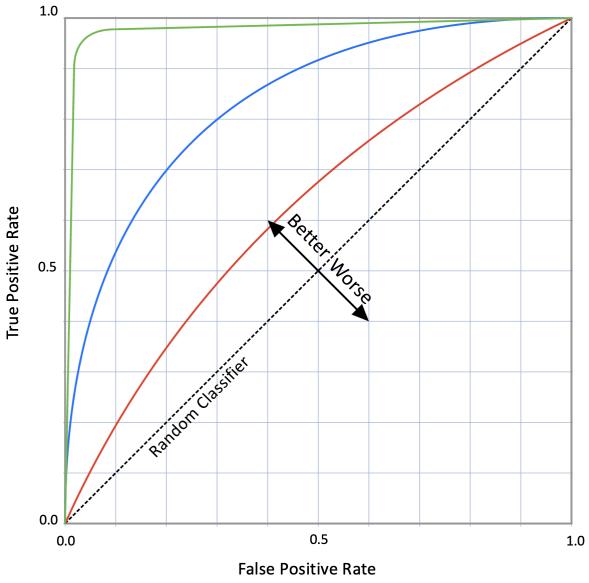

Figure 2.4: Three ROC Curves with varying AUC compared to a randomly guessing the binary distribution.

Some evaluation methods for classification problems can be derived from a corresponding confusion matrix. A confusion matrix is a performance measurement for machine learning, providing an overview of how accurate a model regards predicted and actual values.

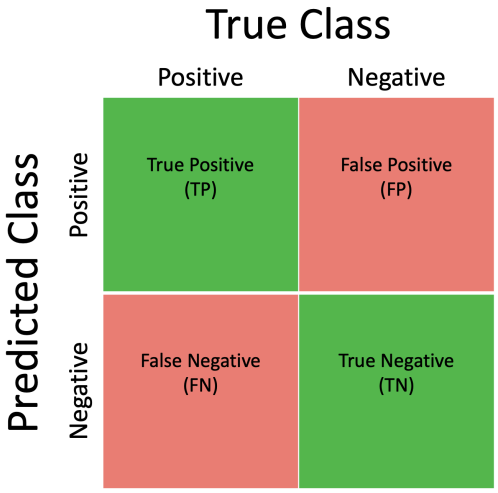

Figure 2.5: Confusion Matrix.

A confusion matrix summarizes and visualizes the performance of a classifier. Some performance metrics such as accuracy, precision, recall, and f1-score can be derived from the confusion matrix, representing the classifier's performance in other ways.

Accuracy

Accuracy is the proportion of true results among the total number of samples examined. Accuracy is helpful for datasets that are well balanced and not skewed towards a class.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{2.14}$$

Precision

Precision provides essential information when the true predictions are true is of high importance. This means that precision measures how exact the classification task is.

$$Precision = \frac{TP}{TP + FP} \tag{2.15}$$

Recall

Recall provides a metric for the portion of actual positives that are correctly classified. This evaluation method provides essential information when it is crucial to avoid classifying true positives as negatives.

$$Recall = \frac{TP}{TP + FN} \tag{2.16}$$

F_1

F_1 score is the harmonic mean between precision and recall. This is a useful evaluation method as precision punishes false positives, while recall punishes false negatives. The F_1 score gives equal weight to recall and precision.

$$F_1 = 2\frac{Precision * Recall}{Precision + Recall} = \frac{2TP}{2TP + FP + FN} \tag{2.17}$$

2.6 Convolutional Neural Networks

Convolutional Neural Networks (CNN) separate themselves from other neural networks by being designed around image and audio inputs. ANN uses one input node for each input value. This means that given a `512` pixel wide and `512` pixel tall input image, the

ANN will have 262144 input nodes. As a result, overfitting can occur. Also, spacial information is lost in ANN as the image structure is flattened to pass the data into the network.

This problem is solved with CNN. The size of the network determines the complexity of the patterns the CNN can learn. The first layers in a network often learn simple features such as distinct edges and colors. Then, the last layers can find patterns in larger, more complex areas, leading to a classification result of the whole image. CNN mainly consists of thee types of layers along with activation functions; the convolutional, the pooling, and the fully-connected layer.

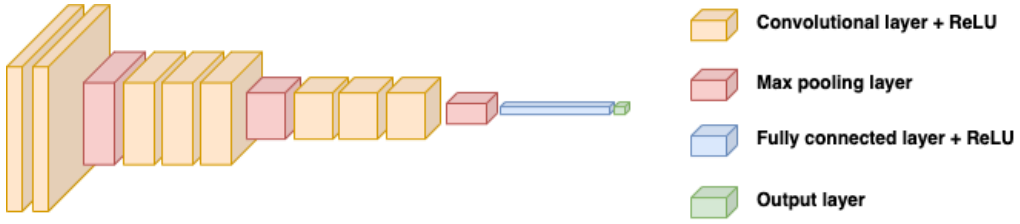

Figure 2.6: Architecture of a CNN classifier.

Convolutional Layer

The convolutional layer applies a kernel to the input image. The kernel is often initiated with random values, then updated and adjusted through back propagation to detect relevant patterns and features better. The kernel consists of a matrix with the same or fewer dimensions as the input matrix. The shape of the kernel (k) is commonly found to be an odd value between 3 and 7 in all dimensions. The kernel is applied through a dot product on an input area. The kernel is then shifted across the input with a given stride (s), determining how large each shift is. Finally, the kernel sweeps the whole input matrix, generating the feature map matrix.

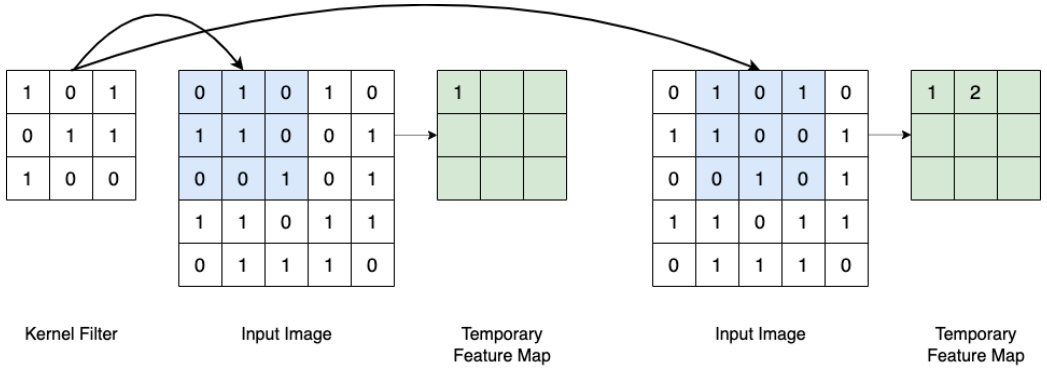

Figure 2.7: Convolutions with a kernel size of 3, a stride of 1 and 0 padding.

Padding (p) can also retain the input's original shape or obtain another output shape. This includes adding pixel values around the image. Standard techniques include padding the input with a constant or mirroring the input matrix outwards. The output shape of the convolution can be found through equation 2.18.

$$n_{out} = \frac{n_{in} + 2p - k}{s} + 1 \tag{2.18}$$

Transposed Convolution

Transposed convolution layers are the opposite of regular convolutional layers. Unlike an encoder, transparent convolutional layers increase the spatial dimension instead of reducing it.

$$n_{out} = (n_{in} - 1)s + k - 2p \tag{2.19}$$

Pooling Layer

The pooling layer is responsible for reducing the number of parameters in the model. This is known as downsampling or dimensionality reduction. The pooling layer operates similarly to the convolutional layer by sweeping the input with a kernel. The difference is that the kernel used in the pooling layer does not have weights. Instead, the kernel applies an aggregation function with some given shape and stride.

Some of the most common pooling techniques are max pooling and average pooling. The largest value in each respective field is used as output values with max pooling. Average pooling calculates the average value within the respective field and uses the calculated values as output values.

Much information is lost when downsampling the input. Despite this, the pooling layer reduces complexity and limits the risk of overfitting the model to the training data.

Fully-connected Layer

Fully-connected layers are used to flatten the data before classification. Flattening the data removes the spatial information, so fully-connected layers are often used at the very end of CNN. An alternative to a fully-connected layer is to use a convolutional layer with a filter size equal to the input size. Having the filter size equal to the input size results in only one convolutional computation, resulting in the desired output of only one value.

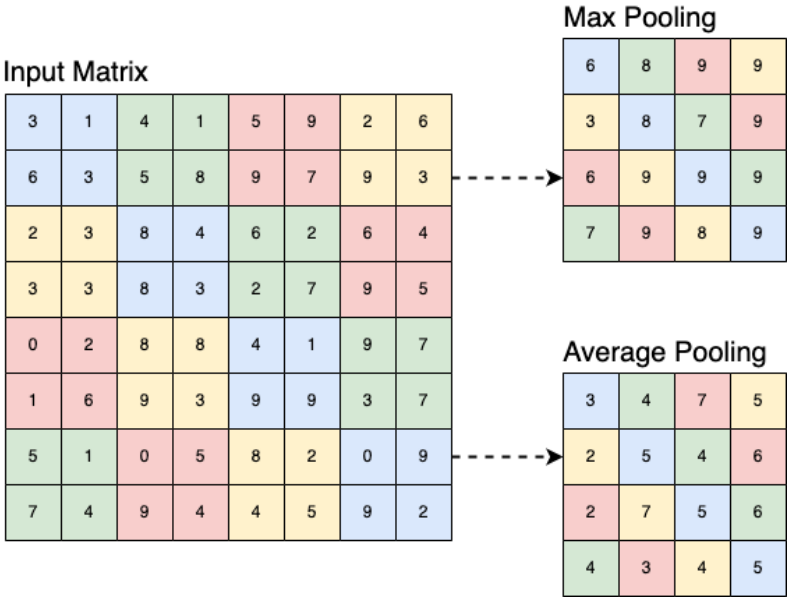

Figure 2.8: Max and average pooling with a kernel shape of 2 and a stride of 2.

2.7 Generative Adversarial Networks

The generative adversarial network (GAN) is an adversarial net framework proposed by Ian Goodfellow in 2014 [3]. GANs consist of a generative model being pitted against an adversary. Adversaries are discriminative models that learn to distinguish between samples from a dataset and samples generated by the generative model. The generative model can be compared to a painter trying to create a replica of famous paintings. The discriminative model is analogous to an art expert trying to distinguish original paintings from replicated paintings created by the generative model. A competition to create better painting replicas and distinguish between the paintings more accurately drives both parties to improve their methods until the differences between the paintings are indistinguishable.

In other terms, the discriminator $D(x)$ can be associated with a traditional binary classifier. Here x is an input sample from the dataset. The discriminator network outputs a scalar probability of the sample coming from the dataset instead of being generated by the generator. The generator $G(z)$ represents a network that maps the latent vector z to data-space. The generator's goal is to estimate the distribution of the dataset ($p_{dataset}$) to generate samples to the estimated distribution p_g. The discriminator and the generator are part of a min-max game where the discriminator tries to maximize the probability of correctly classifying real and generated samples ($\log(D(x))$). In contrast, the generator tries to minimize the probability that the discriminator correctly classifies the generated samples ($\log(1 - D(G(z)))$) [3]. This leads to the GAN loss function 2.20.

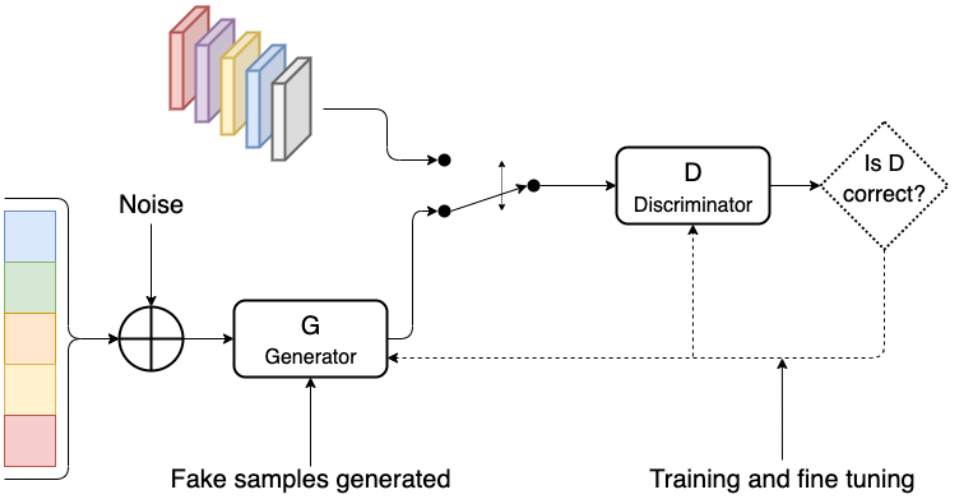

Figure 2.9: The original GAN structure.

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}\left[\log(D(\boldsymbol{x}))\right] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}\left[\log(1-D(G(\boldsymbol{z})))\right] \tag{2.20}$$

The theoretical solution to this equation is when $p_g = p_{data}$. This implies that the discriminator cannot distinguish samples from the dataset from the generated samples.

The original GAN structure uses ANNs for both the generator and the discriminator. However, fully connected neural networks diminish the quality of generated images as these neural networks lack information on the spatial structure. This is a common problem for ANNs and can be solved using CNNs, as they can learn hierarchical features by preserving spatial structures.

2.7.1 Deep Convolutional GAN

Deep Convolutional Generative Adversarial Networks (DCGAN) improve the original GAN structure by implementing convolutional layers to the networks [19]. Using convolutional layers allows a neural network to retain the spatial information of a sample. Another advantage of using convolutional layers is the elimination of pooling layers. Layers such as max-pooling layers have no learnable parameters and can not be trained to retain critical information.

DCGAN utilizes traditional convolutional layers in the discriminator in the same way CNN classifiers work. In addition, the generator uses transposed convolutional layers to decode the latent vector and increase the spatial dimension to the desired shape. Other methods available for decoding or upsampling latent vectors include interpolation techniques. The

advantage transposed convolution has over interpolation is that interpolation techniques lack learnable parameters. Without learnable parameters, the decoding layers cannot be trained to learn how to generate images of a given domain.

2.7.2 Conditional GAN

Conditional Adversarial Networks (CGAN) is a supervised type of GAN, conditioning both the generator and the discriminator on external information [20]. This type of external information could be labels or data from other modalities. The generator is parameterized to learn patterns associated with each given condition, while the discriminator learns to distinguish real and generated samples apart. The discriminator also evaluates the input based on the same condition used in the generator. The generator relies on both latent space and a condition to generate samples, corresponding to 2.21. The task of the discriminator also changes compared to DCGAN as it has to take the external information into account when evaluating each sample.

$$G(z, y) = x|y \tag{2.21}$$

2.7.3 Pix2Pix

Paired Image-To-Image translation (Pix2Pix), proposed in 2017 [21], deviates from the previously mentioned generative adversarial networks. Pix2Pix translates samples from one domain to another by learning the mapping between the two domains. As such, Pix2Pix does not use latent space to generate samples in the generator. Pix2Pix builds upon CGAN, but instead of relying on a condition and a latent vector, Pix2Pix solely relies on an input sample and a target sample. Pix2Pix also uses a more delicate generator and discriminator, consisting of a U-Net generator and a Patch-GAN discriminator.

U-Net Generator

Earlier GAN implementations used a generator structured as a decoder to decode a latent vector into a sample. Pix2Pix does not use latent vectors or such a generator architecture. Instead, Pix2Pix uses an encoder-decoder structure known as a U-Net.

The U-net used in Pix2Pix has an encoder-decoder network architecture. The encoder reduces the spatial dimensions while increasing the number of features. This process learns an abstract representation of the input image. Each step of the encoder part of the U-net consists of a convolutional layer, an instance normalization layer, and

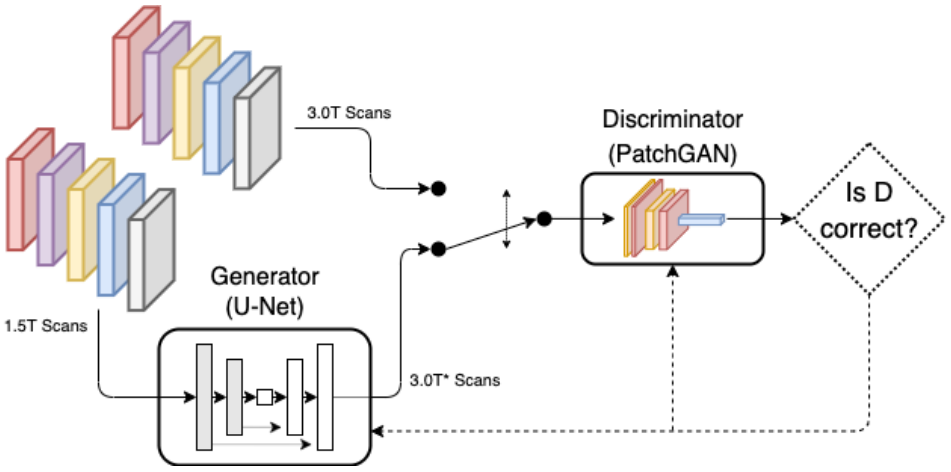

Figure 2.10: The structure of Pix2Pix.

a LeakyReLU activation function. The output from the activation function is also used as a skip connection between the encoder layer and the corresponding decoder layer. Skip connections improve the flow of gradient in backpropagation, resulting in the U-net learning a better representation of the domain. Skip connections also provide the decoder with features that can be lost due to the depth of the network [22]. The abstract representation created by the encoder can be compared to the latent vector used in DCGAN and CGAN. The decoder uses the abstract representation of the input to generate an output in the desired domain. Compared to the encoder, the decoder architecture consists of transposed convolution layers instead of regular convolution layers and ReLU instead of LeakyReLU. Before each transposed convolution, the input of the current layer is concatenated with the corresponding skip connection feature map from the encoder block. The final layer of the decoder uses a Tanh activation function before returning the output of the U-net [21].

PatchGAN Discriminator

Discriminators in GANs such as CGAN and DCGAN use CNNs that evaluate the whole input sample, returning a single probability value indicating whether the sample is real or generated. Pix2Pix utilizes another approach called a PatchGAN. Patch-GAN only penalizes the structure of a sample at the scale of local patches instead of evaluating each pixel. Therefore, PatchGAN effectively models the input sample as a Markov random field, assuming independence between voxels separated by more than a patch diameter [21]. Like normal CNNs, a Patch-GAN uses convolutional layers, instance-normalization layers, and LeakyReLU activation functions. The main difference is that the output of a PatchGAN is a feature map of predictions that can be averaged to form a single score.

2.8 Previous Work

Earlier research has illustrated how GANs can be used to generate new MRI scans of the brain, thus increasing the size of the dataset [23]. The preprocessing steps used to generate new MRI scans have been adapted and further developed in this thesis. GANs have also been used to enhance MRI images to classify Alzheimer's disease, and cognitive normality [24]. This thesis expands upon this idea by classifying cognitive normality, mild cognitive impairment, and Alzheimer's disease. Furthermore, this thesis explores using Pix2Pix with a U-net generator and a PatchGAN discriminator instead of regular CNNs to convert MRI scans from the 1.5T domain to the 3.0T domain.

Other works show favorable results in using Pix2Pix together with MRI. For example, the paper *"Inferring PET from MRI with Pix2Pix"* [25] concludes that Pix2Pix can sufficiently and reliably generate positron emission tomography (PET) scans from MRI scans. Pix2Pix is a promising and potentially cost-saving method for creating PET scans.

The paper *"Generative Adversarial Networks for Image-to-Image Translation on Multi-Contrast MR Images"* [26] implements the use of Pix2Pix for generating T1 and T2 MRI images from CT images. The authors state that the GAN used in the paper can generate MRI scans visually indistinguishable from real MRI scans. However, the paper does not include any metrics, making any comparison challenging to achieve. Similarly, the paper "MRI Cross-Modality Image-to-Image Translation" [27] utilizes GANs to change the modality of MRI scans, capable of generating MRI scans with modalities of T1, T2, T2-Flair, and PD. This paper also shows encouraging results in cross-modality registration among some widely adopted brain datasets. Using a GAN named "CGAN + L1" to generate T2 MRI images from PD MRI scans and vice versa yielded a structural similarity score of 0.89 and 0.88, respectively, compared to real MRI scans.

Chapter 3

Solution Approach

The development throughout this master's thesis has revolved around gradually implementing new ideas and features. The final solution has been copied from the original development platform and can be accessed through GitHub [1].

3.1 Dataset

The dataset used in this thesis consists of T1 weighted clinical MRI scans of the brain from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database. An application has to be submitted to get access to the data and samples contained in the database. The dataset used to conduct experiments in this thesis was obtained by introducing filters to the ADNI database to only extract T1 3D scans. The clinical MRI scans were downloaded as Digital Imaging and Communications in Medicine (DICOM) files and stored in a UNIX cluster at the University of Stavanger. The DICOM files were converted to NIfTI files, a file format with broader support among Python libraries.

3.2 Preprocessing

3.2.1 Dataset filtering

Two detailed subject overview files containing information for some relevant subjects were generated from the ADNI database. The first file contained an overview of the subject identification and the time point of the visit. The downloaded dataset discussed in section 3.1 had to be filtered to match the content of the overview file. The other

[1] https://github.com/HGodal/master-thesis

file provided the current diagnosis of some of the patients at the time of the visit. The dataset had to be filtered to only contain visits documented in this file, and the diagnosis information had to be included in the dataset.

3.2.2 Brain Extraction

The brain is extracted from the clinical MRI scans using the Brain Extraction Tool (BET) from FSL [28]. BET first uses a histogram-based threshold estimation. The result is used to compute the center of gravity and a spherical surface initialization of the brain region. This spherical surface is a mesh of connected triangles. For each iteration, the triangles are subdivided, and the surface is expanded. Finally, the spherical surface is deformed based on local intensities, resulting in an estimated brain region after the final iteration.

The Brain Extraction Tool has a threshold setting that had to be adjusted based on the quality of the MRI scans to generate the best brain extraction result. This fractional intensity threshold is used to determine where the edge of the brain is located, ranging between 0 and 1. Finding the best threshold was done manually by using a range of threshold values on a random set of scans multiple times to ensure that minimal excess data was kept while keeping as much of the brain intact as possible.

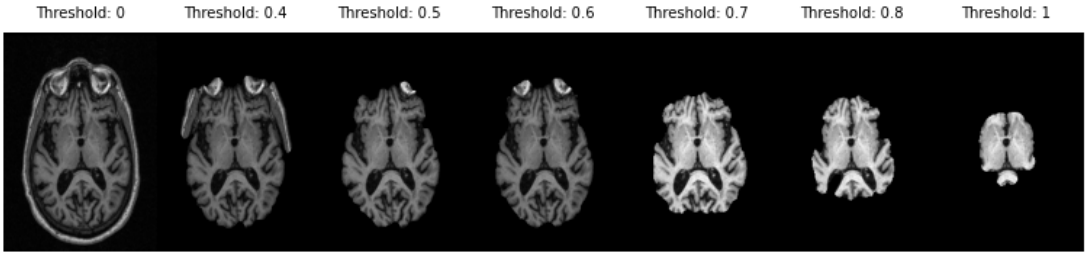

Figure 3.1: Brain Extraction over multiple threshold values.

Based on the analysis of the result of the fractional intensity threshold, the desired result was created with a threshold of `0.7`, with `0.5` being the default threshold value of the tool.

BET also includes an option for using a more robust brain center estimation compared to the standard configuration. This option repeatedly runs BET, adjusting the starting center of the brain estimation each iteration. This adjustment is set to use the center of gravity of the previously extracted brain. The iteration process stops after ten iterations or when the center of gravity stops moving. This approach prevents the tool from wrongly assuming other body parts such as the neck to be part of the brain, thus improving the brain extraction result.

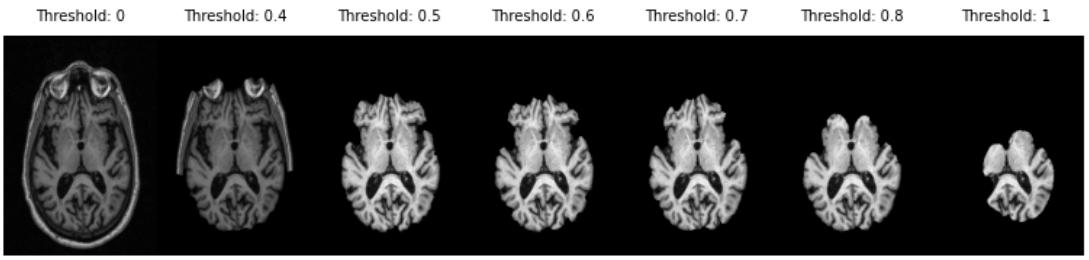

Figure 3.2: Brain Extraction over multiple threshold values with robust estimation.

Using the robust brain center estimation improves the result of BET. A clear improvement can be observed by comparing figure 3.1 and 3.2, with figure 3.2 suggesting a threshold value of `0.5` being sufficient. Upon further analysis of the dataset, `0.5` is too low to achieve sufficient brain extractions on all scans. Increasing the threshold value to `0.6` fixes this issue. Including the robust brain center estimation drastically increases the computation time for each brain extraction but improves the result.

Not all brain extraction procedures produced a satisfactory result. Example of a satisfactory and unsatisfactory result can be seen in figures 3.3 and 3.4. Unsatisfactory brain extractions were removed by evaluating each pair-wise scan's structural similarity and mean squared error. See section 3.2.5 for more details.

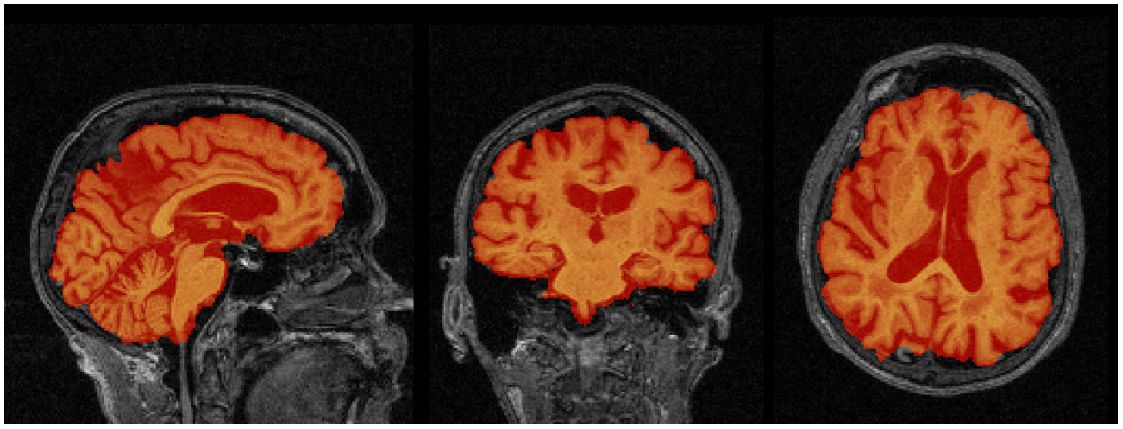

Figure 3.3: Satisfactory brain extraction.

3.2.3 Bias Field Correction

The ADNI dataset used in this thesis consists of scans created by multiple scanner systems such as General Electric medical systems and Philips medical systems. The condition of each scanner used to create the dataset is unknown, which can lead to discrepancies between some scanners. Therefore, it is essential to do a bias field analysis and correction of all images to ensure field similarities. Bias field correction will also aid in generating 3.0T images from 1.5T Tesla images. This is because MRI scans often have

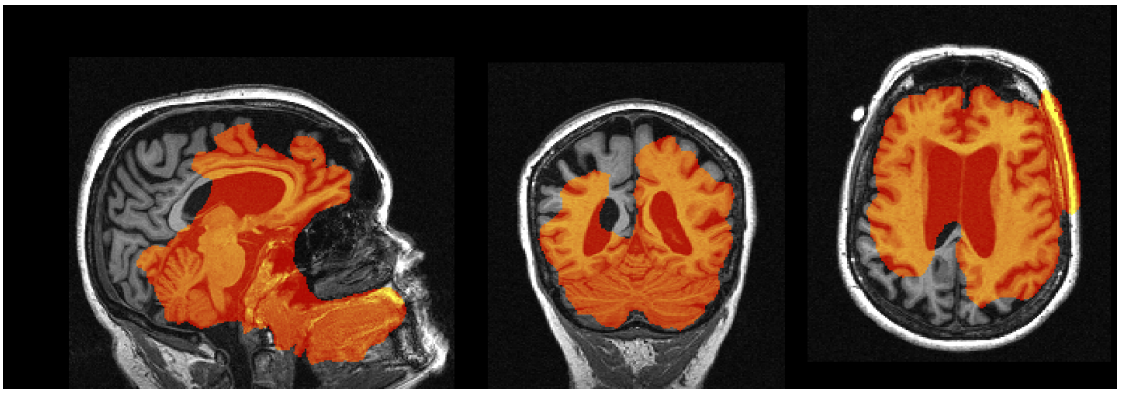

Figure 3.4: Unsatisfactory brain extraction.

a non-uniform intensity field which can arise when the field coils used in a scanner system are imperfect or other interferences. Such magnetic field variations can be mistaken for anatomical differences and should be avoided.

FAST is FMRIB's Automated Segmentation Tool, and one of the functionalities of FAST is to correct spatial intensity variations [29]. This is also known as bias field correction. The tool is fully automated and robust, capable of creating bias field corrected images without being sensitive to noise. FAST has several configurations to adjust but comes preset with initial values thought to be the best settings for T1-weighted brain MRI scans. Comparing the output of multiple parameter configurations resulted in similar images, which justified the choice of using the default FAST configuration values for the dataset.

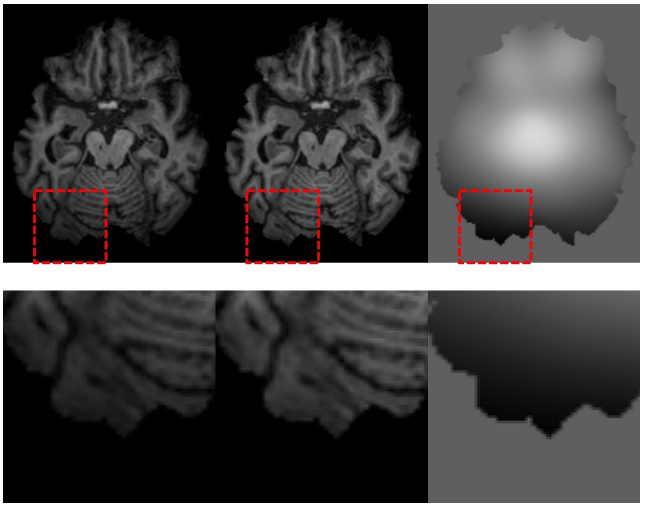

Figure 3.5: The original extracted brain (left) compared to the bias field corrected brain (middle), and a visualization of the bias field itself (right).

The result from FAST can be seen in figure 3.5. The bias field visualization shows a clear bias towards the back of the brain.

3.2.4 Image Normalization and Registration

Another important MRI scan process technique is to perform a registration of each scan. Such a registration transforms 3D volumes into the same space by performing affine transformations to the volume. Registering all MRI scans to the MNI152 standard-space T1-weighted average structural template image is one approach. The MNI152 template is derived from 152 structural images, averaged together after high-dimensional nonlinear registration into a common coordinate system. This template is integrated into FSL, signifying the practicality and the versatility of MNI152.

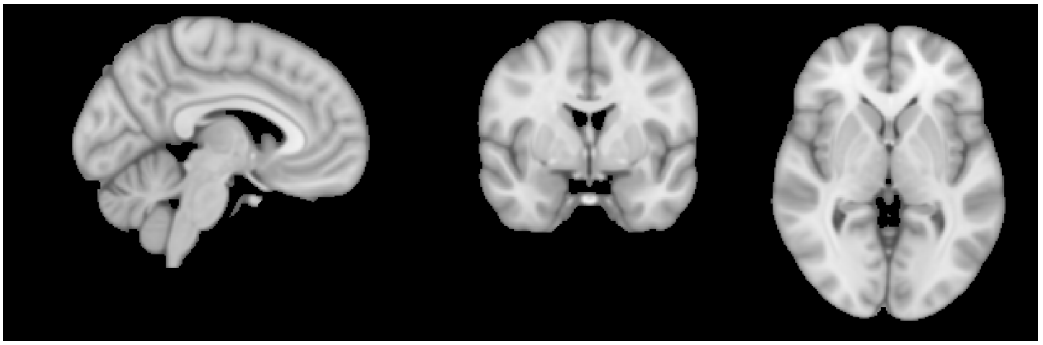

Figure 3.6: The MNI152 T1-weighted template.

FMRIB's Linear Registration Tool (FLIRT) is a fully automated, robust, and accurate tool for affine brain image registration. Using a linear registration tool with 12 degrees of freedom is satisfactory as the brain-extracted MRI images all have a similar structure. Twelve degrees of freedom are possible when the image has three dimensions.

Figure 3.7 clearly illustrates how FLIRT uses rotation around the x-axis (left column) and transformation in the z-dimension (right column) to align the MRI scan of the brain to the MNI152 template. Registering each scan to the template ensures that the orientation and shape of the brains are equal.

Having an approximately equal structural similarity among all MRI scans aids the GAN as it does not need to learn how to apply varying affine transformations to generate 1.5T* MRI images. Instead, the network can specialize in detecting signal-to-noise changes between the scans, providing better results.

3.2.5 Outlier Removal

The brain extraction tool does not always create satisfactory results 3.4. Including unsatisfactory brain extractions in the dataset could prevent the Pix2Pix and the CNN classifier from learning important patterns, resulting in poor image generation and

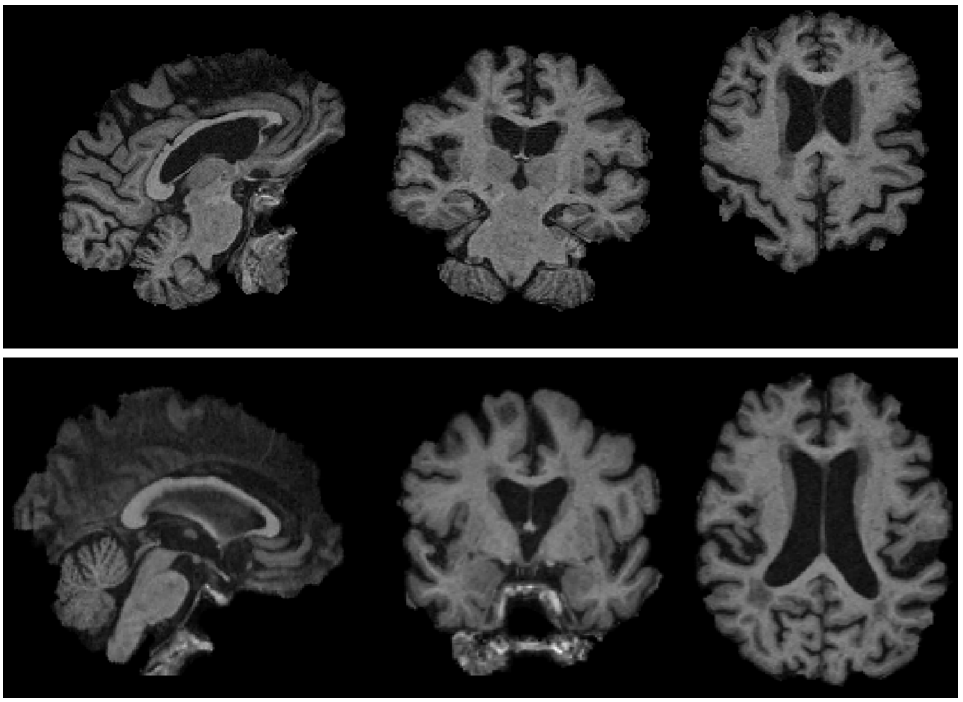

Figure 3.7: An MRI scan (top) registered to the MNI152 template to create a more normalized MRI scan (bottom).

multi-class classification. Removing the undesired scans is done by calculating the mean squared error and the structural similarity pair-wise of each scan.

Structural similarity (SSIM) is a perception-based model independent of visibility conditions and threshold values. The main aim of SSIM is to extract structural information from an image, which is done by analyzing luminance, contrast, and structure. Using a threshold of 0.7 when calculating the pair-wise SSIM for all MRI scans removes some of the unsatisfactory images without decreasing the size of the dataset by a substantial amount.

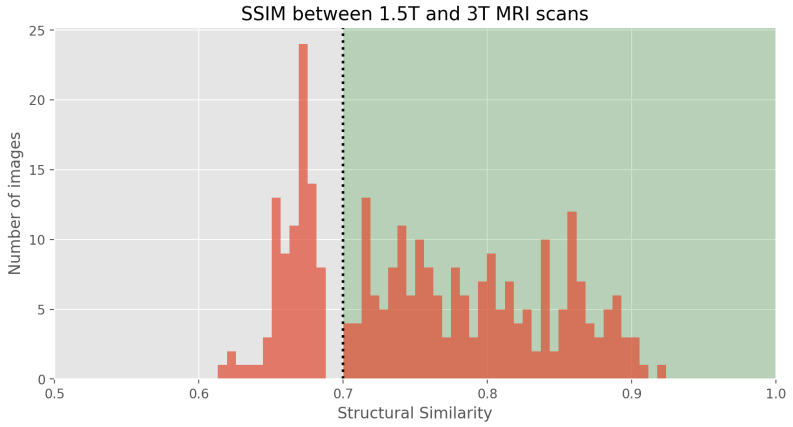

Figure 3.8: SSIM distribution with a threshold of 0.7.

When analyzing the structure of the MRI scans, SSIM might not always provide a score that reflects the pair-wise difference between MRI scans. This is because the general structure of a satisfactory extracted brain and an unsatisfactory extracted brain will be relatively similar. This is also the case for luminance and contrast. The inclusion of analyzing the mean squared error (MSE) partially solves this issue. MSE punishes the difference between voxel intensities more, resulting in another way of detecting dissimilarities between MRI scans. Using a threshold value of 60000 again removes some of the unsatisfactory images without decreasing the size of the dataset by a substantial amount.

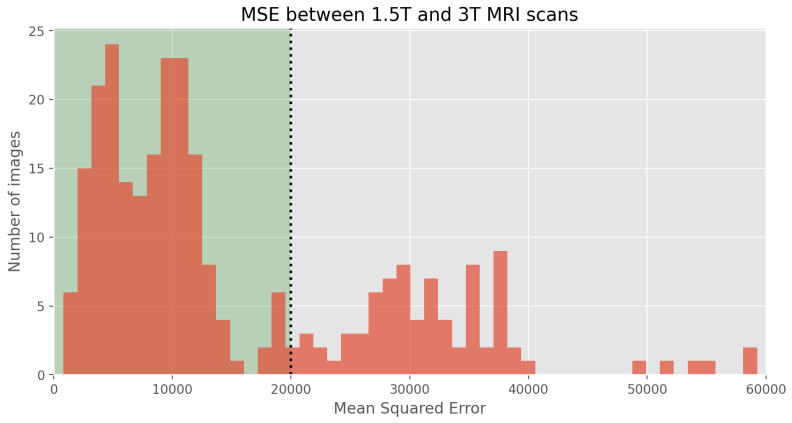

Figure 3.9: MSE distribution with a threshold of 20000.

A final manual evaluation of the dataset was required to eliminate the final unsatisfactory MRI scans, resulting in a final dataset of 290 1.5T and 290 3.0T MRI scans.

3.2.6 Histogram Matching

The final preprocessing step matches the pixel intensity distribution among the 1.5T and the 3.0T domains. Histogram matching is important to consider as patients with excessive fat in the head or neck region will have a skewed pixel intensity distribution. In T1 MRI scans, fat is marked as bright spots in the image. This affects the FSL tools and must be corrected to make the pair-wise pixel intensity more uniform. This prevents the GAN from having to guess whether the 3.0T MRI image will have the same intensity or not, which in return allows for more accurate results.

Therefore, a 1.5T MRI template and a 3.0T MRI template were created by generating the average intensity image from each domain. Figure 3.10 shows how visually similar the two templates are.

The cumulative density function (CDF) for the voxel intensities of the templates shows a slight difference in the intensity distribution. Both templates have a similar distribution

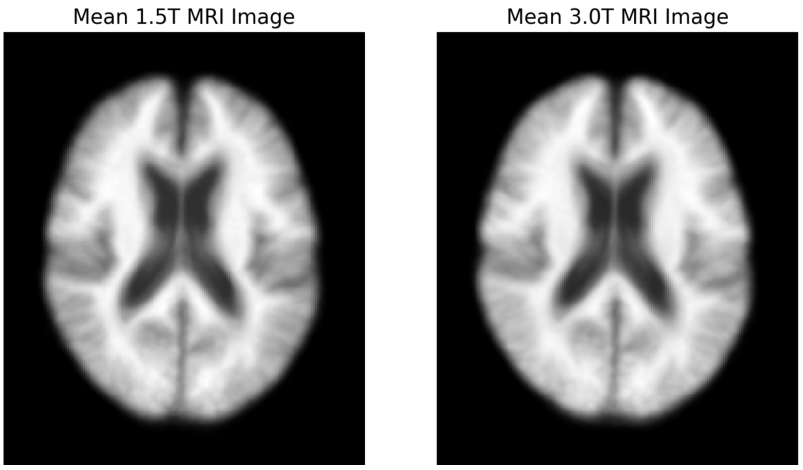

Figure 3.10: Averaged MRI images for the 1.5T and the 3.0T domain.

of the voxels with lower intensity values. On the other hand, the 3.0T domain have a higher concentration of high voxel intensities, resulting in a steeper CDF incline from intensity 0.6 and upwards.

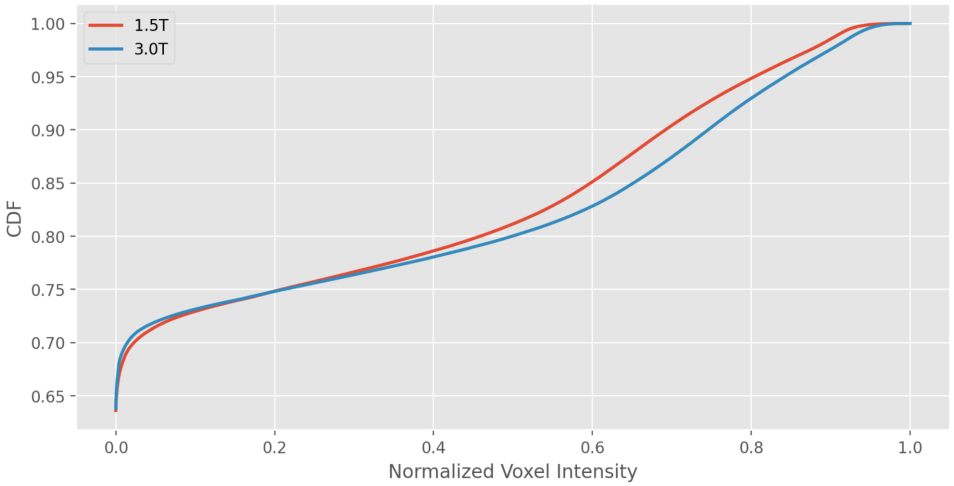

Figure 3.11: CDF of the averaged MRI images for the 1.5T and the 3.0T domain.

The histogram matching technique will ensure that Pix2Pix does not need to learn a voxel intensity distribution pattern for all pair-wise MRI scans. However, applying histogram matching involves a risk of removing important information from the MRI scans. The information loss is minimized by using the domain-specific templates created and is also the reason why the effect of the process is challenging to notice visually.

3.2.7 Augmentation

The dataset consists of 290 1.5T MRI scans and 290 3.0T MRI scans. Generative adversarial networks benefit significantly from large datasets as they introduce more

variance, which helps avoid overfitting. This is especially true regarding the dataset used in this thesis, as all scans have a similar shape. Therefore, dataset augmentation is necessary to increase the dataset variation and avoid the GAN finding an undesired local minimum, thus generating an unsatisfactory final result.

The augmentation used consists of flipping and translating the MRI scans in space. The probability of a flip being applied and a translation taking place is 20 percent. All MRI scans consist of 3 dimensions, meaning that the scans can be flipped along three axes. In addition, the image flipping directions are independent, meaning that multiple flips can occur on the same image. Translating a scan includes moving the brain part of the MRI scan ten voxels in any direction. The same augmentations are used for both the 1.5T and the 3.0T image of a patient given a visit and are applied dynamically each time a new image set is used to train the GAN.

Augmentation virtually increases the size of the dataset and introduces more diversity to the dataset. Including augmentation techniques generalizes the training procedure [30], which in Pix2Pix terms allows the higher frequency details to be detected. Several augmentation methods are not included in this thesis, such as rotation. MRI scans consist of square voxels placed out in a three-dimensional grid. Rotating the contents of an image causes each voxel intensity to be distributed in a new grid. These processes introduce interpolation, such as bilinear and trilinear interpolation. Figure 3.12 illustrates a two-dimensional example of this effect. Smearing caused by rotation is an undesired effect as finer details in the 3.0T MRI scans become harder to detect and notice. This effect is not present in flip/flop and transformation operations.

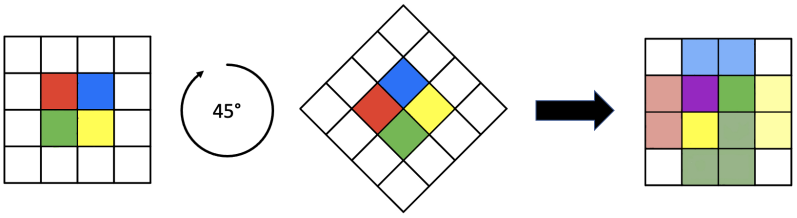

Figure 3.12: Interpolation of a rotated image, causing smearing.

3.3 Dataset Analysis

The class distribution within the data set is essential when using both GANs and CNN classifiers. A skewed class distribution makes it difficult to learn the characteristics of the minority classes. This leads to a more complex process of distinguishing minority classes from majority classes. Most machine learning algorithms for classification assume an equal distribution among classes [31].

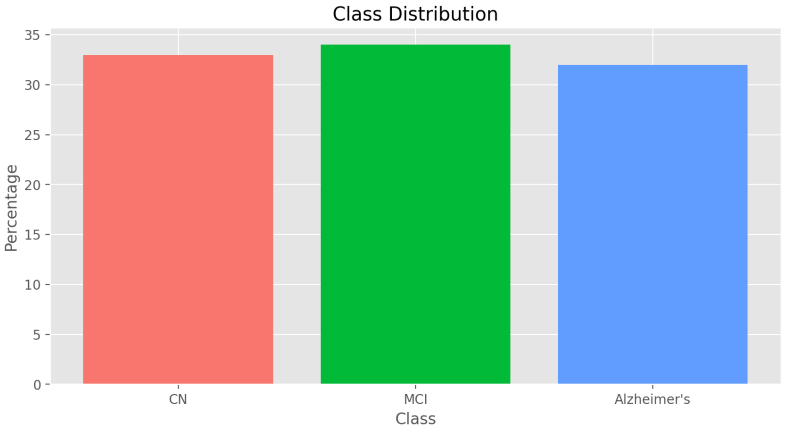

Figure 3.13: Dataset Class Distribution.

The preprocessed dataset has a very even class distribution, as shown in figure 3.13. Another important property to analyze is the mean voxel intensity for each class. This could have a significant impact on classification tasks. For example, having one class stand out regarding voxel intensity could make the classifier focus on such features instead of brain features unique to the class. The preprocessed dataset also avoids this problem, as shown in figure 3.14.

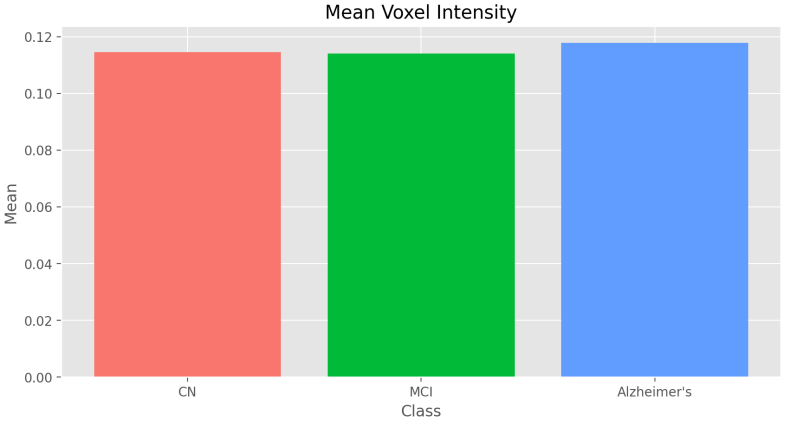

Figure 3.14: Voxel Intensity Distribution.

3.4 Models

3.4.1 CNN Classifier

The classifier architecture used to perform multi-class classification on each part of the dataset consist of three convolutional blocks. The first two blocks are composed of a convolutional layer, a batch normalization layer, a ReLU activation function, dropout, and a max-pooling layer. The last block only consists of a convolutional layer.

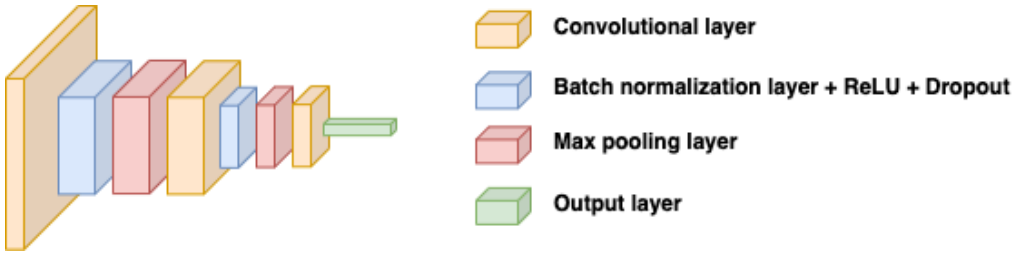

Figure 3.15: Architecture of the CNN multi-class classifier.

A more complex architecture containing two more convolutional blocks was also proposed. However, this architecture had to be simplified to the current solution due to memory issues in the GPU. Graphics processing units (GPU) are built to handle matrices and vectors efficiently. Utilizing the performance of GPUs includes storing all computed matrix operations and all the weights used in the network on the memory of the GPU. This becomes a problem when working with three-dimensional inputs as the calculations use a lot of memory. A solution is to reduce the input size to 128*128*128 (later referred to as 128 cubed) instead of the desired input shape of 256. Reducing the input shape corresponds to a voxel count 87.5 percent less than a 256 cubed input.

The input is first reshaped to 164 cubed to retain as much information as possible before cropping the input to the correct size of 128 cubed. Reshaping the input to a square format can introduce some information loss. Figure 3.16 illustrates how 218 voxels are reshaped into 164 voxels, essentially removing 54 voxels in that particular axis. The information loss is justified by the fact that the process is performed on MRI scans from both the 1.5T and the 3.0T domains. As such, the amount of information loss is approximately the same, meaning that MRI scans from the 3.0T domain still have a higher SNR than MRI scans from the 1.5T domain.

On the other hand, no critical information is lost by reshaping each MRI scan to 164 cubed and cropped to 128 cubed. This is because the cropping procedure removes dark areas around the brain, not the brain itself.

The first convolutional block uses a kernel size of 5, a stride of 1, and a padding of 2. This results in the output having the same size as the input size. Additionally, the number of channels are increased from 1 to 128. The network uses one input channel since the inputs are grayscale MRI scans. Alternatives include RGB images, which have three channels. Increasing the number of channels increases the abstractions the CNN can extract from the input data. Increasing the number too much causes GPU memory issues. The following batch normalization layer normalizes the values for each batch, standardizing the learning process. A ReLU activation function is used to prevent the exponential growth in the computation required to operate the CNN, thus speeding up the training process. The dropout layer limits the possibilities of the classifier overfitting to the

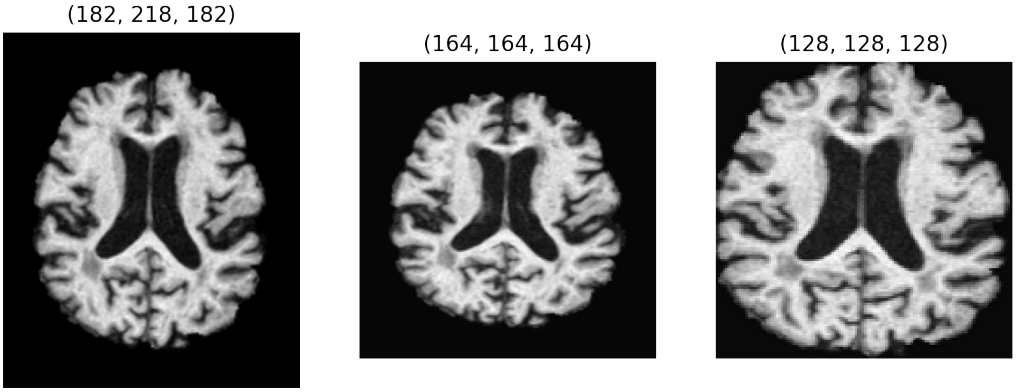

Figure 3.16: Reshaping and cropping an MRI scan.

training data. The dropout implementation has a 30 percent chance of deactivating each neuron in the CNN. This process generalizes the CNN, meaning that the classifier will have reduced performance on the training data but an improved performance of the test and validation datasets. The final part of the convolutional block includes a max-pooling layer, reducing the variance and the number of computations. The max-pooling layer uses a kernel size of 2 and a stride of 2, reducing the size of the MRI scan by a factor of 2.

The second convolutional block operates similarly to the first block. The second block reduces the number of features from 128 to 64 while also reducing the MRI scan size further, from 64 to 32 cubed.

The last convolutional block operates only consists of one convolutional layer. The convolutional operation uses a stride of 1 and the same kernel size as the input of 32 cubed. The number of features is reduced from 64 to the number of desired classes, which in this case is 3. The MRI scan is assumed to belong in the class with the highest value. The CNN classifier uses the Adam optimizer. Adam converges faster than other alternatives such as SGD, which is a required effect due to the dataset's limited size. Overfitting can occur when using a small dataset with an optimizer that converges slowly. SGD optimization task also introduces a risk of converging to local minima instead of a global minimum. This is why Adam is preferred in this type of architecture. The beta parameters of Adam reflect the initial decay rates used when estimating the first and second moments of the gradient descent. These values reflect the momentum of Adam and are set to 0.9 and 0.999, respectively. High beta values are the default configuration of Adam, with one of the reasons being that the values are exponentially multiplied by themselves. Having the beta values set to 0.5 or lower drastically changes the characteristics of Adam, which is undesired in this instance.

The classifier uses a cross-entropy criterion to calculate the loss. This is required as the CNN performs multi-class classification. Other alternatives for classification problems include binary cross-entropy loss, but this criterion only works in binary classification problems.

A batch size of two was used as a larger batch size would result in GPU memory issues. Having a batch size lower than two would eliminate the purpose of using a batch normalization layer. The combination of 30 epochs and a learning rate of 0.002 was chosen as the desired values. The combination resulted in the classifier avoiding overfitting while providing a decent classification score.

Comparing Classifier Results

Deep learning applications tend to produce varying results, even when the same piece of code is used. This is because of the random nature of deep learning. Some of the most important random processes in this classifier consist of the weight initialization of the CNN and how the dataset is separated into training, validation, and test sets. Cross-validation is a common technique to minimize the downsides of the random nature but is not implemented in this thesis due to time constraints. Instead, the seed of all random libraries and functions used are defined. This results in deterministic outputs, and the same piece of code will always produce the same output.

Making the randomness of deep learning deterministic is not a good solution when the goal is to understand how well a classifier performs. This is why the results of this thesis are never directly compared to other results. Instead, using a deterministic seed for the random functions provide a reliable way of internally comparing the classification results between 1.5T, 3.0T, and 3.0T* MRI images. This ensures that the same data is separated into training, validation, and test datasets across all the datasets, resulting in results that can be compared against each other.

3.4.2 Pix2Pix

A learning rate of 0.0002 was used when training the GAN. Other learning rates were tested in the early stages of the work but did not provide satisfactory results, resulting in an early termination of the training procedure.

U-Net

The U-net generator of Pix2Pix had to be configured to operate with the brain MRI scans. This included implementing three-dimensional convolutional layers, transposed convolutional layers, and instance normalization layers. The architecture also had to be adjusted in minor ways to allow grayscale, three-dimensional images to be used as input. As a result, the U-net can be operated with the desired input shape of 256 cubed without introducing memory issues with the GPU.

The U-net consists of an initial convolutional layer which increases the number of features in the input from 1 to 64 while also reducing the input size from 256 cubed to 128 cubed. This is done for the same reasons discussed regarding the CNN classifier 3.4.1. The next part of the U-net consists of 6 equal convolutional blocks consisting of a convolutional layer, an instance normalization layer, and a leaky ReLU activation function. Training Pix2Pix utilizes a batch size of 1, meaning that the instance normalization layer can be compared to a batch normalization layer. This technique has been demonstrated to be effective at image generation tasks [32]. Each convolutional block doubles the number of features (up to 512 features) and halves the image size, with the final convolutional block having 512 features and an image size of 2 cubed.

Another type of convolutional block known as the bridge connects the encoder part of the U-net with the decoder part. The bridge consists of a convolutional layer and a ReLU layer, ensuring that the data being passed through the network has 512 features and a size of 1 cubed.

The decoder part of the U-net mirrors the architecture of the encoder by having seven transposed convolutional blocks consisting of a transposed convolutional layer, an instance normalization layer, and a ReLU activation function. The first three transposed convolutional blocks also introduce dropout with a probability of 50 percent, as suggested in the Pix2Pix paper [21]. Each transposed convolutional block uses a concatenated input of the previous block combined with the output from the corresponding convolutional block in the encoder. See Figure 3.17 for clarification. These blocks reduce the number of features and increase the size of the image. A final block is responsible for matching the size of the output with the size of the input, being 1 feature and an image size of 256 cubed.

All convolutional layers, both regular and transposed, use a kernel size of 4, a stride of 2, and a padding of 1. This convolution architecture is carefully chosen as it avoids creating checkerboard artifacts on the image [33].

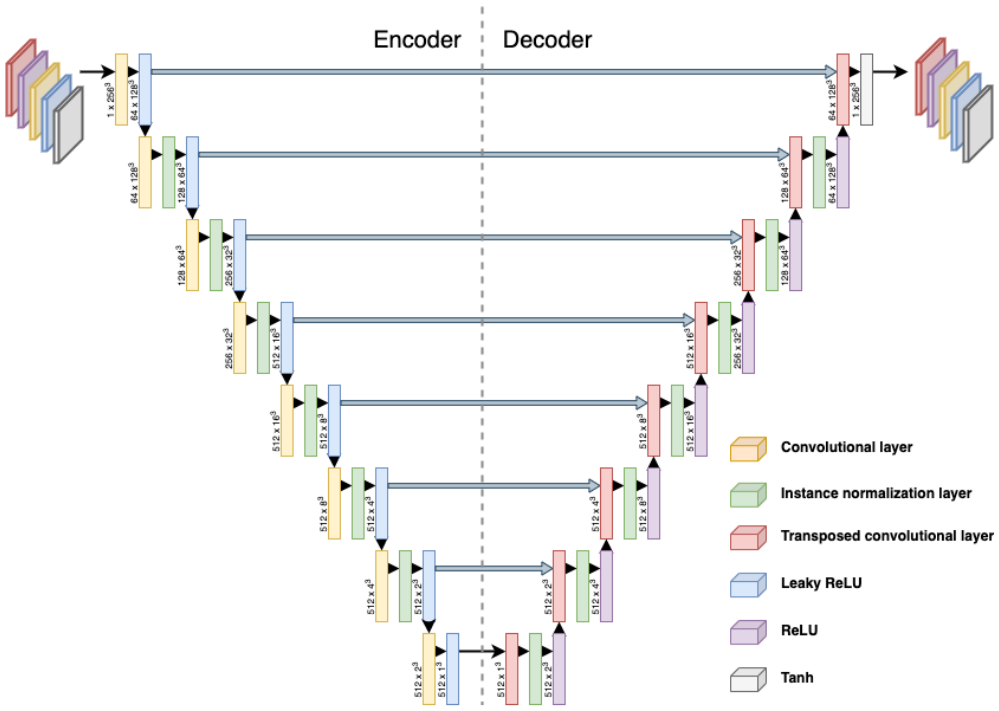

Figure 3.17: The U-net architecture used in this thesis.

A tanh activation function replaces the ReLU activation function in the last transposed convolutional block. This is done to match the voxel intensity distribution of the input MRI scans.

PatchGAN

The Pix2Pix implementation uses a PatchGAN as the discriminator. Other alternatives were also explored, such as using a binary version of the CNN classifier. The PatchGAN discriminator was chosen because it uses less complex parameters. A less computational approach is important considering that all data from the generator and the discriminator are kept in the GPU.

The architecture of the PatchGAN consists of 5 convolutional blocks, each containing a convolutional layer, an instance normalization layer, and a leaky ReLU activation function. The only exception is that the first convolutional block does not contain an instance normalization layer. Similar to the U-net encoder, the layers of the PatchGAN gradually increase the number of features while reducing the image size. All convolutional blocks use a kernel size of 4, a stride of 2, and a padding of 1, for the same reasons mentioned in section 3.4.2. The last two convolutional blocks use a stride of 1 instead of 2, which results in the output having the desired size and dimensions. The final layer removes the instance normalization layer and the leaky ReLU activation function. No activation functions are required in the final layer. This is because the output of the

PatchGAN is sent into a binary cross-entropy with logits loss function (BCE*), combining a sigmoid activation function with the binary cross-entropy loss function.

The input of the PatchGAN is a bit different compared to other classifiers. The PatchGAN alternates between receiving two types of inputs. Either the 1.5T and 3.0T MRI scans or the 1.5T and the 3.0T* MRI scan generated by the generator. As described in 2.7.3, the PatchGAN returns a grid instead of a single value. When given the task of classifying a 256 cubed image, the PatchGAN returns an output with the size of 30 cubed. Here, each voxel corresponds to a 70 cubed area of the input. The implementation defines real 3.0T MRI images to belong in class 1 and generated 3.0T* images in class 0.

3.4.3 Calculating loss

When classifying a real 3.0T MRI scan, the PatchGAN output is sent to the BCE* loss function with a grid of ones in the same shape. This is done as the desired output from the PatchGAN would be a cube of only ones. The same goes for generated 3.0T* MRI scans, the only difference being that a cube of zeros is used. The mean loss value from the two processes is then used with Adam to perform backward propagation.

Training the U-net follows a similar approach. In this case, the loss created from the generated 3.0T* MRI scans through the use of BCE* is used in combination with another loss function known as L1-loss. L1-loss uses the mean absolute error between the 3.0T and the 3.0T* MRI images. L1-loss produces very blurry images [21] but excels at capturing the low frequencies of an image. Therefore, combining the L1-loss and the BCE* loss allows the generator to learn both the low and the high frequencies of the MRI scans, thus making the image generation more accurate [21]. The L1-loss tends to be a lot lower than the BCE* loss. For this reason, the L1-loss is multiplied by 100 to make both losses influence the total loss similarly. The same multiplier value was used in the Pix2Pix paper [21].

3.5 Existing Baselines

The result of earlier work revolves around using a fully connected neural network to perform binary classification between cognitive normality and Alzheimer's disease [24]. The results showed that generated 3.0T* MRI scans performed better than the corresponding 1.5T MRI scans. This was measured by an increase in the AUC score from 0.904 to 0.926, corresponding to an increase of 2 percent. The usage of ROC is limited to evaluating binary classification tasks, making other evaluation metrics such as accuracy,

precision, and recall more attractive. One solution to use ROC is to combine classes, resulting in a "one vs. the rest" evaluation. The mean AUC score obtained through multi-class classification cannot be directly compared to a binary classification task but can still be of interest.

Chapter 4

Experimental Evaluation and Results

4.1 Evaluation of the early stages

The earliest stages of using Pix2Pix to generate 3.0T* MRI scans of the brain included generating a dataset containing two-dimensional brain slices. The original Pix2Pix paper researched multiple 2D datasets but no 3D datasets. As such, a 2D dataset was used to verify the implementation of the original architecture [21]. The model showed some promising results without significant modifications, but they were far from satisfactory.

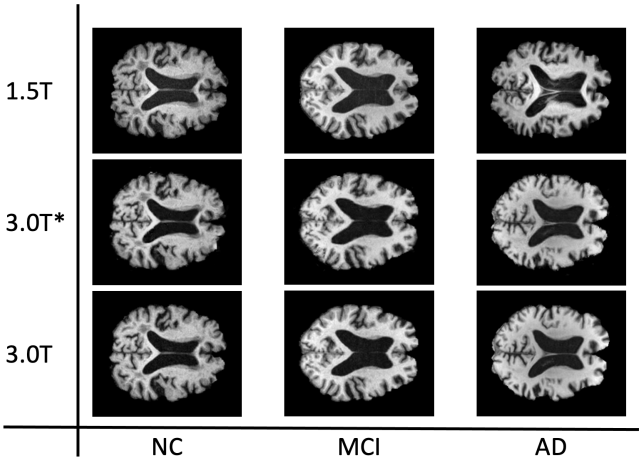

Figure 4.1: Image comparison of the generated from the 2D Pix2Pix.

It is important to note that the earlier versions of the Pix2Pix implementation used a dataset lacking several preprocessing steps. This is because the preprocessing pipeline was continuously developed to include several necessary steps to ensure a high-quality dataset. Therefore, a 2D dataset was only meant as an implementation test and was not further developed to use the improved dataset.

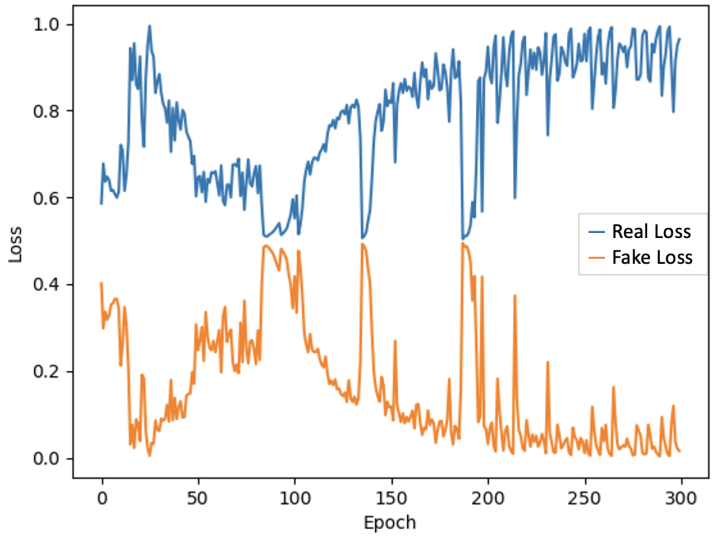

Figure 4.2: Loss ratio of both real and generated MRI scans.

The first Pix2Pix implementation clearly shows the generator and the discriminator working against each other. It appears that the discriminator wins the min-max battle at around epoch 100. From here, the difference between the two losses gradually increases. The blue line shows the ratio of all real 3.0T MRI scans classified as real, while the orange line shows the ratio of all generated 3.0T* MRI scans classified as real. A significant separation between the two losses resulted in the generator being unable to learn new features. Having the two losses grouped around 0.5 is also unsatisfactory. This means that the discriminator correctly classifies the 3.0T and the 3.0T* images half the time. This is the same as the discriminator guessing the classification. Random guesses provide no useful information generator, making it impossible to learn new features. Having the model correctly classify each class around 70 percent of the time is the golden line according to the Pix2Pix paper [21].

4.2 Evaluation of 1.5T MRI Scans

Evaluating how well each dataset performs is done utilizing multiple metrics. The confusion matrix shows that the classifier tends to classify both CN and AD subjects as MCI. This makes sense as transitioning from a cognitively normal brain to being diagnosed with Alzheimer's disease includes a transition through mild cognitive impairment. The matrix also shows that the classifier distinguishes between the CN and AD classes.

The quality measurements form a baseline for comparison against the 3.0T and the generated 3.0T* MRI images. The accuracy was 0.79, meaning that 79 percent of the total number of samples was correctly classified. The f1-score of 0.79 is also important,

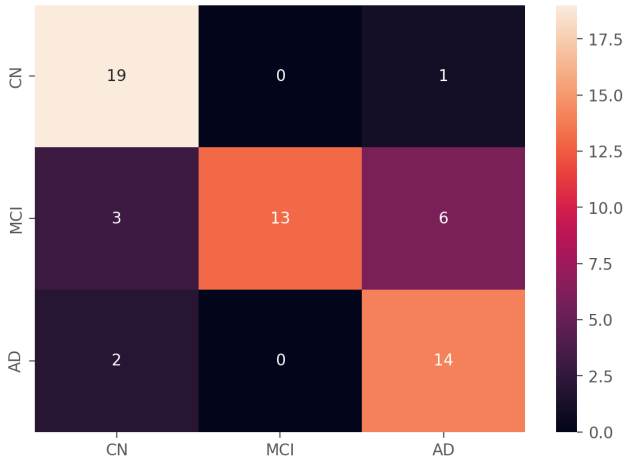

Figure 4.3: Confusion matrix from classifying the 1.5T dataset.

as it represents a combined evaluation of precision and recall. The most notable outlier from the evaluation graph is the recall metric for the Alzheimer's disease class. A low score of 0.58 indicates that several subjects diagnosed with AD have been classified as either CN or MCI. This could indicate a poorly trained classifier or that the characteristic of Alzheimer's disease is not captured correctly in 1.5T MRI scans.

	Precision	*Recall*	*F1-Score*
CN	0.79	0.95	0.86
MCI	1.00	0.59	0.74
AD	0.67	0.88	0.76
Accuracy			0.79
Average	0.82	0.81	0.79

Table 4.1: Evaluation from classifying the 1.5T dataset.

4.3 Evaluation of 3.0T MRI Scans

The 3.0T MRI scans were evaluated in the same manner as evaluating the 1.5T MRI dataset. Having MRI images with a higher SNR should, in theory, provide the classifier with more information which can lead to improved classification. The confusion matrix generated from the 3.0T classification visualizes a more accurate class distribution and shows a more apparent main diagonal. The matrix also illustrates difficulties in learning the AD class properly. Multiple CN and MCI scans have been classified into the AD class. Despite this, the overall number of MRI scans classified incorrectly decreases when classifying the 3.0T MRI scan dataset.

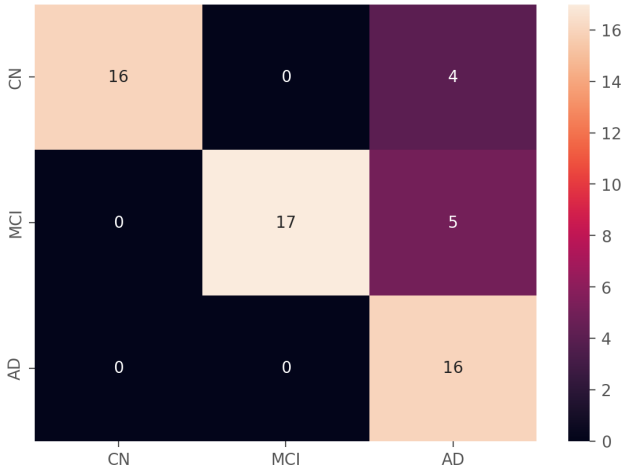

Figure 4.4: Confusion matrix from classifying the 3.0T dataset.

The accuracy and the f1-score increase by approximately 7 percent, indicating that an improved SNR directly affects the performance of the CNN classifier. However, classifying the 3.0T MRI dataset does not yield improved results across all metrics compared to the scores acquired using the 1.5T MRI dataset. The precision for the AD class and the recall for the CN class is lower, which accurately matches the information from the confusion matrix.

	Precision	*Recall*	*F1-Score*
CN	1.00	0.80	0.89
MCI	1.00	0.77	0.87
AD	0.64	1.00	0.78
Accuracy			0.84
Average	0.88	0.86	0.85

Table 4.2: Evaluation from classifying the 3.0T dataset.

4.4 Generating 3D 3.0T* MRI images with Pix2Pix

4.4.1 Evaluation of Histogram Matching

This section contains the two most promising approaches to generating 3.0T* MRI images, with the main difference being with or without histogram matching between the 1.5T and the 3.0T MRI scans.

Without Histogram Matching

One of the solutions for converting MRI scans from the 1.5T domain to the 3.0T domain by generating 3.0T* MRI scans included a preprocessing step without histogram matching of each pair of MRI scans. This approach was tested because of the assumption that removing histogram matching would yield a more desirable result. Histogram matching alters the distribution of voxel intensities of an image to match that of another image, see section 2.3.4. This process can, in some cases, make some grayscale values in the original image indistinguishable from each other when processed to match another intensity distribution. Analysis of the histogram matching preprocessing step illustrated in Figure 3.11 illustrates that, although different, the cumulative distribution function (CDF) of both the 1.5T and the 3.0T MRI scans have the same characteristics.

The first assumption of the evaluation of histogram matching resulted in a preprocessing pipeline without histogram matching. This served as a baseline to compare the use of histogram matching.

The blue line in Figure 4.5 indicates the ratio of real 3.0T MRI scans classified as real, and the orange line indicates the ratio of generated 3.0T* MRI scans classified as real. The area between the two ratios gradually increases for 5000 iterations before flattening out and decreasing slightly in the end. The GAN trained for approximately 11700 steps, corresponding to 50 epochs.

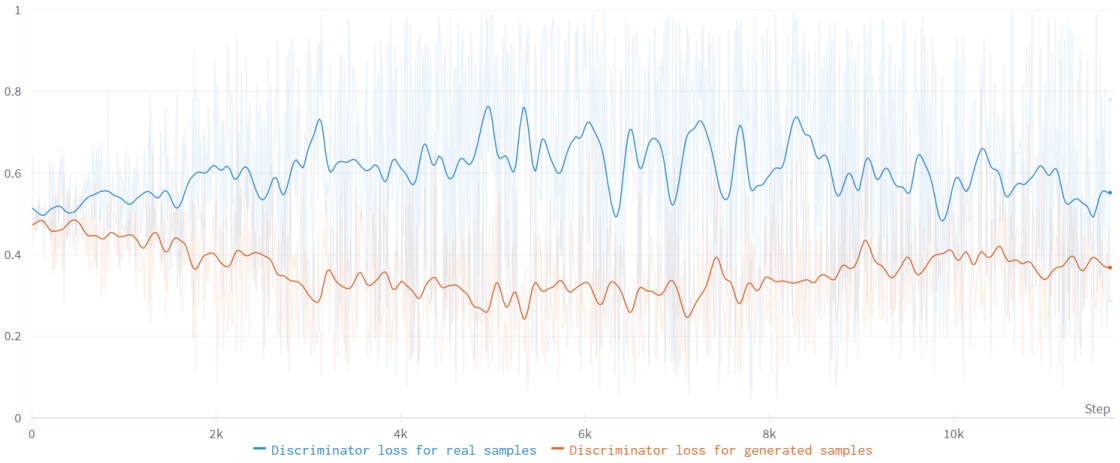

Figure 4.5: Loss-graph of Pix2Pix without histogram matching.

The desired training pattern in Pix2Pix, as described in the paper [21] is having the two loss-ratios stabilized around 0.7 and 0.3, respectively. It is also desirable to avoid a gradually increasing difference between the ratios, as the generator is learning less and less. This pattern results in the generator and the discriminator improving simultaneously, thus avoiding a training collapse. The desired loss ratios in this thesis are expected to be

more centered around 0.5. This is because of the few unique features connected to 1.5T and 3.0T MRI scans. The Pix2Pix paper [21] ran experiments on datasets containing vastly different domains, such as street maps and satellite images. Having two domains that only deviate from each other through SNR results in the discriminator having to guess more often, resulting in loss ratios closer to 0.5. The 1.5T and the 3.0T MRI scans contain a lot of low-intensity voxels around the brain. These dark voxels are impossible to classify into the correct domain correctly.

Figure 4.6 illustrates that Pix2Pix fails to generate realistic 3.0T* MRI images when the voxel intensity distribution between the 1.5T and 3.0T MRI scans differ, despite both images being normalized. This is likely because the GAN focuses on learning the intensity difference instead of the higher frequencies commonly found to distinguish 1.5T and 3.0T MRI scans.

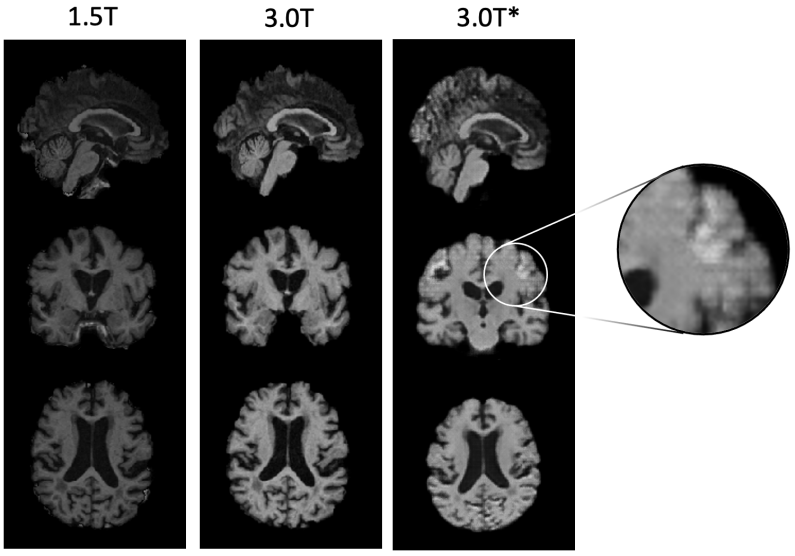

Figure 4.6: A generated 3.0T* MRI image compared to the corresponding 1.5T and 3.0T MRI images.

The varying voxel intensity distribution is not constant within the dataset. For example, some sets of 1.5T and 3.0T MRI scans have an approximately equal distribution, while others do not. This results in a feature Pix2Pix cannot learn and a learning process that cannot improve past a certain point without overfitting.

The generated 3.0T* dataset illustrates promising classification results despite visually unsatisfactory images. The y-axis of the confusion matrix indicates the true class for each MRI scan, while the x-axis indicates the predicted class from the CNN classifier. A confusion matrix with positive values along the main diagonal indicates a classifier perfectly classifying the dataset.

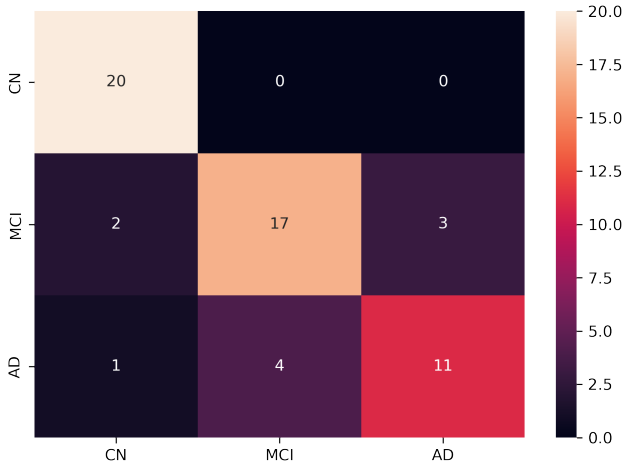

Figure 4.7: Confusion matrix from classifying the generated 3.0T* dataset without histogram matching.

As with the previous classifier results, the classifier struggles with differentiating between MCI and AD, as seen in the low precision score of the AD class. The classifier also struggles with correctly classifying images to the MCI class when using the generated 3.0T* MRI images without histogram matching. This can be seen through the low precision score for the MCI class. These low values can be interpreted as the classifier not finding apparent features that separate MCI from AD in all instances.

	Precision	*Recall*	*F1-Score*
CN	0.87	1.00	0.93
MCI	0.81	0.77	0.79
AD	0.79	0.69	0.73
Accuracy			0.83
Average	0.82	0.82	0.82

Table 4.3: Evaluation from classifying the generated 3.0T* dataset without histogram matching.

With Histogram Matching

The other approach was to include histogram matching in the preprocessing pipeline. As described in 3.2.6, the histogram matching approach consists of creating a mean MRI image for both the 1.5T and the 3.0T domain, then matching all corresponding MRI scans to the template. This approach retains the difference in voxel intensities between the two domains while ensuring a deterministic voxel distribution.

Figure 4.8 shows a more expected learning evolution compared to Figure 4.5. The loss ratios fluctuate at the start before stabilizing at around 0.6 and 0.4. The progression

of the loss ratios might indicate convergence close to 0.5 with prolonged training of the Pix2Pix GAN.

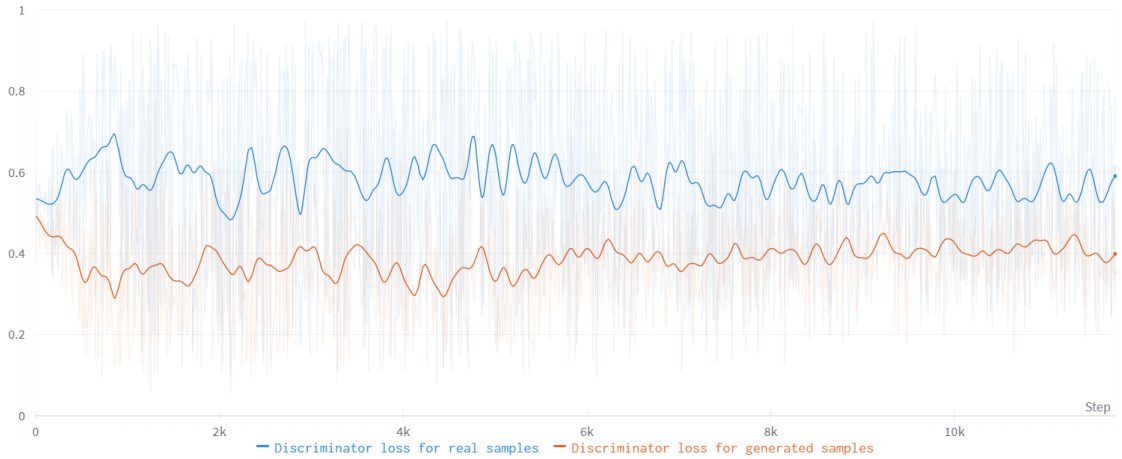

Figure 4.8: Loss-graph of Pix2Pix with histogram matching.

Visual analysis shows how the Pix2Pix GAN manages to generate realistic-looking MRI images. The outer parts of the brain appear to have been smoothed out, making it easy to differentiate generated 3.0T* MRI images from real 3.0T scans. Figure 4.9 also shows the improved quality and SNR originating from histogram matching. Having the voxel intensities equal for each magnetic field strength domain enables Pix2Pix to learn the higher frequency patterns better, resulting in fewer artifacts.

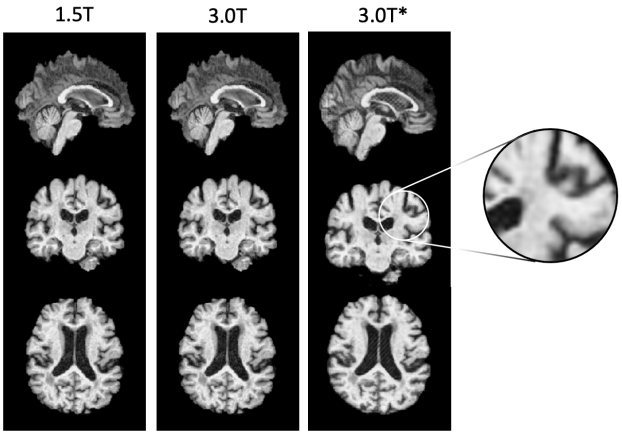

Figure 4.9: A generated 3.0T* MRI image compared to the corresponding 1.5T and 3.0T MRI images.

Analysis of the confusion matrix reveals a similar classification distribution as generating 3.0T* without histogram matching. MRI scans belonging to the CN and the AD class are classified to the MCI class, while the problem of differentiating between MCI and AD persists. Classifying too many MRI images as MCI can indicate that histogram matching removes some features required to distinguish MCI MRI scans.

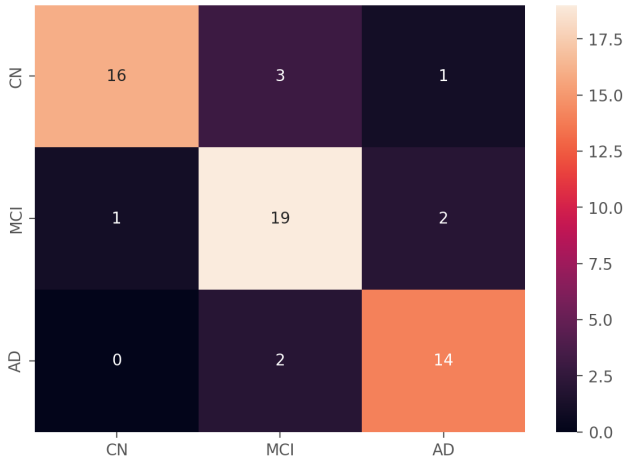

Figure 4.10: Confusion matrix from classifying the generated 3.0T* dataset with histogram matching.

The most apparent outlier for the evaluation metrics is the precision score for the MCI class, a trait that can also be seen in the confusion matrix. The low precision on MCI classification is also reflected in the recall score for the CN and AD classes, but in a smaller manner compared to the previous results.

	Precision	*Recall*	*F1-Score*
CN	0.94	0.80	0.86
MCI	0.79	0.86	0.83
AD	0.82	0.88	0.85
Accuracy			0.84
Average	0.85	0.85	0.85

Table 4.4: Evaluation from classifying the generated 3.0T* dataset with histogram matching.

Overall, histogram matching results in visually better generated 3.0T* images while having superior metric scores compared to not using histogram matching. The results of having histogram matching included in the preprocessing of the clinical MRI scans are therefore chosen as the primary approach.

4.5 Performance Comparison

Performance metrics scores show a distinct improvement of the generated 3.0T* MRI images compared to the original 1.5T MRI scans. The scores approach that of the 3.0T MRI scans. The confusion matrices show that the 1.5T dataset lacks features necessary for classifying MCI correctly while classifying the 3.0T dataset shows sub-optimal scores

for the AD class. The comparison also shows how the confusion matrix of the generated 3.0T* dataset contains traits of both the 1.5T and the 3.0T dataset, which is expected.

The AUC for all datasets and all classes was also created. See Appendix B for figures showing the ROC curves and the corresponding AUCs. The average AUC score for each dataset can be seen in Table 4.5 and clearly shows an improvement from the 1.5T domain to the 3.0T domain, with the generated 3.0T* MRI images in between. The AUCs have been calculated through a "one versus the rest" approach. Having the AUC score of 1 indicates that the class extracted from the rest is perfectly separated.

	3.0T	*3.0T**	*1.5T*
Accuracy	0.84	0.84	0.79
Precision	0.88	0.85	0.82
Recall	0.86	0.85	0.81
F1-Score	0.85	0.85	0.79
AUC	0.953	0.949	0.710

Table 4.5: Comparison of the average classification metrics for each dataset.

The classification results from the generated 3.0T* MRI images show a significant improvement over the results from the 1.5T domain dataset. The average of all metrics used is higher for the generated 3.0T* MRI images, closely matching that of the 3.0T domain. The average of all classification metrics shows an improvement of 0.045, corresponding to a 5 percent increase.

Chapter 5

Discussion

5.1 Comparison to Related Work

The biggest inspiration for the work of this thesis originates from a paper generating 3.0T* MRI images and evaluating them through a binary classification CNN [24]. The main evaluation metric used is the mean AUC, which increased from 0.907 to 0.932 for an ADNI dataset. Comparatively, the mean AUC illustrated in Table 4.5 was increased from 0.710 to 0.949. It is important to mention that these values cannot be directly compared as this thesis revolves around multi-class classification, not binary classification. This might explain why the mean AUC for the 1.5T dataset is comparatively low. Differentiating between MCI and AD using MRI scans with a low SNR could result in the classifier creating class probabilities close to 0.5, which is not reflected in a confusion matrix. The related work [24] avoids this issue through the binary classification approach.

5.2 Result Evaluation

The results indicate that the generated 3.0T* dataset performs better in classification tasks than the 1.5T dataset. The 3.0T dataset is marginally better, which is not surprising. However, despite the promising classification results, human analysis can separate the generated 3.0T* MRI scans from the 3.0T MRI dataset. Improving the visual interpretation could be achieved by a prolonged training period of the Pix2Pix. This is essential for being completely satisfied with the result, as dementia detection and diagnosing are done through a clinical presentation.

The results could also be improved slightly by carefully selecting MRI scans. Currently, the classifier has some difficulties separating Alzheimer's disease from the mild cognitive

impairment class. One of the explanations for this occurrence is the composition of the dataset. The dataset contains several MRI scans of the same patient but at different time points. Some of these visits indicate a diagnosis of MCI but evolve into AD at later stages. This gradual change is a problematic classification task, especially if doctors giving the diagnosis have a slightly different understanding of when MCI evolves to AD. Thus, an accuracy and f1-score of 100 percent are not necessarily achievable.

5.3 Pix2Pix Loss Evaluation

As explored in [24], one option is to include a classifier in the GAN training procedure. This is achieved by sending the generated 3.0T* images to the discriminator and a fully connected network. The discriminator creates a loss score corresponding to the evaluation between real and generated samples. On the other hand, the fully connected network creates a loss score corresponding to the evaluation between each class contained in the dataset. The generator can then use the combined loss score to update its weights. This approach could result in the generated 3.0T* MRI images having better classification metrics but could also result in visually unsatisfactory images. Having the discriminator loss as the most influential part of the combined loss would diminish the effect of including the fully connected discriminator. The opposite decision would generate 3.0T* MRI images designed for deep learning classification and could suffer in terms of visual appearance.

Instead, this thesis focuses on using a robust preprocessing pipeline to eliminate unwanted features from the clinical MRI scans. The preprocessing steps take on the task of the fully connected classifier by making the features of the brain more distinct.

5.4 Classification

Diagnosing patients with Alzheimer's disease is based upon a clinical presentation based on fluid and imaging biomarkers and fulfilling several criteria. The results from the classification metrics show that the MCI and AD classes are the most challenging classes to classify correctly. This is true for all the datasets. Classifying the 1.5T domain results in too many MRI scans being classified to the MCI class. Using the generated 3.0T* dataset for classification shows too many MRI scans being classified to the AD class. In the same way, classifying the 3.0T domain results in MCI and AD patients being mixed up.

One possible reason for the misclassification issue could be the preprocessing pipeline. Some vital information explaining the difference between MCI and AD might be lost through preprocessing tasks such as brain extraction and image registration. However, it is important to note that most MRI scans are classified correctly, as shown in Figure 4.5.

Another reason for the misclassification issue revolves around how each diagnosis is given. Human errors could interfere with the classification, thus resulting in misclassification. The progression between mild cognitive impairment and Alzheimer's disease occurs gradually. As such, neurologists could have different ideas about when MCI becomes AD. A strong MCI could therefore be more severe than a mild AD, which is a situation difficult to handle for a convolutional neural network.

5.5 Registration and Intensity Matching

The MNI152 template provided by FSL was used when registering all MRI scans to a common shape and form. The best approach would be to avoid having to register the MRI scans. Registering images involves altering the voxels, thus running the risk of removing features. A registration-free preprocessing pipeline requires the original clinical 1.5T and 3.0T MRI scans to have the same orientation, a criterion not feasible with the current dataset. An advantage of using the MNI152 template compared to other registration options (such as pairwise registration) is that the configuration of FSL FLIRT is tuned to MNI152. Using other alternatives would include fine-tuning each parameter to achieve satisfactory results.

The histogram matching procedure does, however, not use the MNI152 template. Instead, a template for each domain was created by averaging all MRI scans inside the domains. This created two templates with the voxel intensity distribution of each domain. Keeping the voxel intensity distribution for each domain separated could aid Pix2Pix in detecting essential features. An approach like this would not work for image registration as the two templates created were not uniform. Histogram matching includes a risk of combining ranges of voxel intensities into only one value, a process that removes potentially useful information. Other approaches could therefore improve the results of Pix2Pix further.

5.6 Dataset Size Limitations

Each dataset includes 290 MRI scans. All scans must be included in the training of Pix2Pix due to the limited number of samples. This is because of the complexity and diversity Pix2Pix has to learn to generate realistic and accurate 3.0T* scans. Including

all MRI scans introduces some bias towards the exact data used, which might exaggerate the classifier results. When classifying, the data is split into a training, a validation, and a test dataset. Splitting the dataset this way ensures that the classifier is not biased towards the data, as the final performance evaluation is done with the test dataset. The classifier only utilizes the training and validation dataset when training, making the test dataset function as new MRI images never seen before.

A better approach would be to generate 3.0T* MRI images from a new dataset not present in the Pix2Pix training. This approach would create more realistic results but is not viable due to the small dataset available.

Chapter 6

Conclusion and Future Directions

6.1 Conclusion

This thesis explores the potential of using GANs to increase the SNR of brain MRI scans, which can lead to improved dementia detection and diagnosis. The proposed methods rely on a complex preprocessing pipeline to extract the brain from clinical MRI scans while retaining as much information as possible. Increasing the SNR is done by having a Pix2Pix GAN train on preprocessed MRI scans of the 1.5T and the 3.0T domain, then generating 3.0T* MRI images based on the 1.5T MRI scan input.

A convolutional neural network classifier was used to classify each dataset into CN, MCI, and AD instead of the standard procedure of qualified professionals looking for biomarkers. A combination of best practices and manual hyperparameter configuration was used to tune Pix2Pix and the classifier to provide satisfactory results without overfitting.

Limited time prevented the Pix2Pix implementation from converging. Despite this, the implementation clearly illustrates the advantages and capabilities of working with generated 3.0T* MRI scans. The results of multiple evaluation metrics of the classification performance indicate that the generated 3.0T* dataset is superior to the 1.5T dataset in every way. The mean evaluation score is increased from 0.8025 to 0.8475, while the mean AUC is increased from 0.710 to 0.949.

The results from the classification process look promising, but the visual quality of the generated 3.0T* MRI images is inadequate. Despite this, this thesis clearly illustrates how GANs can increase the SNR of MRI scans and provides a solid foundation for further work using clinical 0.5T MRI scans from portable scanners.

6.2 Future Directions

6.2.1 Alternatives for Preprocessing

The desired preprocessing tool in this thesis was FSL because of its popularity and longevity [34]. Another popular preprocessing tool is FreeSurfer. FreeSurfer is a tool for neuroimaging data and provides several algorithms to process and analyze the human brain [35]. Like FSL, FreeSurfer provides tools for skull-stripping, bias field correction, and image registration for T1-weighted MRI images. Comparing the performance and result from the two preprocessing tools would be interesting, especially as FreeSurfer is open source while FSL is not. An improved preprocessing pipeline could mean better MRI images, resulting in an improved quality of the generated MRI images.

6.2.2 Pix2Pix

The Pix2Pix GAN trained on a 40 GB NVIDIA Tesla A100 GPU for 24 hours, resulting in fifty epochs and a total of 11600 assessments. Figure 4.9 illustrates the generated 3.0T* MRI images from the Pix2Pix training. The loss graph illustrated in Figure 4.8 indicates little to no converging at this state, showing that better results could be achieved by training the Pix2Pix for an extended period.

Using the trained Pix2Pix on new images such as clinical MRI scans from an institute not included in the ADNI database would also be interesting. This is because new images would reveal how robust the implementation is.

The Pix2Pix paper was published in 2017 [21]. An alteration was proposed in 2018 named Pix2PixHD [36]. Pix2PixHD follows the same principles as Pix2Pix but has high-resolution images in focus (e.g., 2048x1024). ADNI provides clinical MRI scans with a size of approximately 256 cubed, far less than what Pix2PixHD uses in their examples. On the other hand, focusing on high-resolution images often includes a focus on retaining more high-frequency details. This trait could be included in the conversion from the 1.5T domain to the 3.0T domain for increased SNR.

The current solution to evaluate the performance of Pix2Pix is through human analysis and classification evaluation. Another option is to calculate the Fréchet Inception Distance (FID). FID is a metric that calculates the distance between the feature vectors from generated and real images. A comparison between the FID from the 1.5T and the 3.0T domain and the 3.0T* and the 3.0T domain could provide an improved overview of how accurate the Pix2Pix-generated MRI images are.

6.2.3 Brain Analysis

Having a radiologist or a neurologist analyze both the preprocessed MRI scans and the generated 3.0T* MRI images would be insightful. Having a proper understanding of which parts of the brain MCI and AD can be detected would provide valuable information regarding the preprocessing and the reshaping methods. For example, a more aggressive brain extraction configuration could be utilized if MCI and AD were detected in the center part of the brain. It would also be interesting to know if the generated 3.0T* MRI images accurately depict regions known to be dementia indicators, as evaluated by a neurologist.

The implementation of Grad-CAM [37] could provide insights into which features the CNN classifier analyses to perform its classification. In addition, comparing regions of interest from Grad-CAM to that of a neurologist could be insightful for learning how to improve the classifier and the diagnosis of dementia.

6.2.4 Exploring the 0.5T Domain

This thesis utilizes datasets with clinical MRI scans from the 1.5T and the 3.0T domain, illustrating how GANs can be used for increased SNR. Doing the same work on a 0.5T dataset could lead to better insight into how valuable GANs can be in dementia diagnosing. Portable MRI scanners with a magnetic field strength of 0.5T are becoming increasingly popular. This will most likely lead to an increased amount of 0.5T datasets. The difference between 0.5T and 3.0T is far greater than the difference between 1.5T and 3.0T, resulting in a greater potential of increasing the SNR, making dementia diagnosis more accurate.

List of Figures

List of Tables

Appendix A

Code Structure

The Python files used in this thesis can be found through GitHub [1]. The MRI datasets used are only available through requesting access to data collected by the Alzheimer's Disease Neuroimaging Initiative (ADNI).

A.1 Pix2Pix

config.py - Contains the configuration options for Pix2Pix such as learning rate, number of epochs and the location of the dataset to be used.

dataset.py - Responsible for preparing NIfTI files to be used by PyTorch and applying the desired transformations.

discriminator_model.py - Consists of the architecture for the PatchGAN used in Pix2Pix.

generator_model.py - Consists of the architecture for the U-net used in Pix2Pix.

localtransforms.py - Contains transformations not included by PyTorch, such as reshaping and cropping.

train.py - The driver code of the classifier. This file also contains the configuration used for the CNN.

utils.py - A collection of methods used by several of the files mentioned above. Grouping methods allow for a cleaner code structure.

[1] https://github.com/HGodal/master-thesis

A.2 Classifier

dataset.py - Responsible for preparing NIfTI files to be used by PyTorch and for applying the desired transformations.

localtransforms.py - Contains transformations not included by PyTorch, such as reshaping and cropping.

model.py - Consists of the architecture for the CNN classifier.

train.py - The driver code of the classifier. This file also contains the configuration used for the CNN.

utils.py - A collection of methods used by several of the files mentioned above. Grouping methods allow for a cleaner code structure.

A.3 Helpers

env.yaml - Contains the python version and the packages used to conduct research. Used to create an Anaconda environment.

queue_run.sh - A bash script responsible for adding the execution of a python file to the UNIX GPU queue.

dataset_pipeline.py - Consist of several methods used in the preprocessing procedure of clinical MRI scans.

Appendix B

ROC Curves

All ROC curves created in this thesis correspond to the "one versus the rest" principle. One class is chosen as the positive class, while the remaining classes are grouped as the negative class. Having three classes results in three ROC curves. The AUC score is computed using the trapezoidal rule and provides one score for each class. All figures include a black line corresponding to randomly guessing the class distribution. The AUC score ranges from 0 to 1, with randomly guessing having a score of 0.5. Any score lower than 0.5 corresponds to a classification procedure performing worse than randomly guessing, and a score higher than 0.5 represents the opposite.

B.1 1.5T MRI Scan Dataset

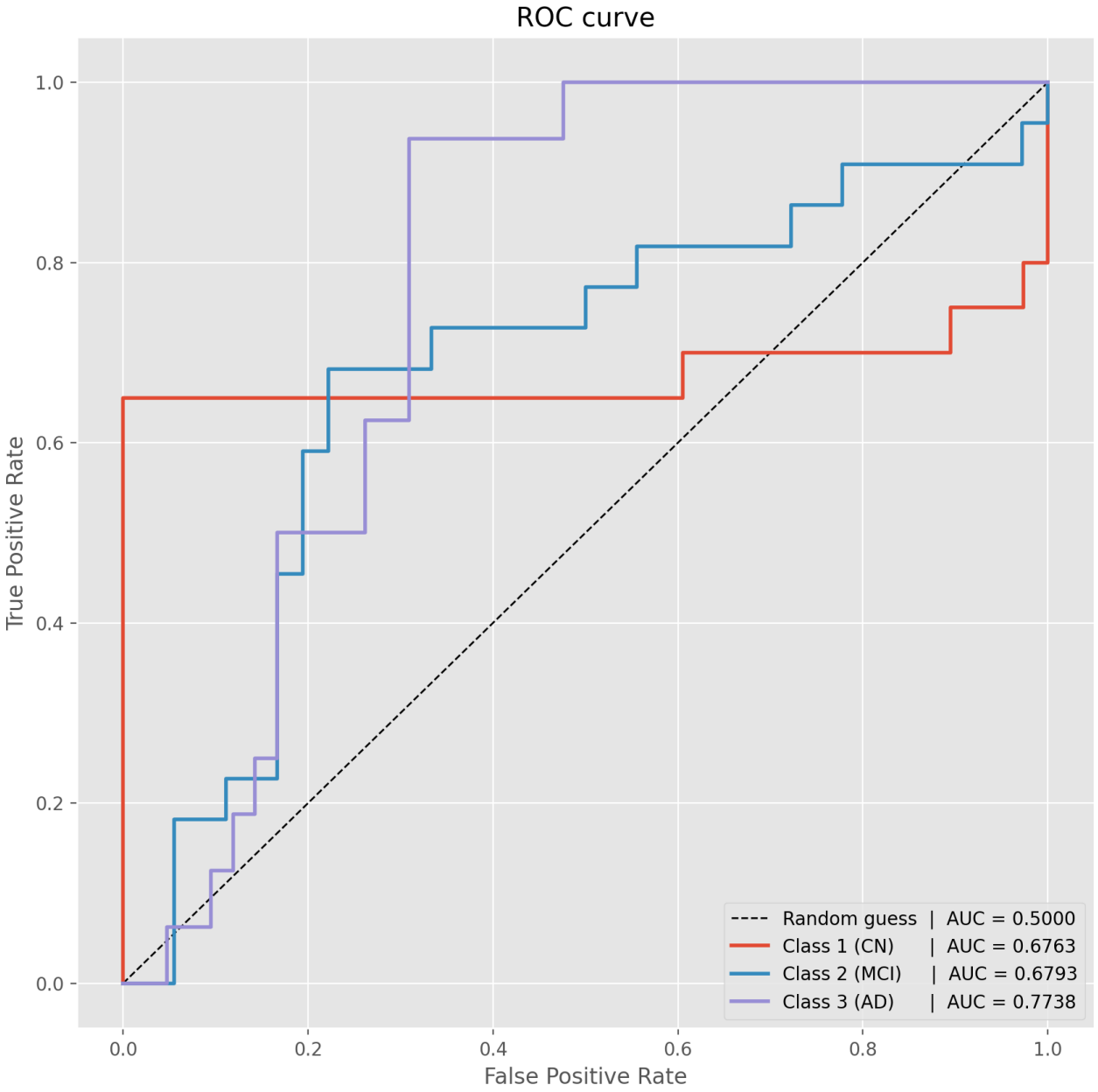

Figure B.1: ROC Curves for the 1.5T MRI Scans.

B.2 3.0T MRI Scan Dataset

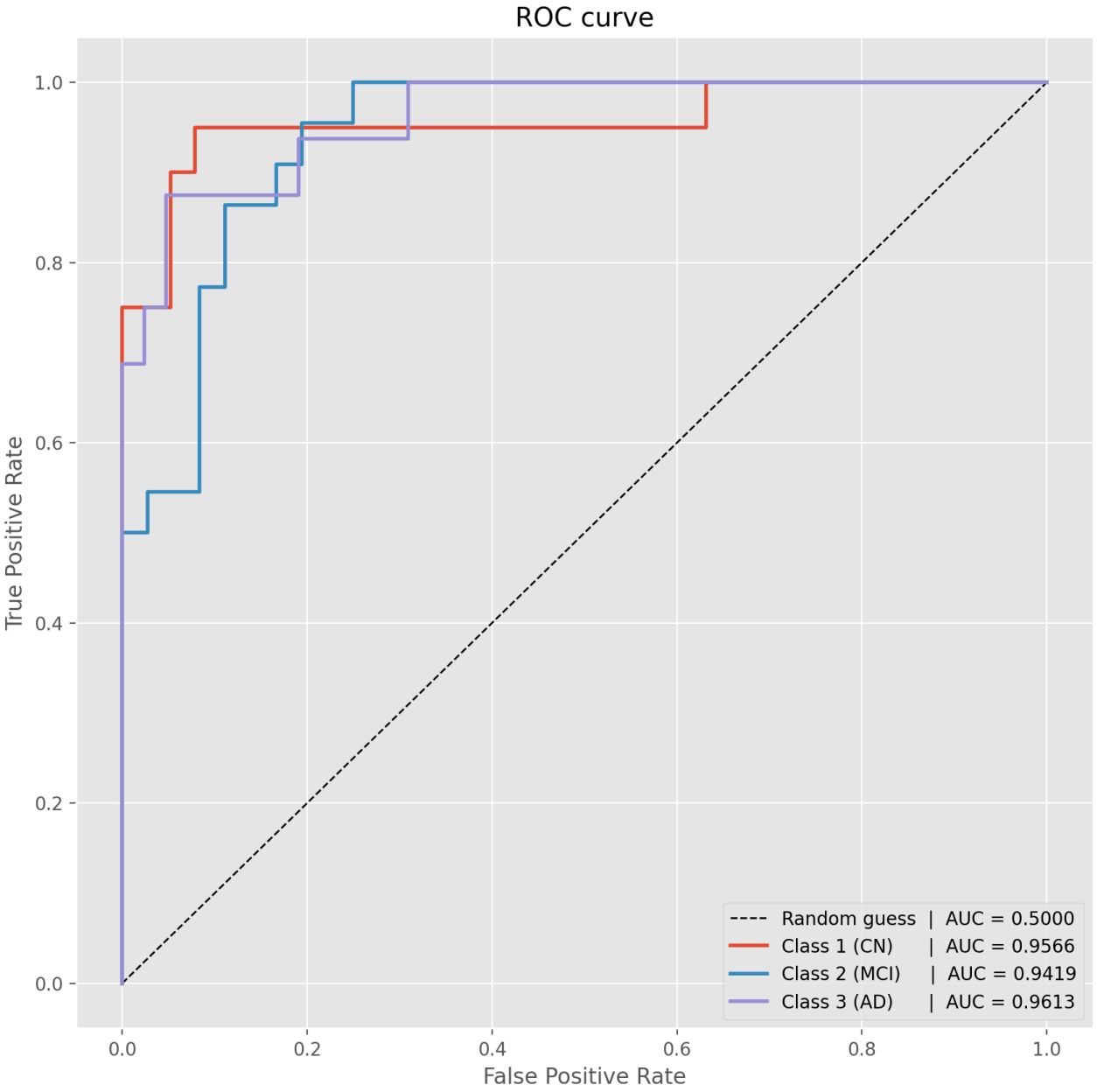

Figure B.2: ROC Curves for the 3.0T MRI Scans.

B.3 3.0T* MRI Scan Dataset

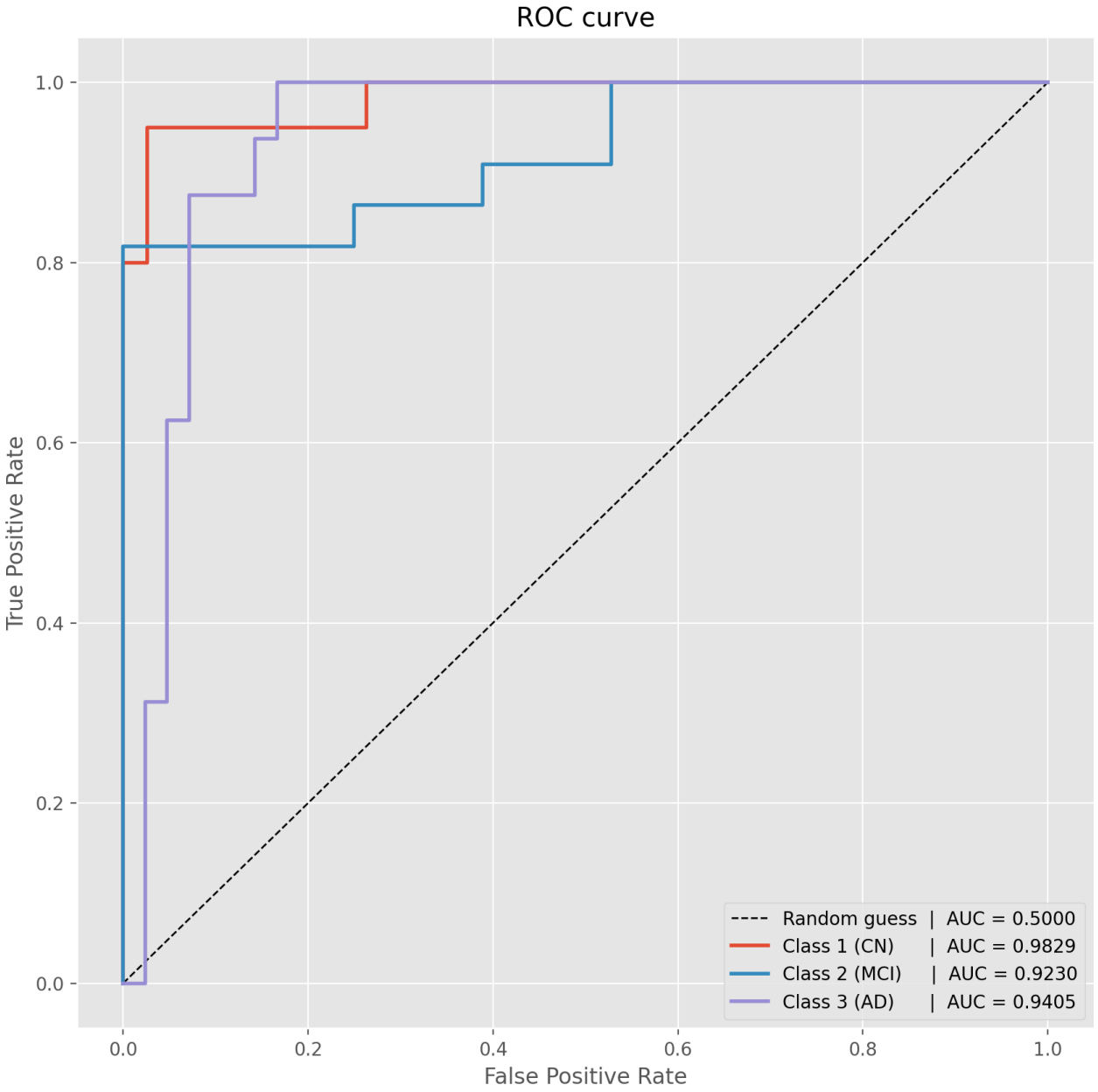

Figure B.3: ROC Curves for the 3.0T* MRI Scans.

Appendix C

Iteration Insights

C.1 Pix2Pix

C.1.1 Visual Training Evaluation

The GAN was trained for 50 epochs, corresponding to 24 hours on the UNIX GPU system, resulting in over 10000 iterations. After each epoch, a two-dimensional evaluation image was created to document and inspect the training progression. The evaluation also illustrates how some image augmentations were implemented at random.

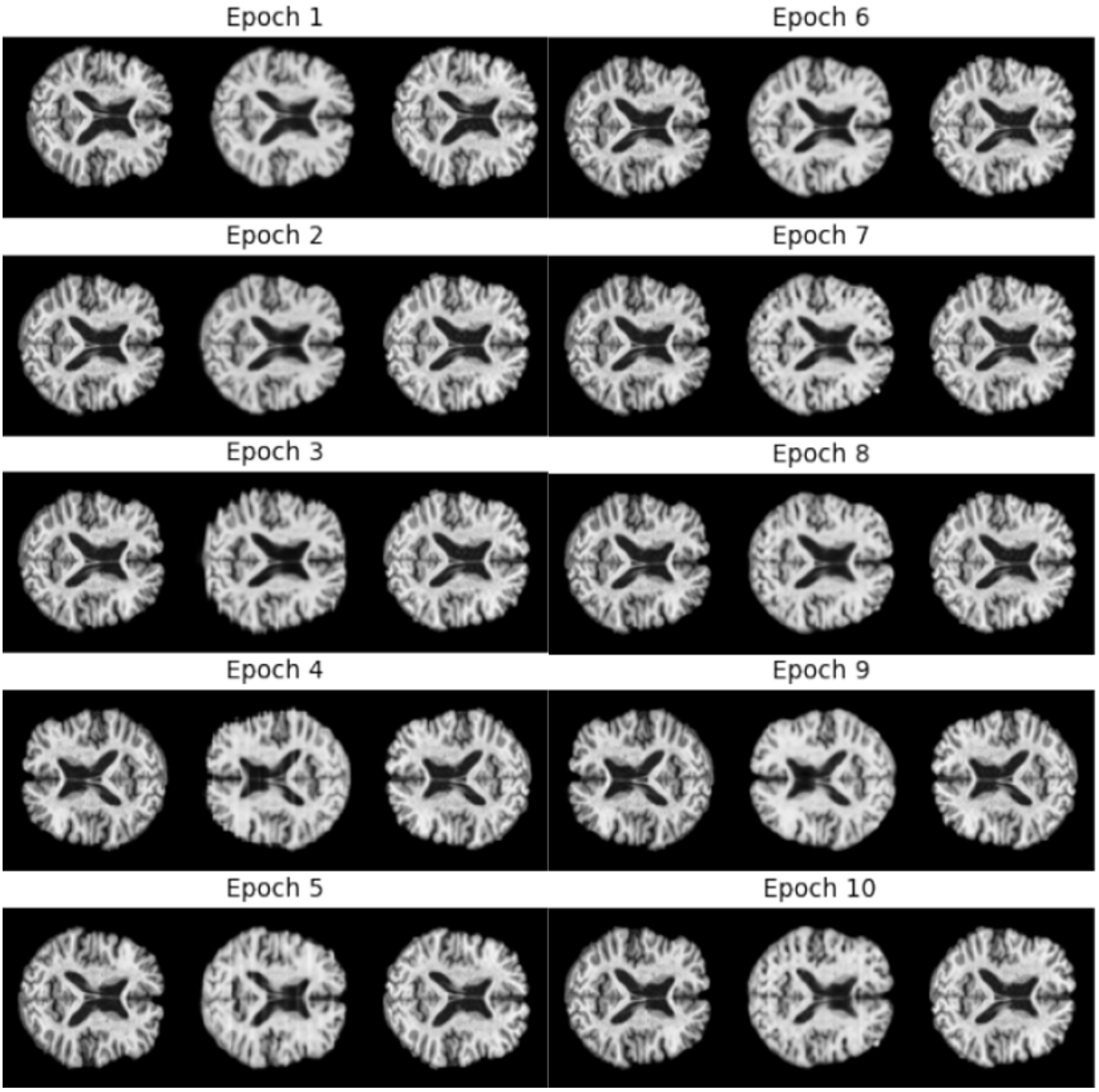

Figure C.1: Evaluation of epoch 1-10.

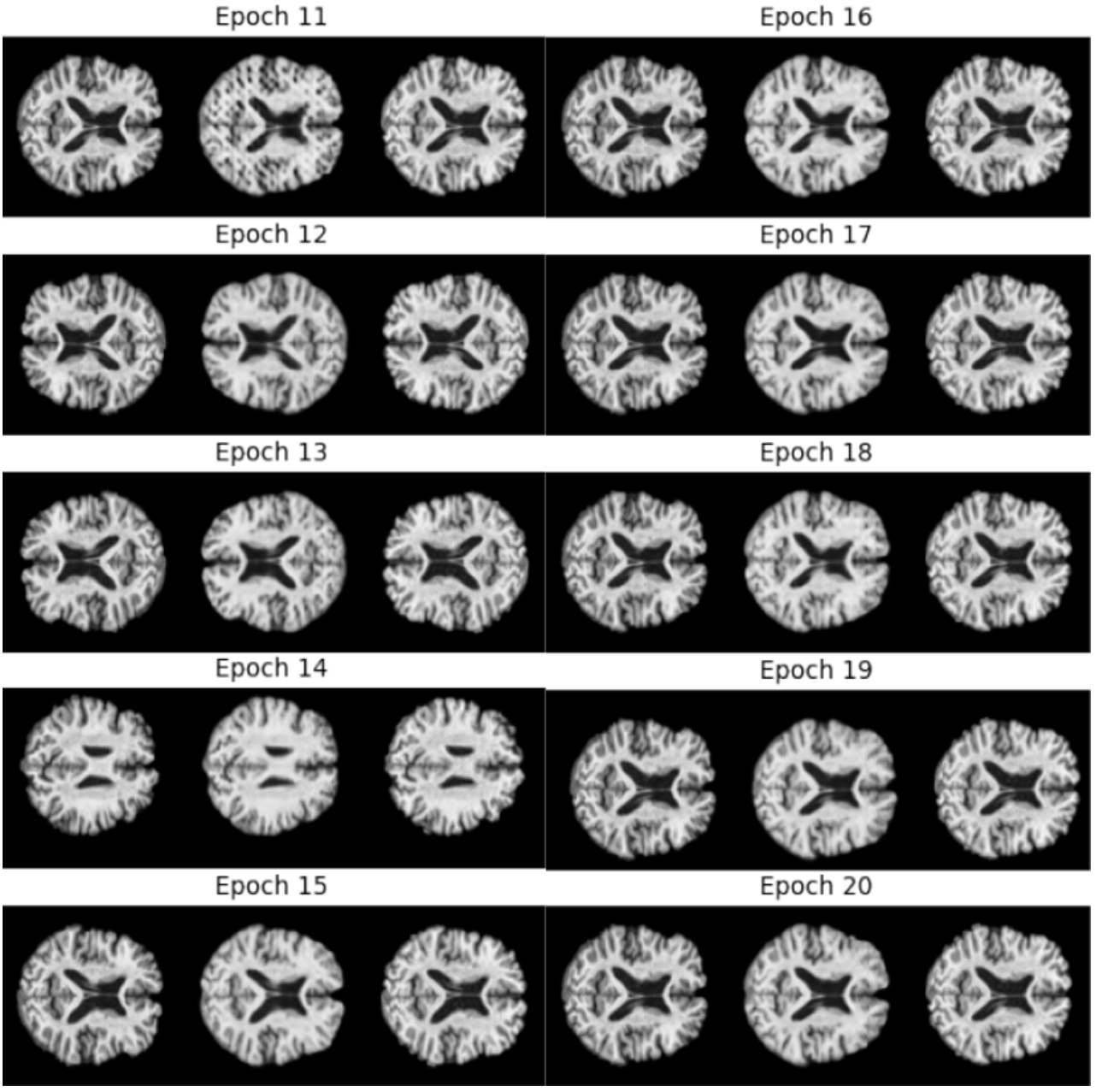

Figure C.2: Evaluation of epoch 11-20.

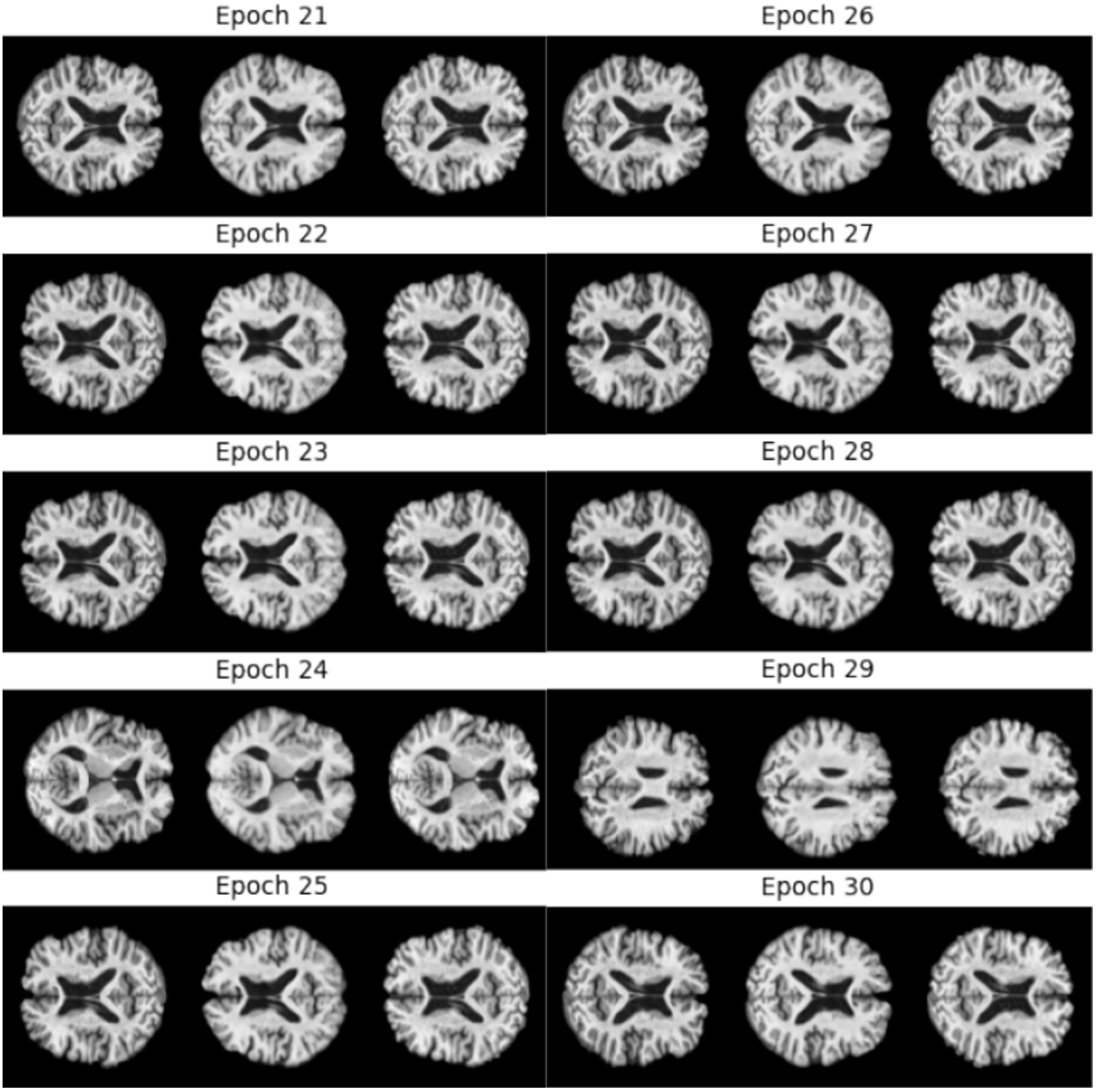

Figure C.3: Evaluation of epoch 21-30.

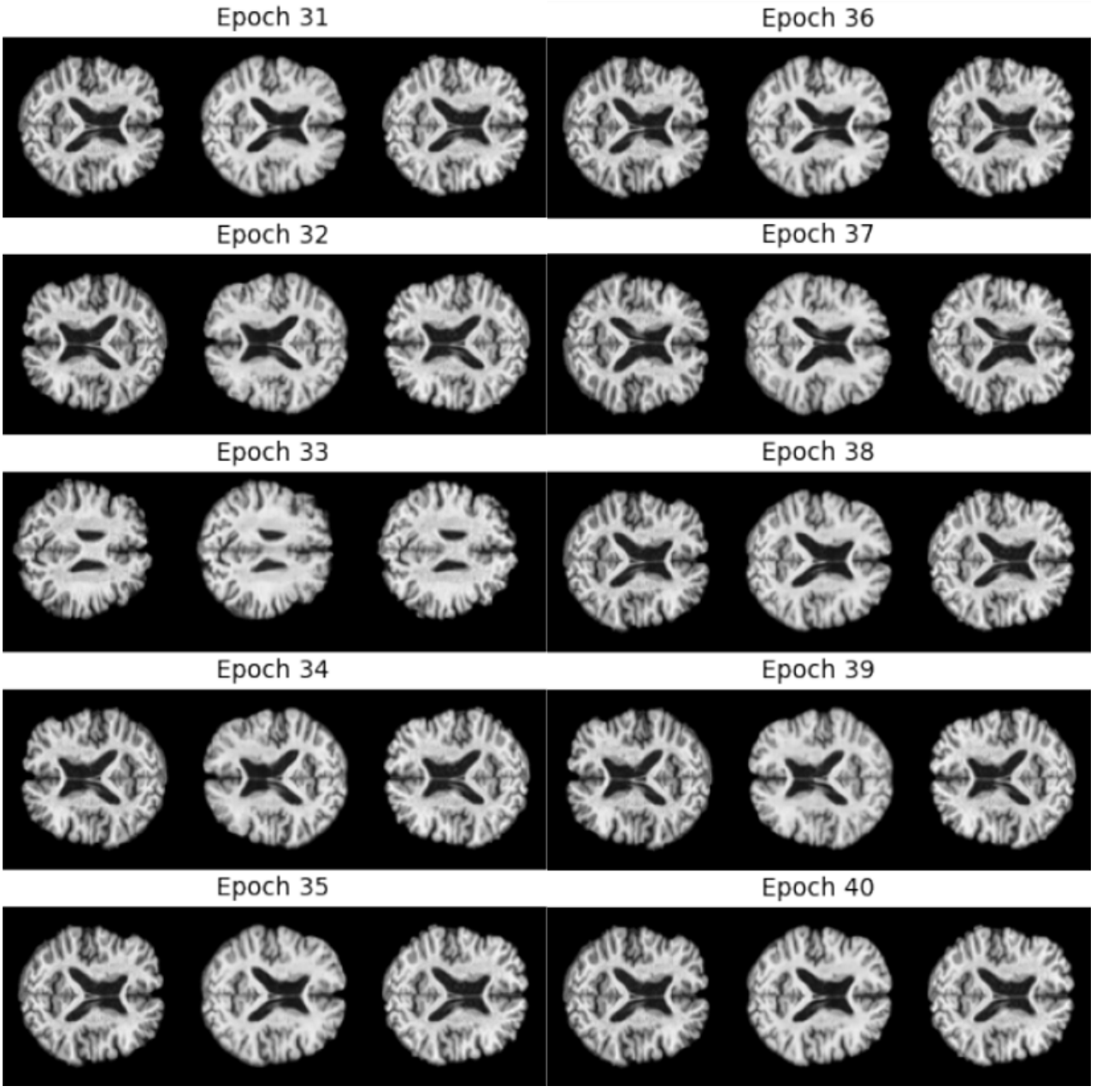

Figure C.4: Evaluation of epoch 31-40.

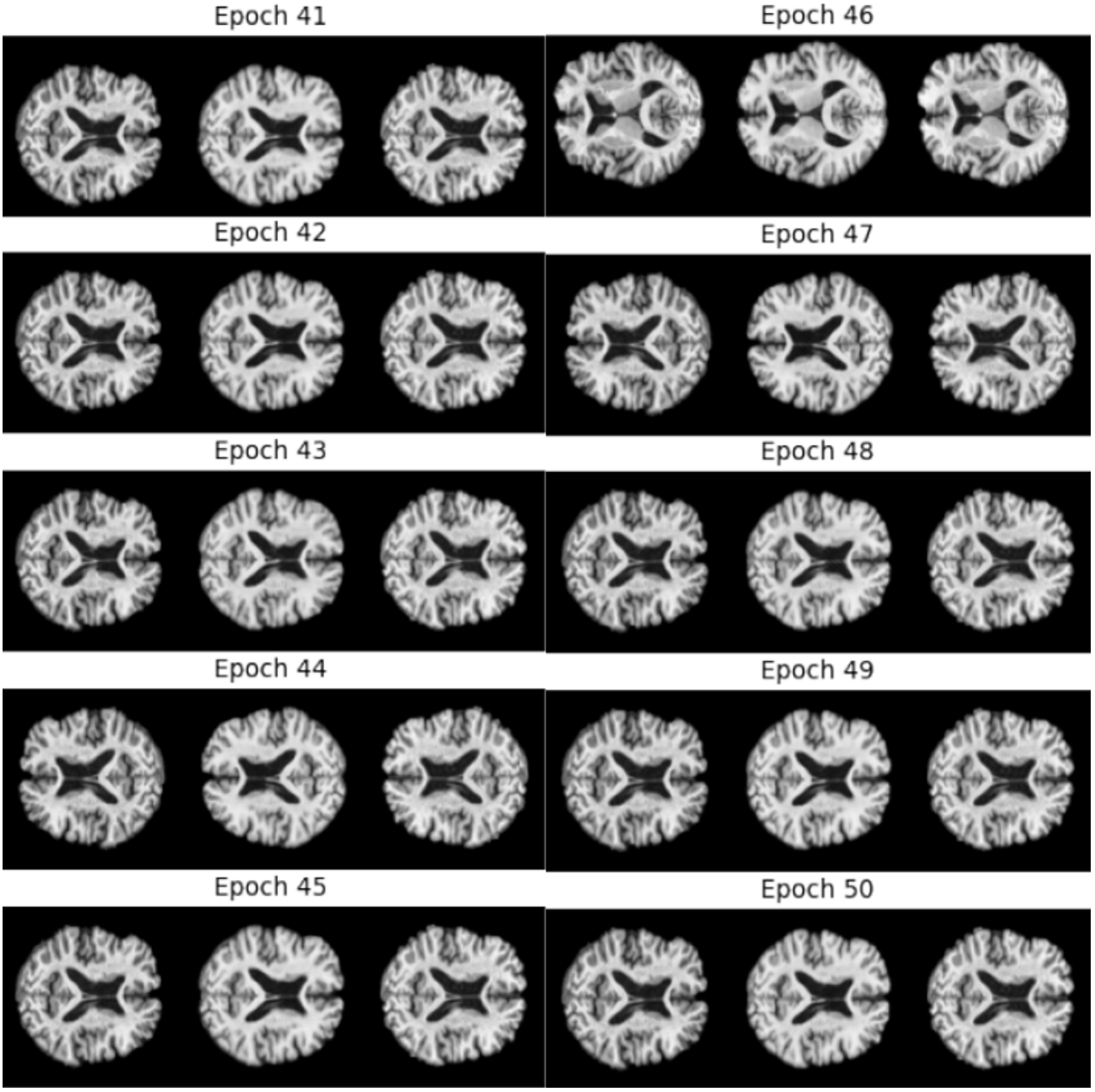

Figure C.5: Evaluation of epoch 41-50.

C.1.2 Training Loss

The loss for the discriminator and the generator was logged during the training of Pix2Pix. An exponential moving average of 0.99 was used to visualize the losses because of the vast number of training iterations.

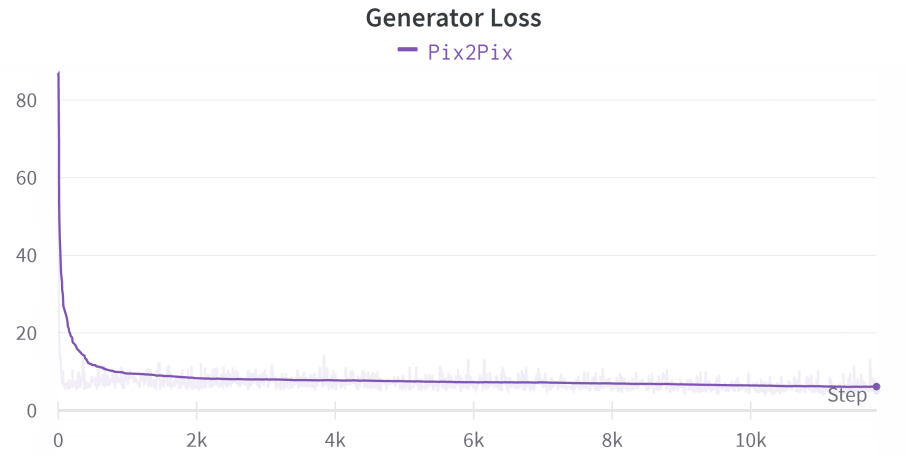

Figure C.6: Generator loss.

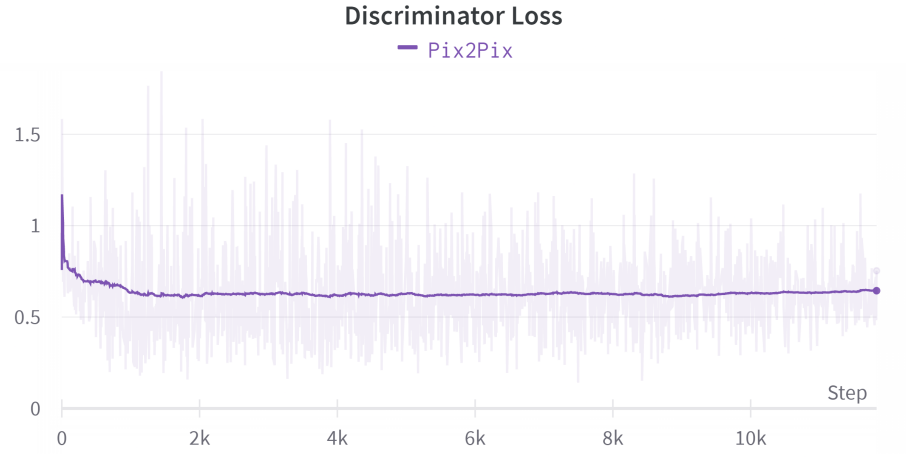

Figure C.7: Discriminator loss.

C.2 Classifier

The classifier was trained for 30 epochs, corresponding to approximately 4 hours on the UNIX GPU system. The accuracy and loss were logged during the training of the classifier for the training and validation datasets. The results from the test dataset can be found in section 4. The lines have been smoothed with an exponential moving average of 0.4 for a more straightforward interpretation.

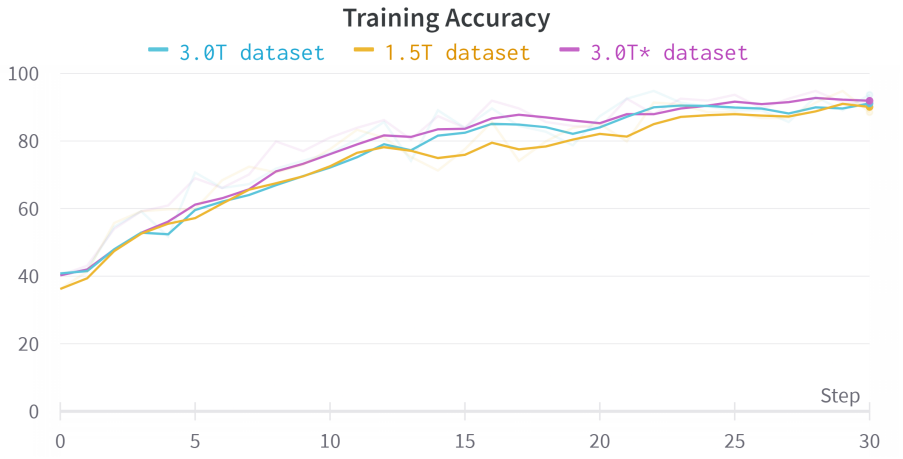

Figure C.8: Training accuracy for all datasets.

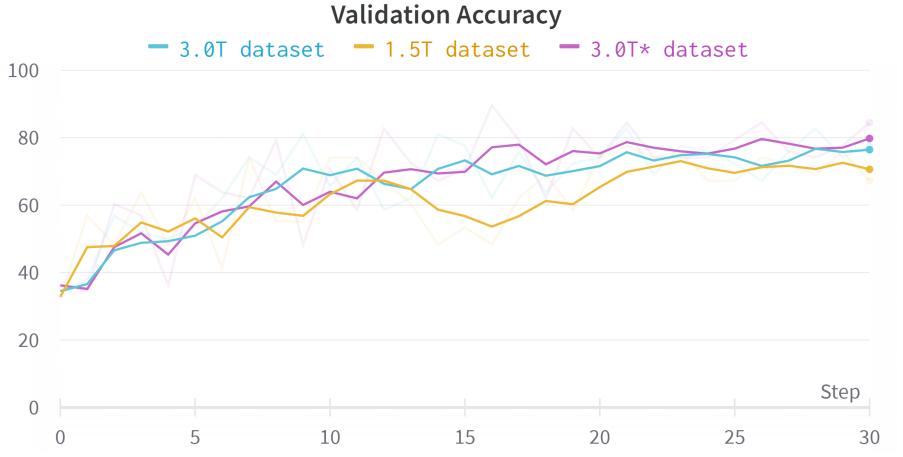

Figure C.9: Validation accuracy for all datasets.

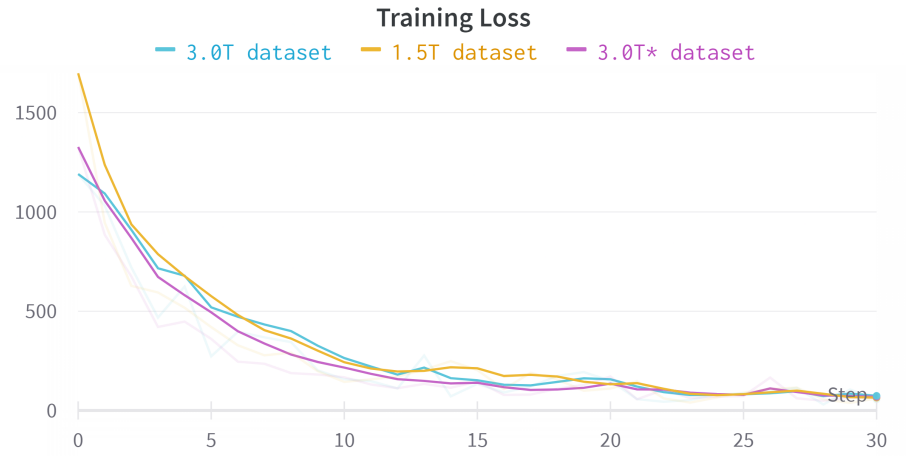

Figure C.10: Training loss for all datasets.

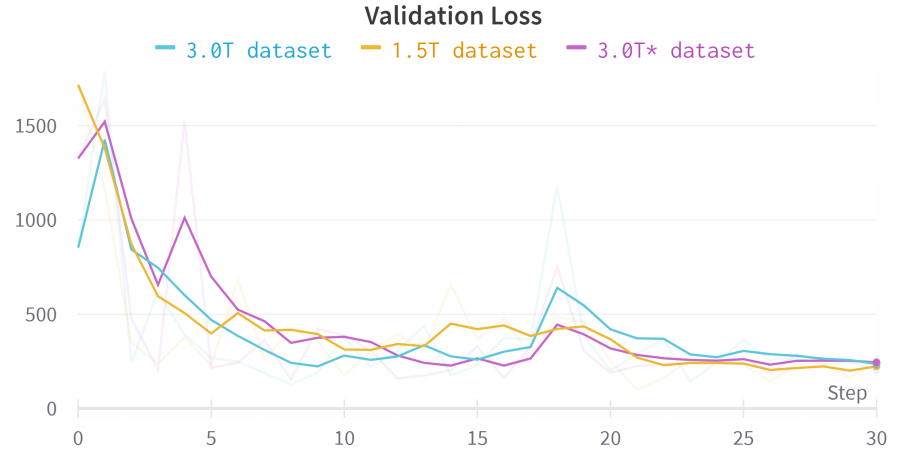

Figure C.11: Validation loss for all datasets.

Bibliography

[1] Jeffrey L. Cummings and Greg Cole. Alzheimer Disease. *JAMA*, 287(18):2335–2338, 05 2002. ISSN 0098-7484. doi: 10.1001/jama.287.18.2335. URL https://doi.org/10.1001/jama.287.18.2335.

[2] Lawrence L Wald, Patrick C McDaniel, Thomas Witzel, Jason P Stockmann, and Clarissa Zimmerman Cooley. Low-cost and portable mri. *Journal of Magnetic Resonance Imaging*, 52(3):686–696, 2020.

[3] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[4] Clifford R. Jack Jr., Matt A. Bernstein, Nick C. Fox, Paul Thompson, Gene Alexander, Danielle Harvey, Bret Borowski, Paula J. Britson, Jennifer L. Whitwell, Chadwick Ward, Anders M. Dale, Joel P. Felmlee, Jeffrey L. Gunter, Derek L.G. Hill, Ron Killiany, Norbert Schuff, Sabrina Fox-Bosetti, Chen Lin, Colin Studholme, Charles S. DeCarli, Gunnar Krueger, Heidi A. Ward, Gregory J. Metzger, Katherine T. Scott, Richard Mallozzi, Daniel Blezek, Joshua Levy, Josef P. Debbins, Adam S. Fleisher, Marilyn Albert, Robert Green, George Bartzokis, Gary Glover, John Mugler, and Michael W. Weiner. The alzheimer's disease neuroimaging initiative (adni): Mri methods. *Journal of Magnetic Resonance Imaging*, 27(4):685–691, 2008.

[5] Mark Jenkinson, Christian F Beckmann, Timothy EJ Behrens, Mark W Woolrich, and Stephen M Smith. Fsl. *Neuroimage*, 62(2):782–790, 2012.

[6] David S. Geldmacher and Peter J. Whitehouse. Evaluation of dementia. *New England Journal of Medicine*, 335(5):330–336, 1996. doi: 10.1056/NEJM199608013350507. URL https://doi.org/10.1056/NEJM199608013350507. PMID: 8663868.

[7] World Health Organization et al. Dementia. Technical report, World Health Organization. Regional Office for the Eastern Mediterranean, 2019.

[8] Anders Wimo, Maëlenn Guerchet, Gemma-Claire Ali, Yu-Tzu Wu, A. Matthew Prina, Bengt Winblad, Linus Jönsson, Zhaorui Liu, and Martin Prince. The worldwide

costs of dementia 2015 and comparisons with 2010. *Alzheimer's & Dementia*, 13(1): 1–7, 2017. ISSN 1552-5260. doi: https://doi.org/10.1016/j.jalz.2016.07.150. URL https://www.sciencedirect.com/science/article/pii/S1552526016300437.

[9] Jason Weller and Andrew Budson. Current understanding of alzheimer's disease diagnosis and treatment. *F1000Research*, 7:F1000 Faculty Rev–1161, 07 2018.

[10] Ronald C Petersen. Mild cognitive impairment. *Continuum (Minneapolis, Minn.)*, 22(2 Dementia):404–418, 04 2016.

[11] Jason Weller and Andrew Budson. Current understanding of alzheimer's disease diagnosis and treatment. *F1000Research*, 7:F1000 Faculty Rev–1161, 07 2018.

[12] Girish Katti, Syeda Arshiya Ara, and Ayesha Shireen. Magnetic resonance imaging (mri)–a review. *International journal of dental clinics*, 3(1):65–70, 2011.

[13] F Schmitt, D Grosu, C Mohr, D Purdy, K Salem, KT Scott, and B Stoeckel. [3 tesla mri: successful results with higher field strengths]. *Der Radiologe*, 44(1): 31—47, January 2004. ISSN 0033-832X. doi: 10.1007/s00117-003-1000-x. URL https://doi.org/10.1007/s00117-003-1000-x.

[14] Jamuna Kanta Sing, Sudip Kumar Adhikari, and Sayan Kahali. On estimation of bias field in mri images. In *2015 IEEE International Conference on Computer Graphics, Vision and Information Security (CGVIS)*, pages 269–274. IEEE, 2015.

[15] Hugo J Kuijf, J Matthijs Biesbroek, Max A Viergever, Geert Jan Biessels, and Koen L Vincken. Registration of brain ct images to an mri template for the purpose of lesion-symptom mapping. In *International Workshop on Multimodal Brain Image Analysis*, pages 119–128. Springer, 2013.

[16] Guangyu Robert Yang and Xiao-Jing Wang. Artificial neural networks for neuroscientists: A primer. *Neuron*, 107(6):1048–1070, 2020.

[17] Jingzhao Zhang, Sai Praneeth Karimireddy, Andreas Veit, Seungyeon Kim, Sashank J Reddi, Sanjiv Kumar, and Suvrit Sra. Why adam beats sgd for attention models, 2019.

[18] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

[19] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[20] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[21] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[22] Nahian Siddique, Sidike Paheding, Colin P Elkin, and Vijay Devabhaktuni. U-net and its variants for medical image segmentation: A review of theory and applications. *IEEE Access*, 2021.

[23] Petter Minne and Hesseberg Ruben. A study on 3d classical versus gan-based augmentation for mri brain image to predict the diagnosis of dementia with lewy bodies and alzheimer's disease in a european multi-center study. In *Medical Imaging 2022: Computer-Aided Diagnosis*, volume 12033. International Society for Optics and Photonics, SPIE, 2022. doi: 10.1117/12.2611339.

[24] Xiao Zhou, Shangran Qiu, Prajakta S Joshi, Chonghua Xue, Ronald J Killiany, Asim Z Mian, Sang P Chin, Rhoda Au, and Vijaya B Kolachalama. Enhancing magnetic resonance imaging-driven alzheimer's disease classification performance using generative adversarial learning. *Alzheimer's research & therapy*, 13(1):1–11, 2021.

[25] Merel M Jung, Bram van den Berg, Eric Postma, and Willem Huijbers. Inferring pet from mri with pix2pix. In *Benelux Conference on Artificial Intelligence*, volume 9, 2018.

[26] Guang Yang, Jun Lv, Yutong Chen, Jiahao Huang, and Jin Zhu. Generative adversarial networks (gan) powered fast magnetic resonance imaging–mini review, comparison and perspectives. *arXiv preprint arXiv:2105.01800*, 2021.

[27] Qianye Yang, Nannan Li, Zixu Zhao, Xingyu Fan, Eric I Chang, Yan Xu, et al. Mri cross-modality image-to-image translation. *Scientific reports*, 10(1):1–18, 2020.

[28] Stephen M Smith. Fast robust automated brain extraction. *Human brain mapping*, 17(3):143–155, 2002.

[29] Yongyue Zhang, Michael Brady, and Stephen Smith. Segmentation of brain mr images through a hidden markov random field model and the expectation-maximization algorithm. *IEEE transactions on medical imaging*, 20(1):45–57, 2001.

[30] Luis Perez and Jason Wang. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*, 2017.

[31] Haibo He and Edwardo A Garcia. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284, 2009.

[32] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

[33] Augustus Odena, Vincent Dumoulin, and Chris Olah. Deconvolution and checkerboard artifacts. *Distill*, 1(10):e3, 2016.

[34] Javier Quilis-Sancho, Miguel A Fernandez-Blazquez, and J Gomez-Ramirez. A comparative analysis of automated mri brain segmentation in a large longitudinal dataset: Freesurfer vs. fsl. *bioRxiv*, 2020.

[35] Bruce Fischl. Freesurfer. *Neuroimage*, 62(2):774–781, 2012.

[36] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8798–8807, 2018.

[37] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.