University of Stavanger

Faculty of Science and Technology

MASTER'S THESIS

Study program/ Specialization: **Robotics and signal processing**	Spring semester, 20.23. Open / ~~Restricted access~~
Writer: **Ørjan Vier** **Carl Henrik Hovland Christiansen**	Ørjan Vier Carl Henrik Hovland Christiansen (Writer's signature)
Faculty supervisor: **Mahdieh Khanmohammadi** External supervisor(s): **Luca Tomasetti**	
Thesis title: **Using vision transformer to synthesize computed tomography perfusion images in ischemic stroke patients**	
Credits (ECTS): 60 (2x 30)	
Key words: **Perfusion CT, Ischemic stroke, Image generation, Machine Learning, Generative adversarial nets (GAN), Vision Transformer (ViT) , HiT-GAN**	Pages: 96 + enclosure: Appendix A: 10 pages Stavanger, 15.06.2023 Date/year

University of Stavanger

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

Using vision transformer to synthesize computed tomography perfusion images in ischemic stroke patients

Master's Thesis in Robotics and signal processing

by

Ørjan Vier and

Carl Henrik Hovland Christiansen

Internal Supervisors

Mahdieh Khanmohammadi

Luca Tomasetti

June 15, 2023

"Concern for man and his fate must always form the chief interest of all technical endeavors ... Never forget this in the midst of your diagrams and equations."

Albert Einstein

Abstract

Computed tomography perfusion (CTP) imaging is crucial for diagnosing and determining the extent of damage in cerebral stroke patients [1]. Automatic segmentation of ischemic core and penumbra regions in CTP images is desired, given the limitations of manual examination. Self-supervised segmentation has gained attention [2], but it requires a large training set that can be obtained by synthesizing CTP images. Deep convolutional generative adversarial networks (DCGANs) have been used for this purpose [3], but high-resolution image synthesis remains a challenge. To address this, we propose to tailor the high-resolution transformer-based generative adversarial network (HiT-GAN) model, proposed by Zhao et al. [4], which utilizes vision transformers and self-attention mechanisms for the purposes of generating high-quality CTP data.

Our proposed model was trained using CTP images from 157 patients, categorized based on vessel occlusion. The dataset consisted of 70,050 raw data images, which were normalized and downsampled. Comparative evaluation with DCGAN showed that HiT-GAN achieved a significantly lower fréchet inception distance (FID) score of 77.4, compared to 143.0 for the DCGAN, indicating superior image generation performance. The generated images were visually compared with real samples, demonstrating promising results. While the current focus is on generating 2D images, future work aims to extend the model to generate 3D CTP data conditioned on labeled brain slices.

Overall, our study highlights the potential of HiT-GAN for synthesizing high-resolution CTP images, although its significance in advancing automatic segmentation techniques for ischemic stroke analysis is yet to be examined.

Acknowledgements

We would like to extend our heartfelt appreciation to our supervisors for their exceptional enthusiasm and invaluable assistance in the completion of this master's thesis. First and foremost, we express our deepest gratitude to Mahdieh Khanmohammadi for their continuous feedback, guidance, and unwavering support throughout the thesis. Their expertise and insightful inputs have significantly contributed to the quality and depth of this research. Additionally, we extend our sincere thanks to Luca Tomasetti for their valuable feedback and assistance during the thesis, which has greatly enhanced our work.

We are grateful to the NOBIM conference for extending an invitation to present our thesis. The conference provided us with a valuable opportunity to showcase our research and receive constructive feedback. The feedback we received at the conference has been immensely beneficial in refining our thesis and expanding our understanding of the subject matter.

Furthermore, we would like to express our sincere appreciation to the data providers at Stavanger University Hospital (SUH). Their willingness to share valuable data and collaborate with us has been instrumental in conducting meaningful research and drawing meaningful conclusions.

Finally, we would like to acknowledge the support and understanding of our families and friends throughout this academic journey. Their encouragement and belief in our abilities have been a constant source of motivation.

The completion of this master's thesis would not have been possible without the contributions and support of all these individuals and organizations. We are truly grateful for their guidance, assistance, and collaboration.

Abbreviations

CTP	**C**omputed **T**omography **P**erfusion
DCGAN	**D**eep **C**onvolutional **G**enerative **A**dvesarial **N**etwork
CNN	**C**onvolutional **N**eural **N**etwork
SUH	**S**tavanger **U**niversity **H**ospital
GAN	**G**enerative **A**dvesarial **N**etwork
HiT-GAN	**Hi**gh-Resolution **T**ransformer-based **G**enerative **A**dvesarial **N**etwork
LVO	**L**arge **V**essel **O**cclusion
NCCT	**N**on-**C**ontrast **C**omputed **T**omography
CBF	**C**erebral **B**lood **F**low
CBV	**C**erebral **B**lood **V**olume
MTT	**M**ean **T**ransit **T**ime
ViT	**Vi**sion **T**ransformer
ReLU	**Re**ctified **L**inear **U**nit
GELU	**G**aussian **E**rror **L**inear **U**nit
IS	**I**nception **S**core
FID	**F**réchet **I**nception **D**istance
NLP	**N**atural **L**anguage **P**rocessing
RNN	**R**ecurrent **N**eural **N**etwork
LSTM	**L**ong **S**hort-**T**erm **M**emory
ViTGAN	**Vi**sion **T**ransformer **G**enerative **A**dvesarial **N**etwork
MLP	**M**ulti-**L**ayer **P**erceptron
HU	**H**ounsfield **U**nit

Contents

Abstract iv

Acknowledgements v

Abbreviations vi

1 Introduction 1
- 1.1 Background and motivation . . . 1
- 1.2 Problem definition . . . 3
- 1.3 Objectives . . . 3
- 1.4 Main contributions . . . 4
- 1.5 Thesis outline . . . 4

2 Medical background 7
- 2.1 Cerebral stroke . . . 7
  - 2.1.1 Hemorrhagic stroke . . . 8
  - 2.1.2 Ischemic stroke . . . 8
- 2.2 Computed tomography . . . 9
  - 2.2.1 Computed tomography perfusion . . . 10

3 Technical background 13
- 3.1 Machine learning . . . 13
  - 3.1.1 Backpropagation . . . 15
  - 3.1.2 Activation functions . . . 15
- 3.2 Generative adversarial networks . . . 19
  - 3.2.1 Generator . . . 20
  - 3.2.2 Discriminator . . . 21
  - 3.2.3 Training GANs . . . 22
  - 3.2.4 Deep convolutional generative adversarial network . . . 22
  - 3.2.5 Progressive growing of GANs . . . 23
- 3.3 Challenges of GANs . . . 24
  - 3.3.1 Mode-collapse . . . 26
  - 3.3.2 Convergence failure . . . 28
  - 3.3.3 Vanishing gradients . . . 29

3.4 Evaluation of GANs . . . 29
3.4.1 Inception score . . . 30
3.4.2 Fréchet inception distance . . . 30
3.5 Transformers . . . 31
3.5.1 Attention . . . 33
3.5.2 Transformer architecture . . . 34
3.5.3 Contrasting running and training phases . . . 36
3.6 Vision transformers . . . 38
3.7 ViTGAN . . . 39
3.8 HiT-GAN . . . 40
3.8.1 HiT-GAN components . . . 41
3.8.2 Multi-axis blocked self-attention . . . 44
3.8.3 Cross-Attention for SelfModulation . . . 45
3.8.4 Generator architecture . . . 46
3.8.5 Low resolution stages . . . 47
3.8.6 High resolution stages . . . 48

4 Related work **49**
4.1 Introduction . . . 49
4.2 GANs for medical images . . . 50
4.2.1 DCGAN for generating synthetic CTP images . . . 50
4.3 Transformers on medical images . . . 51
4.4 Combining transformers and GANs for synthetic medical images . . . 51

5 Dataset and preprosessing **53**
5.1 Raw CTP dataset . . . 53
5.2 Preprocessed CTP dataset . . . 54
5.3 Preparation of data . . . 55
5.4 2D data . . . 55
5.5 3D data . . . 56
5.6 4D data . . . 56

6 DCGAN approach **59**
6.1 Introduction . . . 59
6.2 Experimental setup . . . 60
6.3 Experiments . . . 60
6.3.1 Results on raw data . . . 61
6.3.2 Result on preprocessed data . . . 63
6.3.3 Discussion . . . 65

7 Proposed approach **67**
7.1 Introduction . . . 67
7.2 Preparing the model . . . 68
7.3 Experiment 1 . . . 69
7.3.1 Results and discussion . . . 69
7.4 Experiment 2 . . . 79
7.4.1 Results and discussion . . . 79

8 Comparative analysis **89**
8.1 DCGAN versus HiT-GAN 90
8.2 Comparison with related work 91
8.3 Improvements 91
8.4 ViTGAN approach 92

9 Conclusions **95**
9.1 Conclusion 95
9.2 Future directions 96

A Appendix contents **97**

Bibliography **107**

Chapter 1

Introduction

This chapter will provide an exposition of the background and motivation that underlie the work conducted in this project. It will be followed by a comprehensive explanation of the research challenges. Subsequently, the thesis objectives, designed to address the problem definition, will be presented. Finally, the primary contributions of the research conducted in this project will be introduced, accompanied by a comprehensive overview of the thesis's structure and contents.

1.1 Background and motivation

A stroke, medically known as a cerebrovascular accident, occurs when there is a blockage or rupture of a cerebral artery, leading to a disruption in the brain's supply of oxygen-rich blood [5]. This critical event promptly induces cerebral ischemia, resulting in the rapid demise of brain tissue. The consequences of a stroke can be profound and varied, ranging from long-lasting disability and significant cognitive impairments to fatality. The severity and timeliness of treatment play a crucial role in determining the ultimate outcome for the affected individual, as prompt medical intervention can mitigate the extent of brain damage and enhance the chances of recovery.

In 2019, the global burden of stroke was substantial, with an estimated 113.2 million global cases reported [6]. Among these cases, 12.2 million were incident cases, indicating newly occurring instances, while the remaining 101 million cases were prevalent, representing ongoing cases. Notably, stroke accounted for 6.55 million deaths, highlighting its significant impact on mortality rates. Stroke continues to be a prominent global health concern, ranking second among the causes of death worldwide [7].

Over the past few decades, there has been a prominent increase in the prevalence of strokes. According to Feigin et al. [6], from 1990 to 2019, the number of incident cases rose by 70%, while prevalent cases witnessed an 85% increase. Additionally, stroke-related deaths experienced a 43% rise during the same period. These alarming trends underscore the growing burden of stroke, particularly in low-income countries. Addressing this public health challenge is crucial to mitigate the adverse effects of stroke and improve global health outcomes [7, 8].

Upon admission to the hospital, patients suspected of this medical condition typically undergo a standardized protocol that includes a head scan utilizing Computed Tomography Perfusion (CTP). This procedure generates comprehensive CTP data, providing a three-dimensional depiction of the cranial region. Furthermore, specialized software applications enable the conversion of this data into parametric maps, which effectively illustrate the perfusion dynamics, which is the temporal flow of blood throughout the brain [9]. Subsequently, medical professionals are tasked with carefully analyzing these maps to identify the location and extent of the infarcted region, as well as to assess its severity. This diagnostic endeavor can be intricate and time-sensitive, given that the surrounding tissue, known as the penumbra, may still contain viable and salvageable cells.

Ambitious endeavors, exemplified by the work of Tomasetti et al. (2023) [2], are pushing the boundaries of neural network applications by implementing self-supervised segmentation of the infarcted core, comprising dead tissue, from the surrounding penumbra. While these advancements hold tremendous promise for the future of diagnostic practices, it is essential to acknowledge the limitations inherent in the labeled medical datasets utilized for such complex tasks. Algorithms such as the aforementioned self-supervised segmentation demand copious amounts of annotated data to attain a level of proficiency that yields accurate outcomes. However, the availability of such data within the medical domain remains notoriously scarce and compounded by the sensitivity of the information it encompasses. These limitations will be expounded upon in the subsequent section.

Recent efforts have diligently addressed these limitations, including the notable master thesis conducted by Korkmaz et al. (2021) [3], which made significant advances in synthesizing CTP data using various techniques, notably leveraging a Deep Convolutional Generative Adversarial Network (DCGAN) model [10]. Building upon the foundation established by Korkmaz's thesis, this project aims to further advance the field by adopting the cutting-edge architecture of the vision transformer (Sec. 3.6) as a replacement for the conventional Convolutional Neural Network (CNN) [11] employed in the previous work [3]. This strategic approach is driven by the ambition to generate superior CTP data, which can be utilized for training advanced self-supervised segmentation algorithms,

among other significant applications. By embracing the vision transformer architecture, the project aspires to unlock new potentials in the realm of medical image synthesis and analysis.

1.2 Problem definition

At Stavanger University Hospital (SUH), patients suspected of ischemic stroke are routinely investigated using CTP, and parametric color-coded maps describing the blood perfusion are calculated. These maps aid in the decision on who needs immediate thrombolytic treatment and/or interventional thrombectomy and are important in saving lives and reducing the possibility of severe disability. Nevertheless, these parametric maps are far from perfect in diagnostic accuracy, and further improvement of the methods in use is needed [12–14].

It is possible to utilize deep neural networks (NN) to provide models for identifying the regions of an image that are important in terms of discriminating between patient classes, or tissue classes. However, obtaining enough training medical data for a successful neural network is usually a difficult and time-demanding task. Thus, we propose to use generative adversarial networks (GAN) with vision transformers to synthesize new CTP data to be used for training the classification networks. Given the significance of detailed information in such data, a carefully selected model that prioritizes high-resolution images has been adopted: this will be denoted as the High-resolution Transformer-based GAN (HiT-GAN) [4]. This choice ensures the capacity to capture intricate nuances and enhance the quality of the synthesized CTP data.

1.3 Objectives

The primary objective of this study is to synthesize artificial CTP data by implementing different GAN models. This endeavor aims to artificially augment datasets comprising medical data, enabling their utilization for automated tasks, as highlighted in the preceding section. A notable emphasis will be placed on the integration of vision transformers, as they are a pivotal focal point in this research. The thesis objectives are succinctly summarized in the following paragraphs.

The first approach extends the method proposed in [3], which should serve as a good baseline. It involves employing the DCGAN model to generate CTP images. This is to establish a foundation for comparison with the novel approach.

The second approach introduces a newly adopted model called HiT-GAN into the generative process. HiT-GAN incorporates vision transformers, which have shown promising results in various image-related tasks. By utilizing HiT-GAN, the goal is to enhance the quality and diversity of synthetic CTP images.

To evaluate and compare the performance of the two models, DCGAN and HiT-GAN, a comprehensive comparative analysis of the generated CTP data should be conducted. This analysis focuses on examining the characteristics, strengths, and weaknesses of each model. By thoroughly assessing the outputs from both approaches, valuable insights can be gained into their respective capabilities and areas for improvement.

1.4 Main contributions

This thesis presents several significant contributions in the field of synthesizing CTP data.

- The primary contribution is the implementation of a GAN model that utilizes vision transformers for the purpose of generating CTP images.
- In contrast to previous approaches that utilized preprocessed data [3], our approach focuses on the generation of synthetic data starting directly from raw CTP scans.
- The thesis introduces the use of the Fréchet Inception Distance (FID) metric as an evaluation tool for assessing the quality and similarity between the generated CTP data and the original training data.
- This thesis includes a comprehensive comparison between the vision transformer-based GAN model and the DCGAN model. Through various metrics and qualitative assessments, the strengths and limitations of each model are identified and discussed.

1.5 Thesis outline

The subsequent chapters of this thesis are organized to provide a comprehensive exploration of the topic. Chapter 2 delves into the relevant medical background theory, offering a thorough understanding of the underlying concepts essential for subsequent discussions. Similarly, Chapter 3 focuses on the technical background theory, equipping the reader with the necessary knowledge to delve into the technical aspects covered in the following chapters.

Chapter 4 offers a detailed analysis of previous works closely related to this thesis, providing a foundation and contextual background for the concepts and approaches discussed in this study. This chapter serves to highlight the existing research efforts that contribute to the development and understanding of the subject matter. In Chapter 5, the dataset employed for training the models in this thesis is extensively explained, along with the preprocessing steps undertaken to prepare the data for the training process. This chapter elucidates the composition and characteristics of the dataset, laying the groundwork for the subsequent experimental analyses.

Chapters 6 and 7 form the core of this thesis, presenting the experiments conducted using the DCGAN model and the HiT-GAN model, respectively. These chapters delve into the details of the experimental setup, methodology, and results obtained from each model. The outcomes and findings of these experiments are meticulously analyzed and discussed. Chapter 8 offers a comprehensive analysis of the results obtained from the experiments, providing a detailed discussion of the advantages and limitations of the two models.

Chapter 9 serves as the conclusion of this thesis, summarizing the key findings, reiterating the main contributions, and offering final insights on the research conducted. This section serves to tie together the various aspects explored throughout the thesis, culminating in a coherent and comprehensive conclusion. The chapter also outlines the direction that this project could take in future research and development.

Chapter 2

Medical background

As this project is based upon concepts and practices rooted in the field of medicine, a brief summary will be made of those that are directly associated with what this project is attempting to accomplish. This chapter will elucidate the basics of the two classifications of cerebral stroke in Section 2.1. Once these are clarified, the main diagnostic tool will be explained to give an idea of how this project is connected to the very real-world applications, described in Section 2.2.

2.1 Cerebral stroke

When diagnosing a cerebral stroke, it is crucial to discern between two primary categories: ischemic stroke and hemorrhagic stroke [5]. In essence, ischemic stroke occurs due to the obstruction or blockage of a blood vessel within the brain, resulting in restricted blood flow and subsequent tissue damage. On the other hand, hemorrhagic stroke occurs when a blood vessel in the brain ruptures, leading to bleeding and the accumulation of blood in the surrounding tissue.

Distinguishing between these two types of strokes is of paramount importance as it informs the subsequent treatment approach and management strategies. Ischemic strokes, accounting for the majority of stroke cases, necessitate interventions aimed at restoring blood flow to the affected area, such as thrombolysis or mechanical thrombectomy [15]. Conversely, hemorrhagic strokes require a distinct treatment approach, focusing on controlling bleeding, reducing intracranial pressure, and addressing the underlying cause, such as an aneurysm or arteriovenous malformation [16].

2.1.1 Hemorrhagic stroke

Hemorrhagic strokes are caused by the rupture of a blood vessel in the brain. As a consequence, the escaped blood puts pressure on the surrounding tissue causing damage. Typically, hemorrhagic strokes are subdivided into intracerebral hemorrhage, which is bleeding in the brain tissue, or the parenchyma itself, and subarachnoid hemorrhage, which is bleeding outside of the tissue, into the subarachnoid space [17].

2.1.2 Ischemic stroke

The thesis will specifically focus on ischemic stroke, which represents the predominant form of cerebral strokes, accounting for approximately 80% of the cases [18–20]. In ischemic stroke, a critical disruption of blood flow occurs, depriving a specific region of the brain of oxygen-rich blood supply. This can result from the narrowing of an artery or the formation of a blood clot that obstructs the passage of blood as depicted in Figure 2.1. Notably, ischemic stroke can be further categorized into two subgroups:

- Thrombotic stroke: This type occurs when a blood clot develops inside the cerebral arteries, leading to the blockage and subsequent reduction of blood flow.
- Embolic stroke: This is triggered by foreign particles or debris that migrates through the bloodstream, causing a blockage in the brain [21].

As a consequence, the affected brain tissue is deprived of vital nutrients and oxygen supplied by the bloodstream, leading to a rapid onset of cellular death. Timely intervention is critical to minimize permanent brain damage. Moreover, the severity of ischemic stroke is also influenced by the degree of vascular occlusion. Specifically, the lack of blood flow, known as ischemia, can occur in either large vessels (known as large vessel occlusion or LVO) or smaller vessels (non-LVO). LVO cases are often associated with increased severity and higher mortality rates [22], but they only contribute to approximately 30% of the totality of ischemic strokes [23].

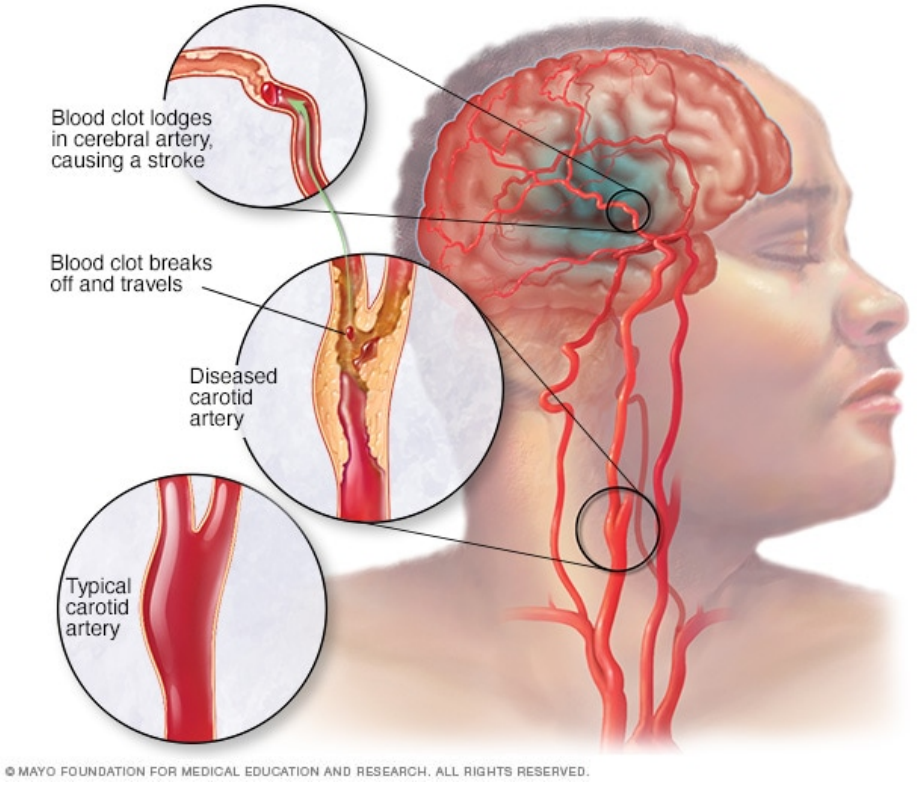


Figure 2.1: Figure illustrating what happens inside the body surrounding the events of an embolic stroke. The figure is reprinted in unaltered form from the MayoClinic.com article "What is a stroke? A Mayo Clinic expert explains" https://www.mayoclinic.org/.

The affected region of the brain in ischemic stroke can be divided into two distinct sections: the ischemic core and the penumbra. The ischemic core represents the initial area of brain tissue that is compromised and consists of irreversibly damaged cells [1, 9]. Conversely, the penumbra refers to the surrounding region that harbors potentially salvageable tissue [1].

It is imperative to restore blood flow to the penumbra in a timely manner to prevent its irreversible transition into the infarcted core [1]. Prompt identification and differentiation of these regions hold significant clinical relevance as they allow for early determination of the extent and severity of brain damage caused by the stroke. Such timely assessments are essential for informing effective treatment strategies for individuals affected by ischemic stroke.

2.2 Computed tomography

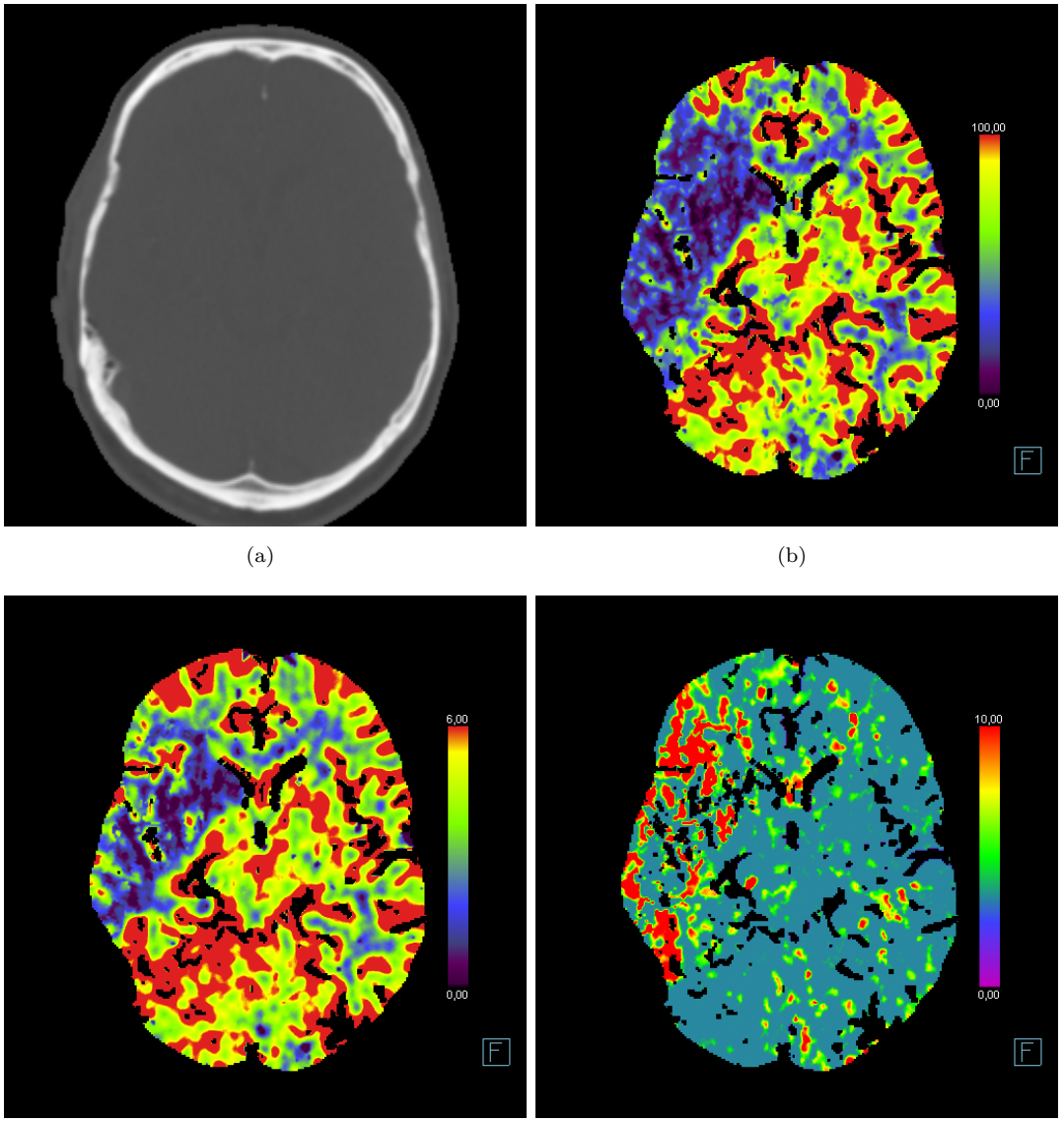
Computed Tomography (CT) serves as a vital diagnostic modality in medical imaging, commonly employed for evaluating patients presented with injuries or symptoms indicative

of disease. By employing a series of X-ray images, CT scans generate detailed tomographic cross-sectional images, offering valuable insights into the internal structures of the body. Moreover, these images can be further processed to generate color-code parametric maps, which play a crucial role in facilitating the diagnostic process. CT scans can be divided into two categories: non-contrast CT (NCCT) and contrast CT, with the latter being the focus of the thesis.

2.2.1 Computed tomography perfusion

A contrast CT procedure commences with the administration of a radiodense contrast agent to the patient. This contrast agent enhances the visibility of certain structures, particularly hollow ones like blood vessels, during the CT scan. A common and widely used contrast technique, known as CT perfusion (CTP), involves augmenting the CT study with the use of the contrast agent [24]. Subsequently, the patient is positioned within a CT scanner, which consists of a rotating X-ray mechanism and a detector array that revolves around the patient, continuously capturing tomographic images.

Through computational analysis of the acquired images, physicians gain valuable insights into the underlying tissues, organs, and blood flow within the body, with the contrast agent aiding in the visualization of blood vessels. Specifically, this data allows for the generation of parametric maps, as depicted in Figure 2.2, enabling the assessment of essential parameters such as cerebral blood flow (CBF), cerebral blood volume (CBV), and mean transit time (MTT) [9, 24]. This comprehensive evaluation proves to be particularly valuable in diagnosing patients presenting with symptoms indicative of a stroke.



(a) (b)

(c) (d)

Figure 2.2: Figure displaying (a) a CTP image and its corresponding parametric maps. Brain perfusion is demonstrated by (b) CBF, (c) CBV, and (d) MTT images. The color maps indicate red for higher values and blue for lower values.

Chapter 3

Technical background

In this chapter, a general overview of the theory behind this thesis is presented. This is to better provide a basis for understanding the technical concepts discussed further in the project.

A short introduction to machine learning will be provided in Section 3.1. In Section 3.2, GANs will be described, followed by Section 3.3 where the challenges of GAN evaluation are discussed, and Section 3.4 where some evaluation methods are described. Subsequently, Section 3.5 will introduce the transformer framework.

Within Section 3.6, the application of transformers in the domain of image processing, specifically referred to as Vision Transformers (ViT), will be introduced. Additionally, the incorporation of ViTs into a GAN model will be explored in Section 3.7. Lastly, the HiT-GAN model will be presented in Section 3.8.

3.1 Machine learning

The main focus of this project is to generate synthetic data using GANs, a concept that will be introduced in Section 3.2. To understand GANs, it is important to know the basic principle of machine learning, namely neural networks.

Neural networks, drawing inspiration from the intricate architecture of the human brain, serve as the foundational underpinning for all machine learning algorithms. They are composed of interconnected neurons, visually depicted in Figure 3.1. Notably, these neurons incorporate activation functions that may vary across different network configurations. Section 3.1.2 within this thesis will provide a comprehensive description of the specific activation functions employed.

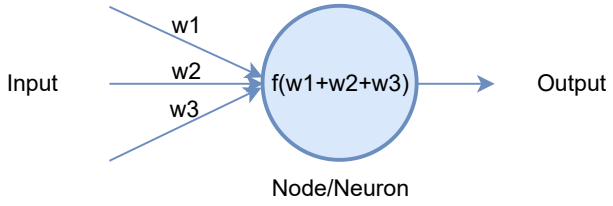


Figure 3.1: A single neuron, equipped with input weights w1, w2, and w3, undergoes an activation function to yield an output.

The inter-neuronal connections within neural networks are established through weighted connections, as exemplified in Figure 3.2. Input data, often in the form of vectorized representations, are passed through a series of hidden layers within the neural network. During the training process, the network's weights are iteratively updated through a backpropagation phase, described in Section 3.1.1. Prominent applications of neural networks encompass natural language processing [25], as well as image [26] and speech recognition [27] domains.

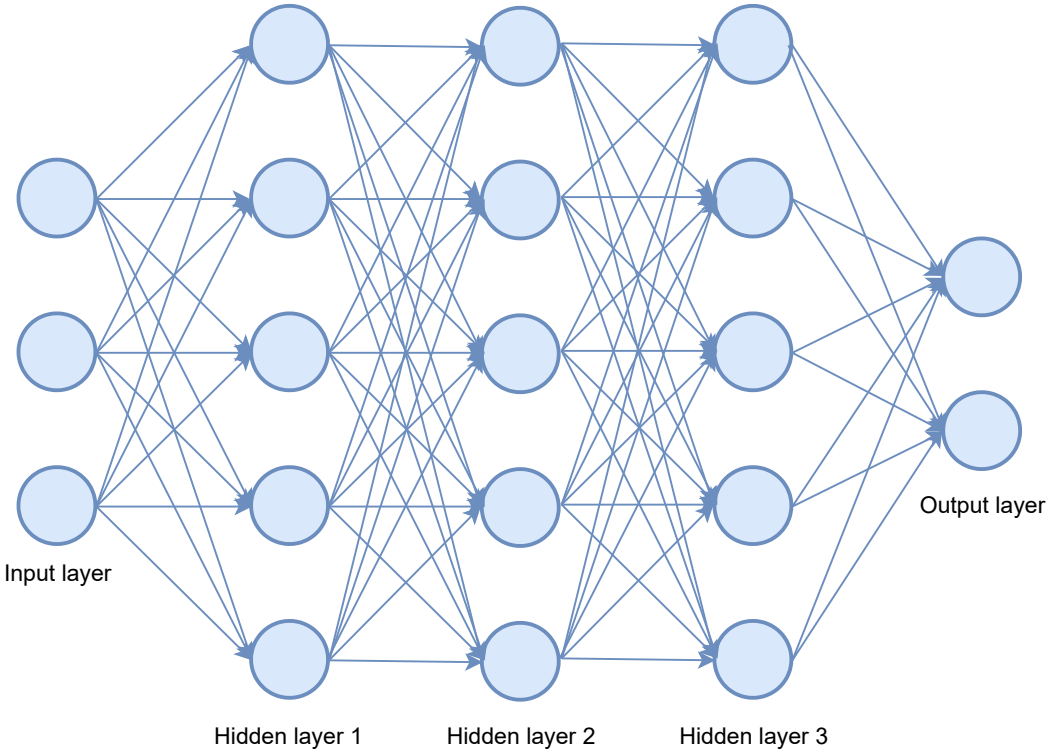


Figure 3.2: A Neural network architecture example. The network comprises an input layer (on the left), an output layer (on the right), and multiple hidden layers, the number of which depends on the depth of the network.

3.1.1 Backpropagation

Backpropagation, an essential technique in machine learning, is employed for training neural networks by computing the gradient of a specified loss function with respect to the network's weights. This gradient information enables weight updates that minimize the loss function [28]. During prediction, the neural network calculates the error between the predicted output and the actual output. Backpropagation operates by propagating this error in a backward manner, from the output layer to the input layer, and subsequently adjusting the network's weights. This iterative process is repeated multiple times until the network achieves accurate predictions.

3.1.2 Activation functions

Sigmoid

One widely used activation function in neural networks is the sigmoid function. The sigmoid function is a nonlinear bounded function that maps an input value to an output within the range of 0 and 1 [29], as shown in Figure 3.3. This characteristic renders it an ideal choice for addressing binary classification problems. The sigmoid function is shown in Equation (3.1).

$$S(x) = \frac{1}{1 + e^{-x}} \tag{3.1}$$

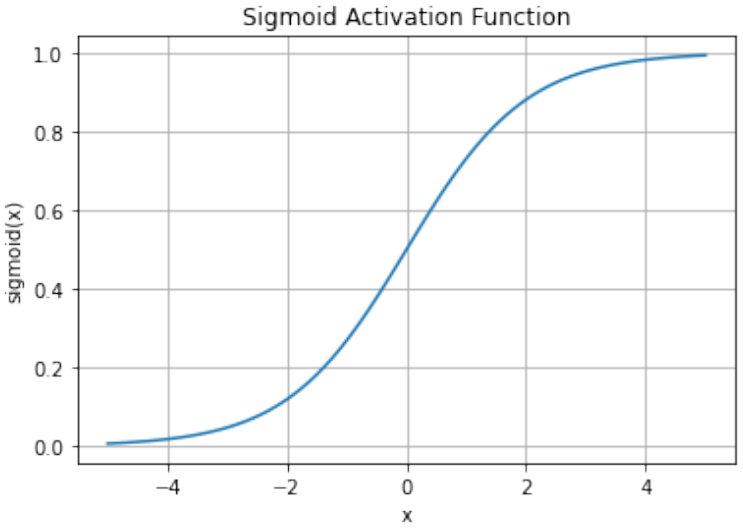


Figure 3.3: Sigmoid activation function plotted for $x = -5$ to $x = 5$ with a resolution of 100 using Equation (3.1).

Softmax

The softmax activation function, represented by Equation (3.2), is a nonlinear unbounded function utilized for transforming a vector of values into a vector of probabilities [30]. It ensures that the resulting probabilities lie within the range of 0 to 1, with their sum equaling 1 for each input vector. Figure 3.4 depicts the graphical representation of the softmax function.

$$\text{softmax}(\mathbf{x})_i = \frac{e^{x_i}}{\sum_{j=1}^{N} e^{x_j}} \tag{3.2}$$

The softmax activation function is frequently employed in classification problems, particularly in the context of multi-class classification tasks. This is primarily due to its ability to calculate the "confidence score" associated with each individual class.

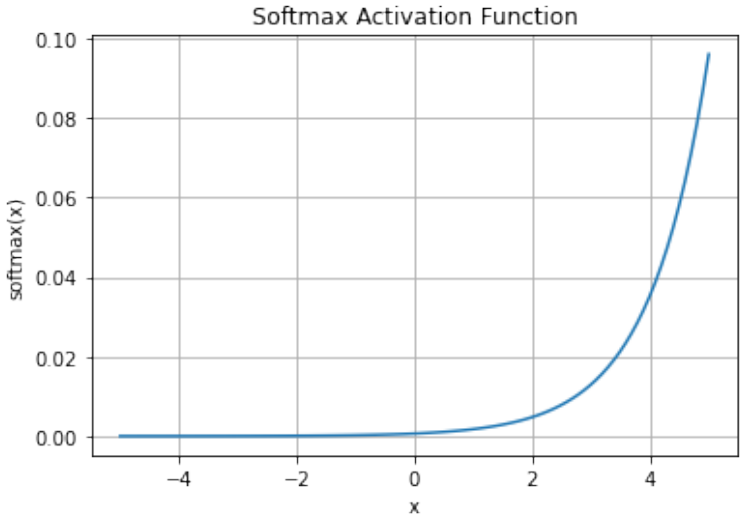


Figure 3.4: Softmax activation function plotted for $x = -5$ to $x = 5$ with a resolution of 100, using Equation (3.2).

Rectified linear activation function

The rectified linear unit (ReLU) activation function in Equation (3.3), can be regarded as an almost linear function, with the exception of a non-linearity occurring only at the point where the input value equals zero. This nonlinearity is visually demonstrated in Figure 3.5.

$$y = \max(0.0, x) \tag{3.3}$$

Due to its simplicity, the rectified linear unit (ReLU) has emerged as one of the most widely adopted activation functions in neural networks [31]. Its straightforward nature facilitates ease of model training, and it frequently outperforms alternative activation functions in terms of performance. In addition, it is less sensitive to the vanishing gradients problem (described in Section 3.3.3) in comparison to alternative activation functions. However, the utilization of ReLU introduces challenges, such as the occurrence of saturated neurons. Saturated neurons occur when the weights have an extremely high value, forcing the ReLU activation function to output a gradient of zero [32].

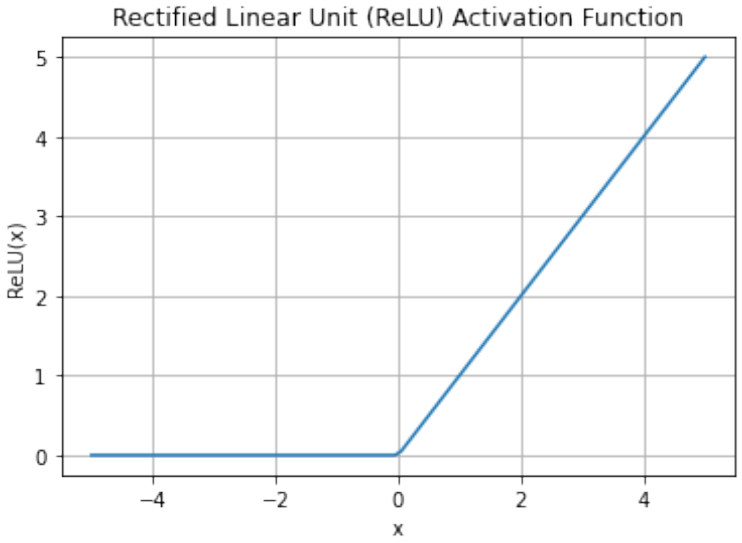


Figure 3.5: ReLU activation function plotted for $x = -5$ to $x = 5$ with a resolution of 100, using Equation (3.3).

Leaky ReLU

In order to minimize the issue of saturated neurons arising from the ReLU activation function, an alternative approach known as Leaky ReLU, as shown in Equation (3.4) can be employed. Leaky ReLU addresses this concern by introducing a small slope for negative inputs within the function, as visualized in Figure 3.6. This small slope ensures that the gradient is non-zero, thereby alleviating the saturation problem [33].

$$f(x) = \begin{cases} 0.1x, & \text{if } x < 1 \\ x, & \text{otherwise} \end{cases} \tag{3.4}$$

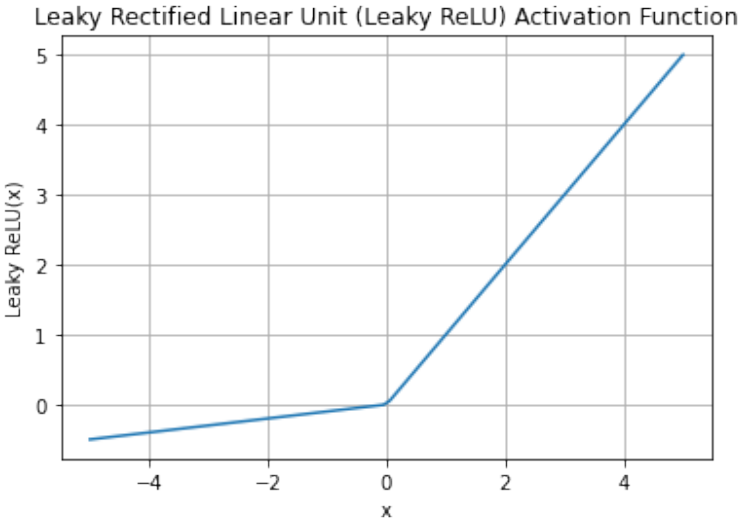


Figure 3.6: Leaky ReLU activation function plotted for $x = -5$ to $x = 5$ with a resolution of 100, using Equation (3.4).

Gaussian Error Linear Units

Gaussian Error Linear Units (GELU) was introduced in [34], where the author describes its features as: *"The GELU nonlinearity weights inputs by their value, rather than gates inputs by their sign as in ReLUs"* [34].

The GELU activation function is defined in Equation (3.5) but can be approximated as (3.6). The approximated GELU Equation is plotted in Figure 3.7.

$$\mathrm{GELU}(x) = xP(X \leq x) = x\Phi(x) = x \cdot \frac{1}{2}[1 + \mathrm{erf}(x/\sqrt{2})]. \tag{3.5}$$

$$0.5x\left(1 + \tanh\left[\sqrt{2/\pi}\left(x + 0.044715x^3\right)\right]\right) \tag{3.6}$$

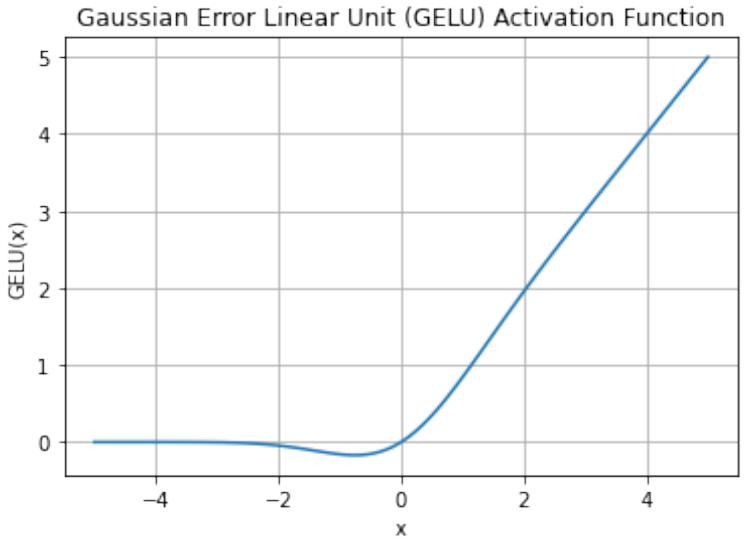


Figure 3.7: The GELU activation function plotted for $x = -5$ to $x = 5$ with a resolution of 100, using the approximated Equation (3.6).

3.2 Generative adversarial networks

Generative adversarial networks (GANs) were first introduced in the research paper "Generative adversarial networks" by Goodfellow et al. in 2014 [35]. The study aimed at generating synthetic data and reviewing the result. Although the results attained in the study were constrained, subsequent advancements in research involving GANs have yielded remarkable progress. Newer GAN architectures have demonstrated their efficacy in generating high-quality images.

GANs have many applications including image translation [36], anomaly detection [37], and image synthesis [10]. This study will focus on generating images given a set of training data with an unconditional GAN model.

The GAN model consists of two neural networks; a generator and a discriminator, as shown in Figure 3.8. The generator is as its name suggests responsible for generating images. While the discriminator grades the images on their quality and classifies them as being real or fake, based on this grade.

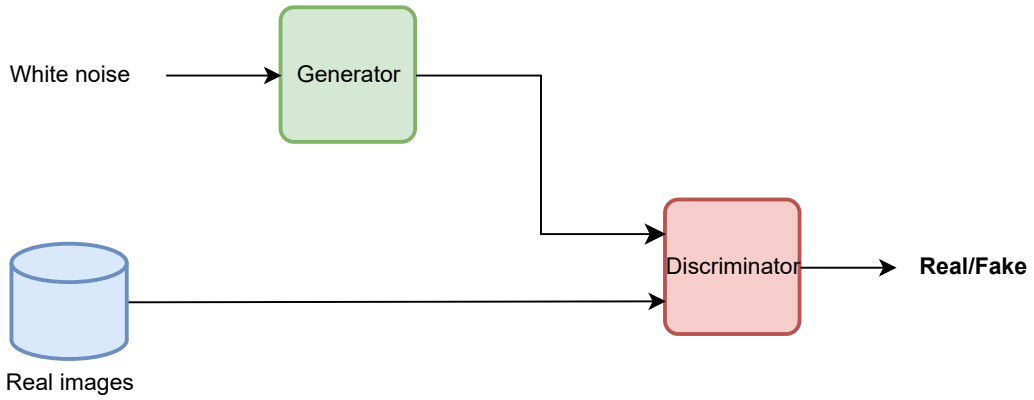


Figure 3.8: Architecture of a Generative Adversarial Network. The generator synthesizes an image, using a vector of white noise as input, which is then presented to the discriminator. The discriminator's task is to differentiate between fake generated images produced by the generator and real images from the dataset.

Both of these neural networks are working against each other, trying to fool the other, hence the "adversarial" attribution. Although none of the networks are especially good at the beginning of the training phase, the idea is that they will learn from each other until the generator is good enough to fool the discriminator a considerable amount of times. When that happens, the generator is able to generate images that are remarkably similar to the images from the training set, which is the goal of training GANs. The discriminator can then be removed and the generator can be used to produce images that are similar to the dataset samples. Because the generated images are made from white noise, the images will differ a bit, ensuring diversity in the synthetic images.

Two main categories of GANs exist, namely conditional and unconditional GANs. Conditional GANs offer the means to exert control over the generation process. This control is achieved by incorporating a label corresponding to different classes into both the latent "z" vector before the generator and the real images in the discriminator. By doing so, the model becomes capable of recognizing and generating specific classes as instructed. Unconditional on the other hand offers no control over the generated output.

3.2.1 Generator

The generator is a neural network, often a convolutional neural network regarding image synthesis, used to learn the data distribution such that it can generate data that is close to the real data, trying to maximize the probability of the discriminator making a mistake. It takes in a vector of white noise (visualized in Figure 3.9), often called latent z, and uses the discriminator's feedback to improve its generated images. As training

continues, the generator will get better at reproducing the images from the dataset it has trained on.

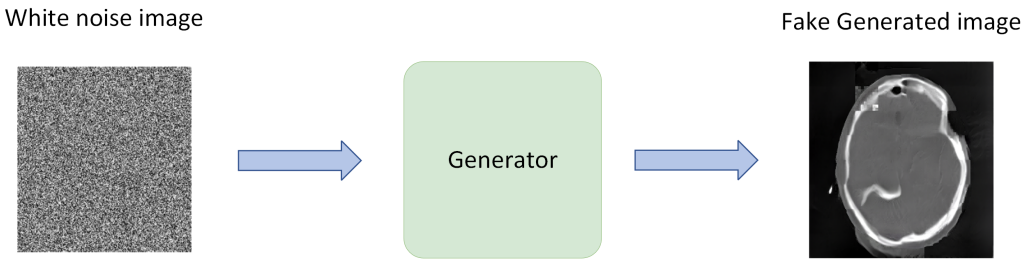


Figure 3.9: The Generator takes a white noise image or vector (white noise image is used in the figure for better visualization) and generates an image based on what it has learned from the discriminator's feedback.

3.2.2 Discriminator

Similar to the generator, the discriminator also constitutes a neural network, but different from the generator the discriminator inputs both fake images from the generator and real images from a dataset, as displayed in Figure 3.10. The discriminator's task is to distinguish between real images and images generated by the generator.

While the generator learns how to generate better fake images, the discriminator learns how to distinguish between real and fake images.

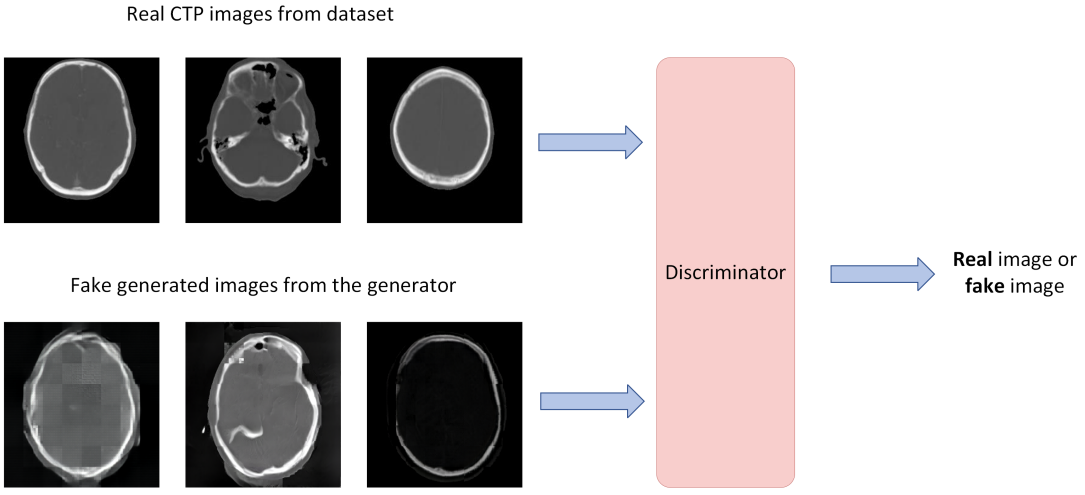


Figure 3.10: Real images and fake images from the generator get classified as fake or real by the discriminator, often as a binary classification.

3.2.3 Training GANs

The training of GANs can pose greater challenges compared to conventional neural network training. This is primarily attributed to the inherently adversarial nature of GANs, where two models engage in a min-max game, constantly competing against each other. Maintaining a good balance between the generator and discriminator models throughout the training process can be difficult, much due to the output of one model impacting the other through the backpropagation phase, as shown in Figure 3.11. During the training process, only one model is activated at a given time, forcing the other model to remain constant.

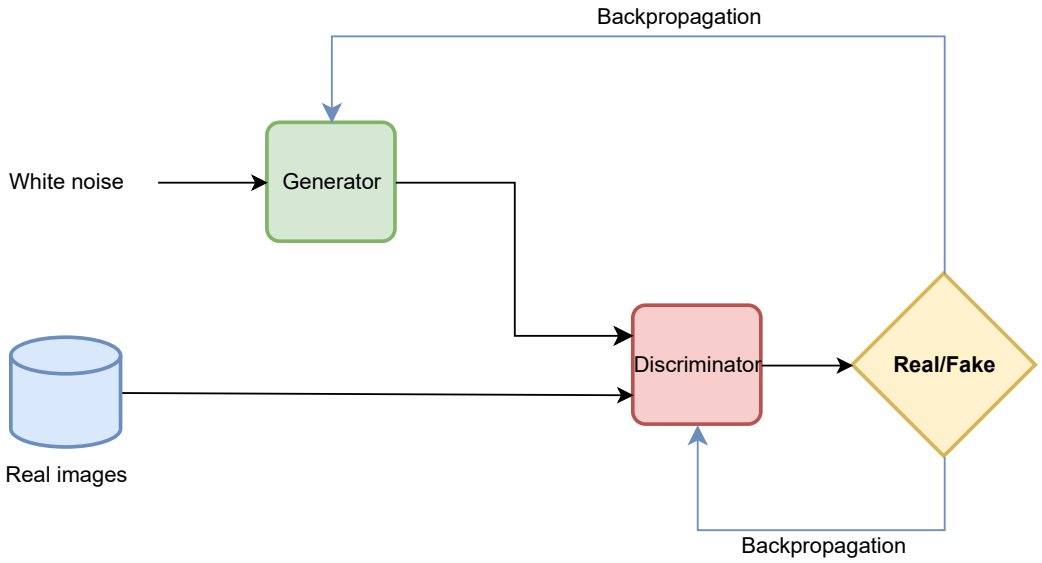


Figure 3.11: A visualization on how a GAN model is trained using backpropagation.

3.2.4 Deep convolutional generative adversarial network

Different types of GANs exist in the literature, but in this thesis, Deep Convolutional Generative Adversarial Networks (DCGANs) will be covered.

The DCGAN was proposed in 2015 by Radford et al. [10]. It was developed as a methodology to enhance the stability of GAN training, as GAN models were known for their inherent training instability, or as stated by the authors themselves: *"We propose and evaluate a set of constraints on the architectural topology of Convolutional GANs that make them stable to train in most settings. We name this class of architectures Deep Convolutional GANs (DCGAN)"* [10].

The way the authors accomplished it can be summarised in 4 steps:

- All spatial pooling functions were replaced with strided convolutions for the discriminator and fractional-strided convolutions for the generator, like in [38]. This can be seen in Figure 3.12, which makes the model able to learn its own spatial downsampling.
- The fully connected layers on top of convolutional features were removed.
- Batch normalization [39] was applied for both the generator and the discriminator. This method helps the gradient flow in deeper models and involves normalizing the input to each unit within a batch to have a zero mean and unit variance.
- The ReLU activation function (discussed in Section 3.1.2) was applied for the generator, except for the output layer, and the Leaky ReLU (Section 3.1.2) activation function was applied for the discriminator.

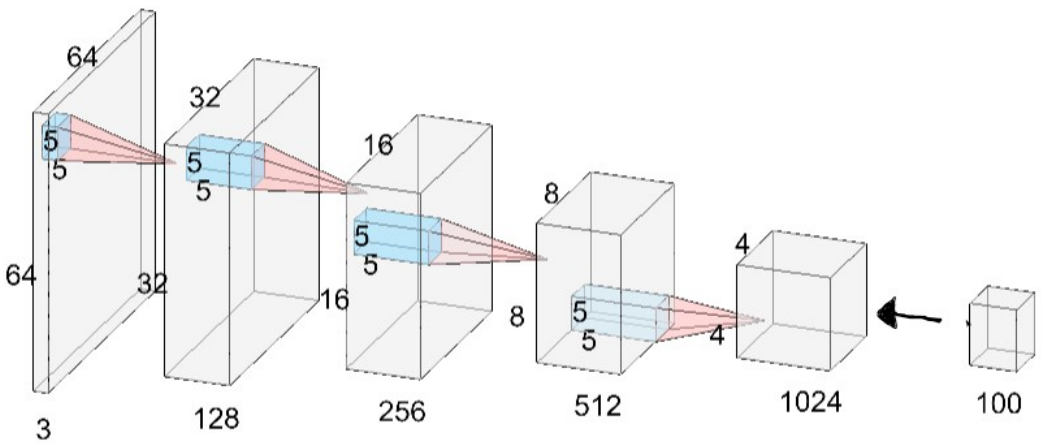


Figure 3.12: Visualization of the fractionally-strided convolution layers of the DCGAN generator. Starting with a 100x1 vector input, the data undergoes processing through four fractionally-strided convolution layers and gets converted to a 64x64 pixel image, with no use of fully connected or pooling layers.

3.2.5 Progressive growing of GANs

Progressive growing of GANs is when the discriminator and generator models employ a training technique known as progressive training, as described in [40]. The progressive training approach begins with low-resolution images and iteratively refines the model until it reaches a stable state. Subsequently, the image area is quadrupled. This process is called one block and is repeated multiple times until the desired image size is achieved, as shown in Figure 3.13. This method speeds up and stabilizes the training process [40].

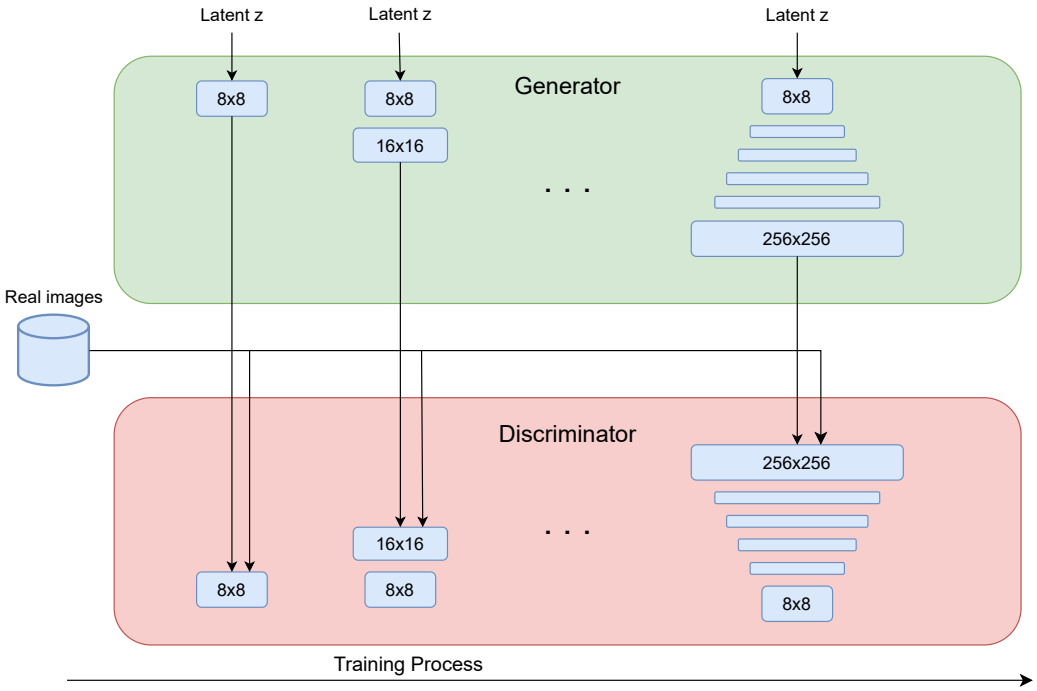


Figure 3.13: The training process begins with a low 4x4 pixel resolution for both the generator (G) and the discriminator (D). Through the training process layers get added to both (D) and (G) with an increased resolution. The figure is adapted from [10].

3.3 Challenges of GANs

Training GANs are widely acknowledged as a challenging task. The inherent nature of GANs, where two neural networks engage in a competitive process, renders them vulnerable to imbalanced training. For instance, the discriminator network may significantly outperform the generator network, impeding the progress of the generator's training and rendering it ineffective or useless. Multiple issues need to be addressed when training GANs to ensure their optimal performance and stability.

A commonly employed approach to identify problems during GAN training involves monitoring the discriminator loss for real and fake images, as well as the generator losses. The "discriminator real loss" represents the discrepancy between the discriminator's predictions for real images and the corresponding ground truth. Conversely, the "discriminator fake loss" quantifies the disparity between the discriminator's predictions for generated fake images and the ground truth. On the other hand, the "generator loss" pertains to the dissimilarity between the generated output and the real images.

To be able to identify problems in the GAN training, it is important to know how a stable GAN model behaves. An example of such a GAN is displayed in Figure 3.14. As can be extracted from the figure, which illustrates the losses that are calculated during training,

there is a long period of instability where both networks are quite poor, before they slowly converge to optimal values, where the discriminator losses should be at around 0,5 and the generator loss should be between 0,5 and 2,0 according to [41]. The variance may be quite high, but the convergence indicates that the output images should be very satisfactory.

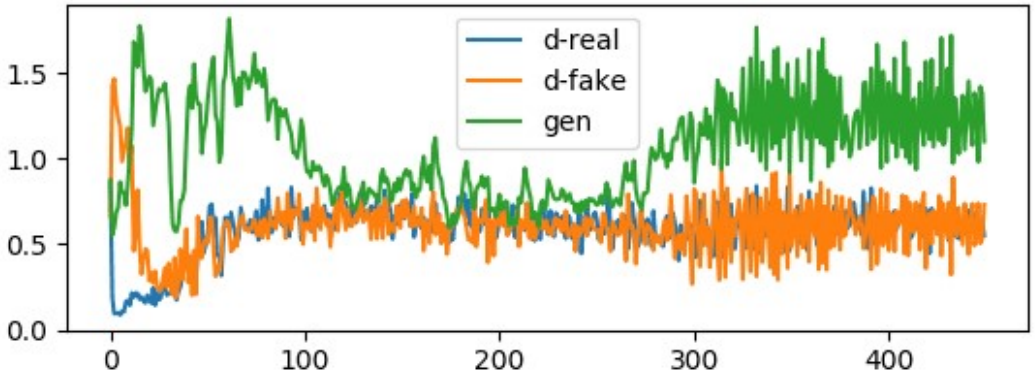


Figure 3.14: Discriminator real loss (blue-line), discriminator fake loss (orange-line), and generator loss (green-line) for a stable GAN plotted with respect to epochs. The following experimental figures are reprinted in unaltered form by permission from Jason Brownlee, [41].

From Jason Brownlee's experiment on stable GANs in [41], the values obtained in epoch 180 in Figure 3.14, resulted in the generated outputs in Figure 3.15.

Figure 3.15: Generated images from epoch 180 by a stable GAN model, resulting in the output images having high diversity. The experiment was conducted by Jason Brownlee in [41] using the MNIST dataset.

3.3.1 Mode-collapse

According to Ian Goodfellow, the author that introduced the GANs in [35], *"mode collapse is a problem that occurs when the generator learns to map several different input z values to the same output point."* Put in different terms, the mode collapse problem occurs when a GAN generates images in only a few groups or modes, resulting in low diversity between the images.

Two types of mode collapse exist: complete and partial mode collapse. In a "complete mode collapse", a number of different values of the latent z input are mapped to the same output, making the model only generate images in a few modes, as illustrated in Figure 3.17. A much more common mode collapse is "partial mode collapse". Partial mode collapse refers to a model that generates images with the same color pattern, and the same motive just from a different angle.

A mode collapse can be identified visually by looking at a big set of generated images. There will be low diversity between the images and the images will repeat themselves, as presented in Figure 3.17. It is also possible to identify a mode collapse from the model's

loss of data. The loss of data, especially the generator loss, will oscillate over time due to different losses in the different modes [41]. This is shown in Figure 3.16.

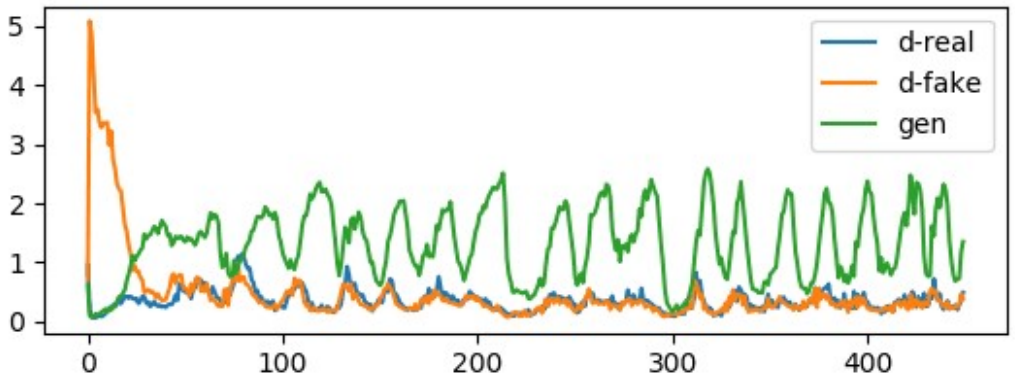


Figure 3.16: Discriminator real loss (blue-line), discriminator fake loss (orange-line) and generator loss (green-line) for a mode collapse, plotted in respect to epochs in an experiment done by Jason Brownlee in [41].

The values for epoch 315 in Jason Brownlee's experiment on mode collapse in Figure 3.16, resulted in the output images in Figure 3.17.

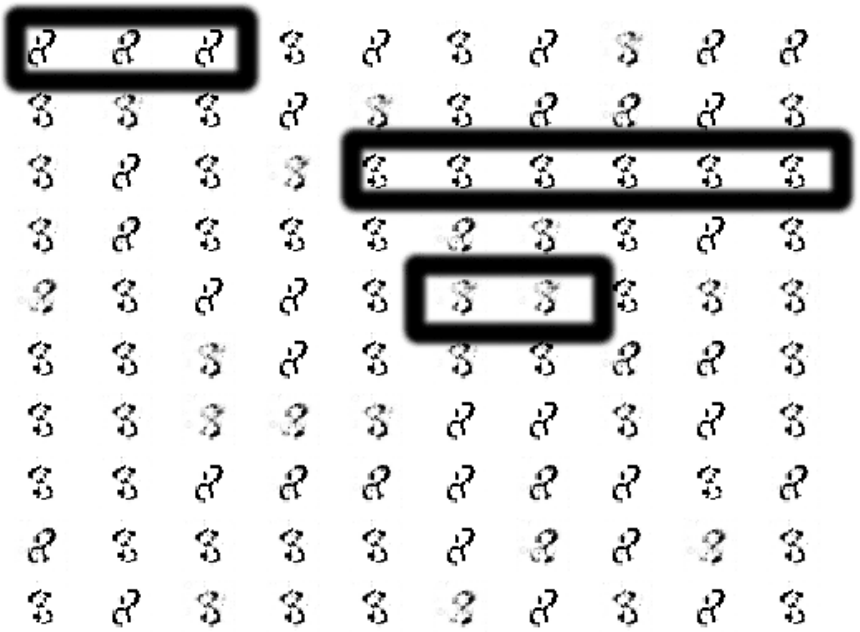


Figure 3.17: Generated images from epoch 350 by a GAN model suffering from mode collapse, resulting in lower diversity. The experiment was conducted by Jason Brownlee in [41].

3.3.2 Convergence failure

Convergence failure is a common problem when training GANs, and refers to when the model does not find an equilibrium between the discriminator and the generator. The way to identify a convergence failure is to monitor the discriminator's loss. If the loss converges to zero or close to zero it is most likely a convergence failure, this specific case is shown in Figure 3.18. It is also common that the generator loss increases during convergence failure as in Figure 3.19. Visually the output images can look something like in Figure 3.20.

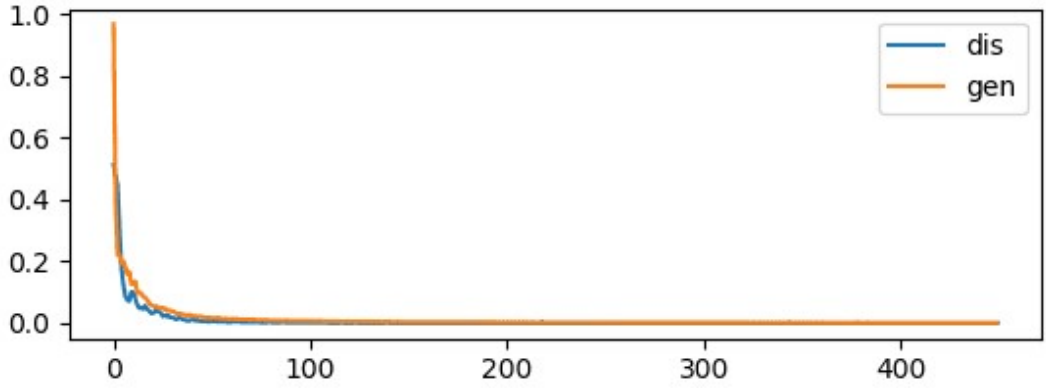


Figure 3.18: Discriminator loss (blue-line), and generator loss (orange-line) for a convergence failure, plotted in respect to epochs in an experiment done by Jason Brownlee in [41].

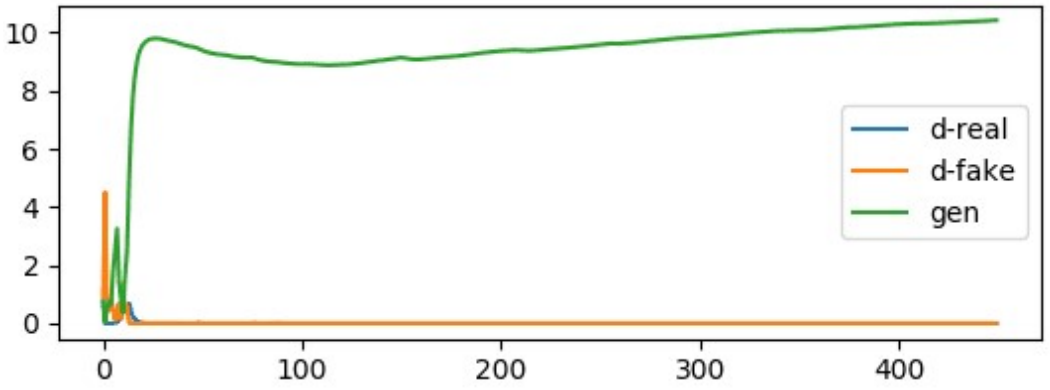


Figure 3.19: Discriminator real loss (blue-line), discriminator fake loss (orange-line) and generator loss (green-line) for a convergence failure, plotted in respect to epochs in an experiment done by Jason Brownlee in [41].

The reason for convergence failure is that the discriminator becomes superior to the generator, making the feedback given to the generator useless. This is because the discriminator is nearly 100% sure the image created by the generator is fake, denying the nuances in the feedback that the generator needs in order to get better. A convergence

failure can appear at the beginning of the training, but also after a few epochs. Usually, GANs do not recover from this failure, although some unstable GANs are able to [41].

From the experiment in Figure 3.18 the resulting output images are shown in Figure 3.20.

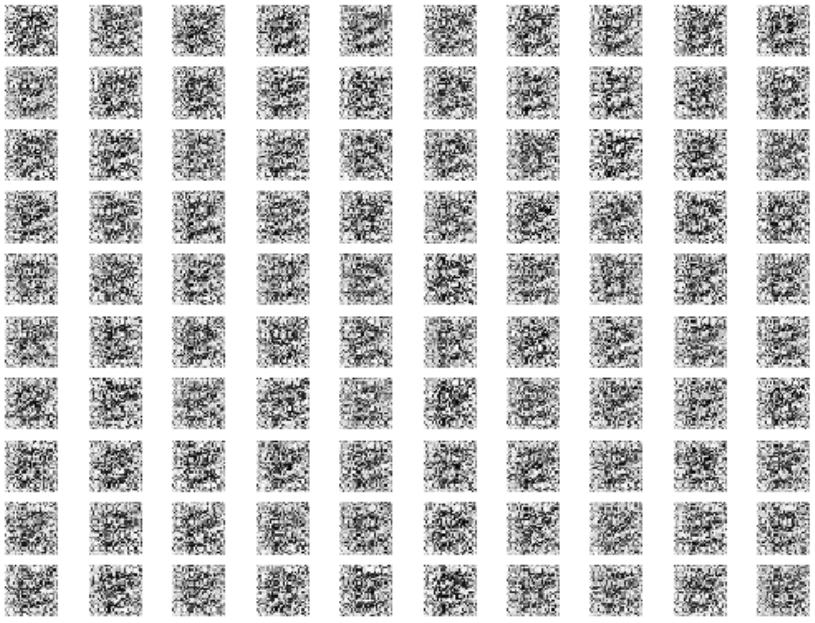


Figure 3.20: Generated images from epoch 450 by a GAN model suffering from convergence failure, resulting in noisy images. The experiment was conducted by Jason Brownlee in [41].

3.3.3 Vanishing gradients

The utilization of sigmoid-like activation functions (described in Section 3.1.2) in neural networks introduces a potential issue wherein the linear portion of a neuron produces excessively large output values after a certain number of epochs. Consequently, this situation leads to a gradient near zero within the sigmoid functions, resulting in very small weight adjustments.

3.4 Evaluation of GANs

In contrast to conventional deep learning models that undergo training with a loss function until convergence, the generator within a GAN relies on the feedback provided

by the discriminator for learning. Consequently, GANs lack an explicit objective function for evaluation [42]. Therefore, the quality of the synthetic images has to be evaluated in another way.

The simplest way to evaluate GANs is through manual evaluation, where you use your own eyes to visually compare the synthetic data created by the generator to the images in the training dataset. This method can be very time-consuming because of big datasets and a big amount of synthetic images and is not especially accurate compared to calculating and comparing the image data itself. Therefore a vast number of techniques are available to evaluate GANs [43].

Apart from manual evaluation, there are two additional principal categories: qualitative GAN generator evaluation, and quantitative GAN generator evaluation. Qualitative GAN generator evaluation typically involves subjective human assessment or comparison-based evaluation, which does not rely on numerical measurements. Various qualitative metrics, such as "Nearest Neighbors", "Rapid Scene Categorization", and the "evaluation of Mode Drop and Mode Collapse", are employed in this context. On the other hand, quantitative GAN generator evaluation involves the computation of numerical values or scores to quantify the quality of generated images. Examples of quantitative metrics include "Average Log-likelihood", "Inception Score (IS)", and "Fréchet Inception Distance (FID)". The latter two methods will be described in Section 3.4.1 and 3.4.2 respectively.

3.4.1 Inception score

The IS is a quantitative evaluation metric and is used to capture image quality and image diversity in images generated by GANs. The inception score was first proposed in 2016 by Tim Salimans, in the paper "Improved Techniques for Training GANs", and is widely used in the evaluation of GANs [44].

The worst possible IS score is 1, while the best possible score is the number of classes supported by the classification model. The classification model used is the pre-trained model inception-v3 [45]. In this case, the best score is 1000, as Imagenet, consisting of 1000 classes, is used to pre-train the inception-v3 model. The IS predicts the possibilities for a generated image to belong to each of these classes and these are then summarized, giving the inception score.

3.4.2 Fréchet inception distance

The FID is a quantitative evaluation metric and was first introduced in 2017 by Martin Heusel et al. in the paper "GANs Trained by a Two Time-Scale Update Rule Converge

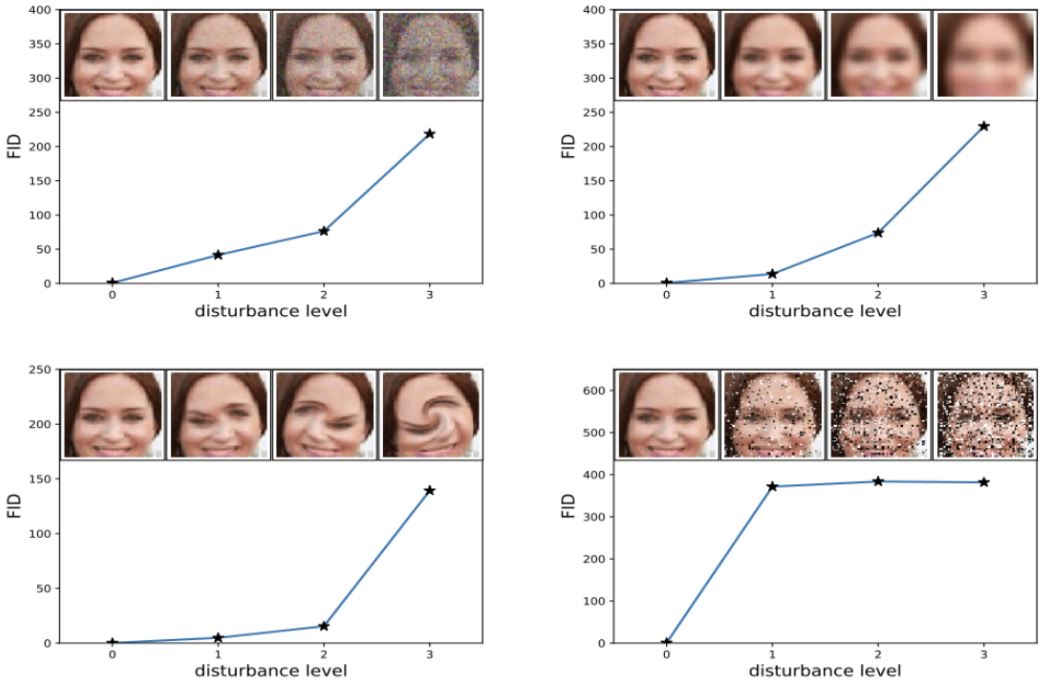


Figure 3.21: A visualization on how the FID score evaluates images. As shown the FID metric can detect white noise (upper left plot), salt and pepper noise (lower right plot), low-resolution images (upper right plot), and disturbed images (lower left plot). The figure is reprinted in unaltered form from [47].

to a Local Nash Equilibrium" [46]. It calculates the distance between feature vectors (or the distribution) for the generated images and the real images from the dataset. The score indicates how similar the feature vectors of a group of generated images are to a group of real images. Low values of the FID score indicate more similarity than a high FID score. Figure 3.21 provides examples of different FID values.

The FID uses the pre-trained Inception-v3 [45] image classifier to capture computer-vision-specific features of an input image [47]. The advantage of the FID score compared to the IS is that the FID takes into consideration the statistics of the real images. Because this thesis is dealing with medical images and it is important to create images that are similar to the original images, the FID score is chosen as the main evaluation method in this thesis.

3.5 Transformers

A transformer is a form of neural network architecture that is able to map one sequence to another by learning complex relationships within the data. Transformers were first introduced in the paper "Attention Is All You Need" in 2017 by Vaswani et al. [48] and have since become an increasingly prevalent architecture. Initially, transformers were

used in natural language processing (NLP) for tasks such as translation, outperforming older architectures such as recurrent neural networks (RNNs) [49] and long short-term memory (LSTM) [50]. This is due to their speed and efficiency, attributed to their parallelizability and their ability to process entire sequences at once, which additionally introduces less inductive bias. Recently, however, they have been applied to more complex tasks such as vision tasks [51] and speech recognition [52]. This section will provide first a high-level overview of the transformer's architecture and core concepts, followed by a more extensive review.

Transformers are entirely based on a self-attention mechanism. The transformer model excels at modeling dependencies between long sequences, as the attention mechanism allows for contextual information to be captured at any point within the sequence. Attention, of which self-attention is a sub-type, is a mechanism that allows the model to emphasize the information that is most relevant, by assigning weights to different parts of the sequential input. It is a way for machines to imitate cognitive attention in humans.

The transformer model, like other competitive neural sequence transduction models, is based on an encoder-decoder architecture [48]. The architecture is composed of multiple layers of identical encoders and decoders, represented in Figure 3.22 by the parameter N. The encoder and decoder can intrinsically be divided into two sub-layers, which are an attention layer and a feed-forward layer. These are made up of a multi-head self-attention mechanism followed by a position-wise feed-forward network. Both of these blocks are accompanied by a residual connection and a layer normalization. In the decoder, the attention layer includes an encoder-decoder attention block, which attends to the output of the encoder stack directly. Prior to input into the transformer, the sequence is reformatted using an embedding layer and positional encoding. The encoder processes the input sequence to create a set of feature maps, which are then passed to the decoder. The decoder attends to these feature maps and generates an output sequence element by element.

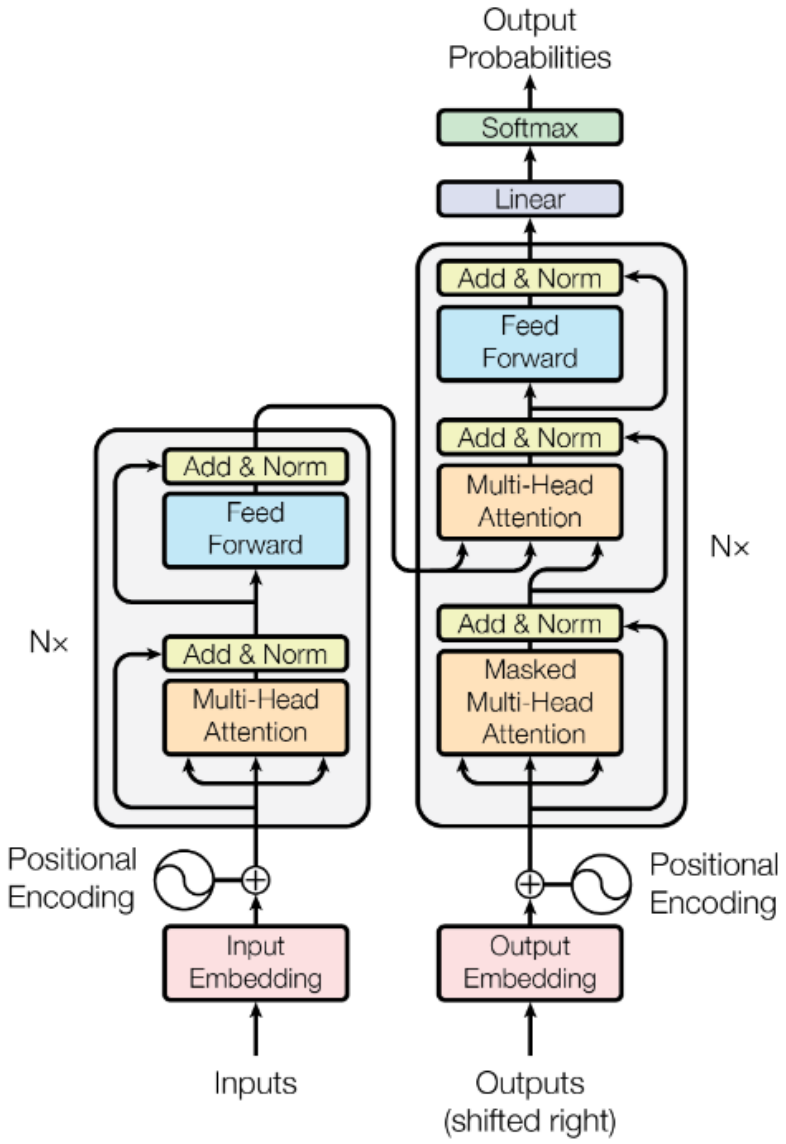


Figure 3.22: Transformer architecture. The figure is reprinted in unaltered form from the paper "Attention is all you need" [48].

3.5.1 Attention

In NLP, the concept of attention can be described as a way of finding the relevance between different words in a sentence. Every word of the input sentence is evaluated with regard to every other word in that same sentence. This is implemented as a weighted sum of all the input elements, where the weights are learned as specified by the similarity between each element pair seen in Equation (3.7). The Q, K, and V parameters will be further discussed in Section 3.5.2. If two words are highly related, i.e. the self-attention has indicated to the model that the input word is associated with another word, then the calculated attention score between these two words will be high. Conversely, two words

that hold no significance to each other, will return a negligible attention score. This score is then used to predict a target word, e.g. a translated word, or the subsequent word of a sentence in a text completion application, although the same principle works in different applications.

$$Attention(Q, K, V) = Softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{3.7}$$

When making predictions, the attention mechanism allows the transformer model to attend to different parts of another sequence whereas, in self-attention, the transformer model attends to different parts of the input sequence [53]. What this means essentially, is that the window of information is "infinitely" big, or at least restricted by hardware as opposed to the architecture, contrary to architectures such as RNNs, where if the input sequence is too long, the model can "forget" pieces of information.

3.5.2 Transformer architecture

Now that a high-level overview of the transformer has been provided, its structure will be reviewed in greater detail to grant a better understanding of how it functions. The majority of the information presented below is based on the research presented by Vaswani et al. [48] in their introduction of the transformer.

Input embedding

For the transformer to be able to read and understand the data, whether it is a sentence or an image, the input sequence has to in some way be embedded with its information in a readable form. The input embedding block maps discrete symbols, called input tokens, using learned embeddings, into continuous vectors in a high-dimensional space. In this way, the transformer is able to read the text as a sequence of vectors. The input can be a sentence to be used for translation tasks, or a patch of an image used for image classification to give some examples.

Positional encoding

Models like RNNs [49] and CNNs inherently store the position of each input token, however on account of the transformer architecture, it does not retain this innately. The order of the input sequence is consequential for each function the transformer is used for. Therefore, the input is passed through a positional encoding block where information

about the relative position is introduced through *sine* and *cosine* functions (3.8) and (3.9). The encoding of the input sequence involves assigning each token a unique position and applying the sine function to tokens with even positions and cosine to those with odd positions. The dimension of the positional encoding vector is defined as d_{model}, which is equal in size to the embedding vector.

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{model}}) \tag{3.8}$$

$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{model}}) \tag{3.9}$$

Multi-head attention

Before getting processed in the attention block within the transformer, the embedded and position-encoded input is separated into three equal matrices; Query (Q), Key (K), and Value (V). These matrices are then linearly projected into N heads. The multi-head attention which is then implemented merely consists of multiple attention mechanisms in parallel (Section 3.5.1). The procedure is illustrated in Figure 3.23.

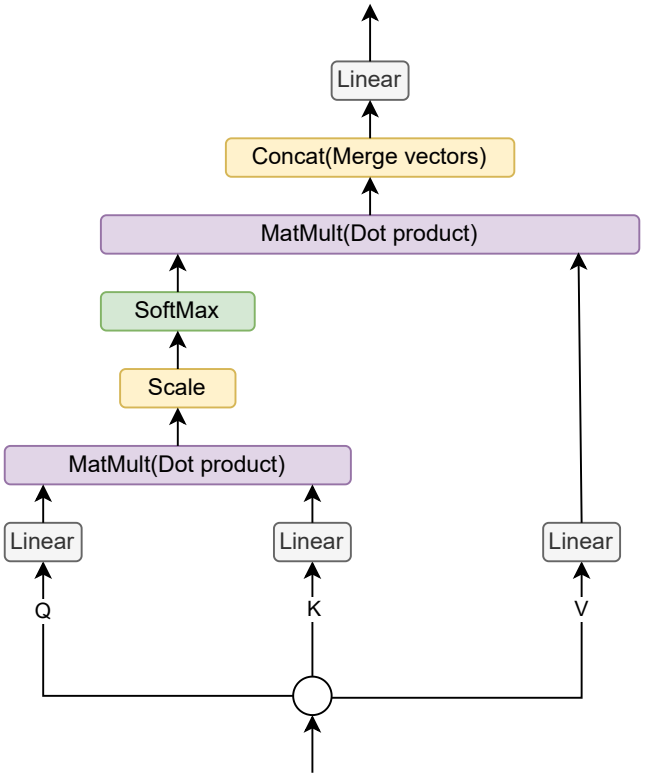


Figure 3.23: The multi-head attention module from Figure 3.22 in more detail. The MatMult (Dot product) block represents Equation (3.7).

Representing the currently processed input token is the query matrix. The key matrix is represented by the entire input sequence and is what the query token is referenced against. The value matrix represents the information that the model should attend to for each input token [54]. The dot product between the query and key matrices is calculated, producing a score for each position of the input token, which is run through a normalization and SoftMax (described in Section 3.1.2) layer to produce proper attention weights. These weights are then added to the values producing the context vector. The context vector produced from each attention head is then concatenated together into a combined matrix.

After the attention layer, there is a residual connection where the resulting matrix is concatenated, or joined together, with the previous input sequence followed by a layer normalization [55], forming the input to the next layer. Another notable part of the architecture is the feed-forward block [56]. It consists of two linear layers separated by a ReLU activation function (described in Section 3.1.2) and applies a point-wise transformation to each position of the input separately. The purpose is to introduce non-linearity and learn complex relationships within the transformer model. To preserve the information from preceding layers, the output is combined with a residual connection, similar to the attention layer.

Within the transformer architecture, there is a significant distinction to make between the multi-head attention blocks. In the encoding stage, the self-attention block enables the input sequence to attend to itself, capturing important relationships and dependencies within the sequence. On the other hand, within the decoder, there exist two instances of the multi-head attention blocks. In the first instance, the target sequence attends to itself, allowing it to understand its own context and refine its representations. In the second instance, referred to as the encoder-decoder attention block, the target sequence directs its attention towards the input sequence, integrating the information from the input during the decoding process [57].

3.5.3 Contrasting running and training phases

The transformer model operates in two distinct modes that reflect its state: training and running (inference). Before being employed for its intended purpose, the transformer must undergo a training phase where it learns to generate target sequences based on a given input sequence and the target sequence. In the encoder part of the architecture, the input is processed and encoded into a latent representation without any alterations between the training and running phases [58]. However, the flow of data in the decoder section differs.

During training, the decoder receives the target sequence prepended with a "start token". This augmented sequence, along with the encoder output, is processed to produce a probability distribution over possible output tokens [58]. The loss function is then applied to compare the generated output with the target sequence (training data), enabling the calculation of gradients for model training through back-propagation.

The training process utilizes a technique known as teacher forcing, wherein the decoder is provided with explicit guidance [58]. Instead of iteratively predicting the next token based on calculated probabilities using the previous token, which may introduce errors due to the accumulation of incorrect predictions, the decoder is given access to the target sequence. This approach mitigates potential errors and facilitates training. Within this context, the masked multi-head attention mechanism plays a crucial role, illustrated in Figure 3.24. While operating on the same principles as described earlier, it incorporates a triangular mask. This mask ensures that the decoder only has access to previous information during the prediction process. Even if an erroneous prediction occurs, the correct token is still available when predicting the subsequent token.

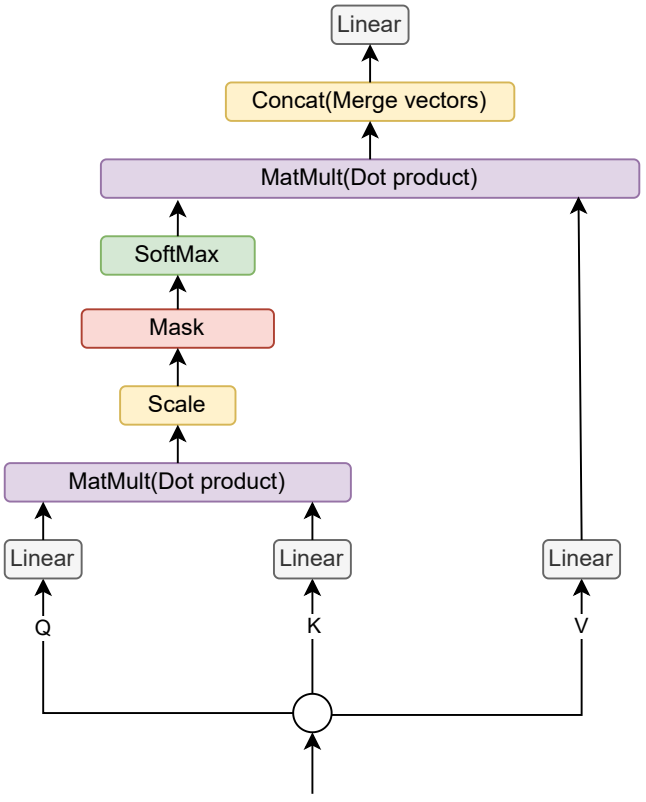


Figure 3.24: The masked multi-head attention module from Figure 3.22 in more detail.

In the inference phase, the decoder receives only the start token as input. Leveraging the learned weights, it encodes this input into a latent representation and derives probabilities for generating the output. The generated output is then appended to the start token and

fed back into the decoder for further processing. This iterative process continues until the model predicts an end-of-sentence token, indicating the completion of the output generation process [58].

3.6 Vision transformers

Vision tasks such as image classification [59, 60] and segmentation [61, 62], object detection [63, 64], and also GANs [3, 10] have historically used convolutional neural networks. Following the significant advancement brought about by transformers in various domains, researchers initiated investigations into the application of transformers in visual tasks, driven by their parallelizability and reduced inductive bias. In the paper titled "An image is worth 16x16 words" [51], the authors presented a classifier termed the vision transformer, specifically designed for image classification utilizing a transformer-based architecture.

The ViT model follows the same encoder architecture as the transformer in Section 3.5, as depicted to the right in Figure 3.25. In contrast to the processing mechanism of transformers in natural language processing, as expounded upon in Section 3.5, the ViT model adopts a similar strategy by representing non-overlapping patches of the image as vectors, akin to how transformers treat words as vectors. This approach is shown to the left in Figure 3.25. Furthermore, an additional learnable "classification token" is incorporated, which signifies the output of the transformer encoder.

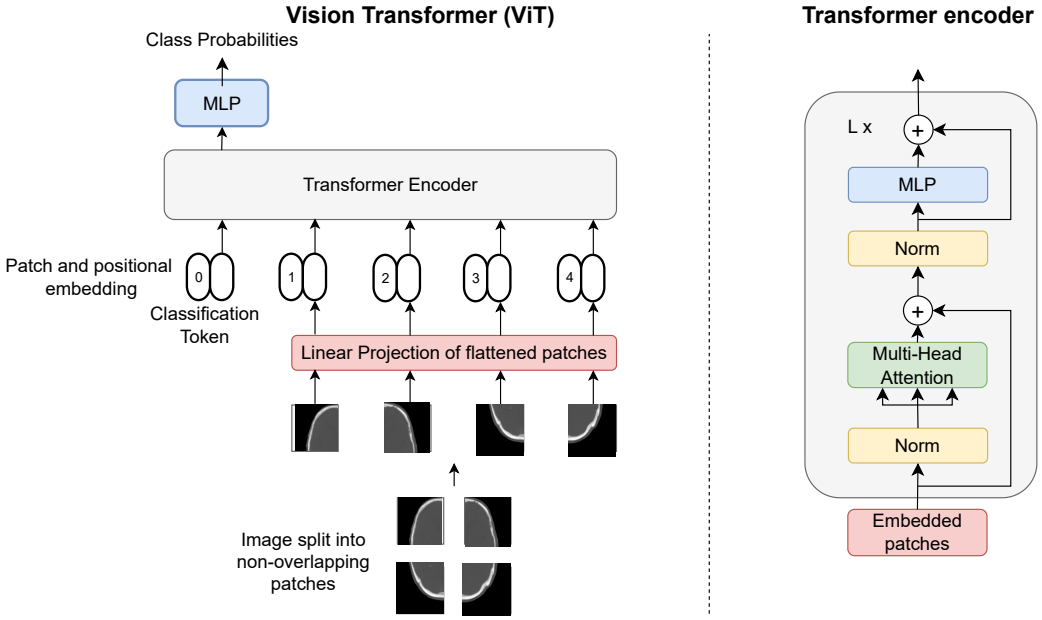


Figure 3.25: The architecture of the Vision transformer model from "An image is worth 16x16 words [51]". The overall architecture is shown on the left-hand side, and the transformer encoder architecture is shown on the right-hand side. The figure is adapted in altered form under the consensus of Dosovitskiy et al. [51].

3.7 ViTGAN

With the growing prevalence of vision transformers in various vision tasks, Lee et al. [65] embarked on the task of extending the ViT model to encompass image generation in addition to its existing capabilities. This endeavor aimed to investigate whether vision transformers could rival the performance of conventional CNN-based GAN architectures in the domain of image generation. The outcome of this exploration was the development of the first GAN model incorporating transformers.

However, the integration of ViT into the GAN framework posed several challenges. Training the GAN with ViT proved to be considerably more unstable compared to conventional approaches. Traditional regularization methods, which have proven effective in mitigating instability in CNN-based GANs, offered little improvement in this novel context. To address these unique challenges, two key concepts were devised:

- Enhanced regularization methods in the discriminator: This involved enforcing Lipschitz continuity, a measure of function stability and control, through the introduction of L2 attention, as presented in [66]. Additionally, an improved version of spectral normalization, a conventional regularization technique, was employed.

To tackle overfitting caused by the ViT discriminators exceeding their learning capacity, the discriminator incorporated overlapping image patches [65].

- Redesigned generator architecture: In the generator, the ViT model was inverted, and the roles of inputs and outputs were swapped. This allows the generator to generate image patches from latent embeddings. The generator comprised a mapping network for the latent z, a transformer encoder, and an output mapping layer.

By addressing these challenges and introducing novel modifications to the discriminator and generator, the ViTGAN model, illustrated in Figure 3.26, aimed to harness the potential of vision transformers for image generation tasks. The obtained results were satisfactory, although the performance of the ViTGAN model did not surpass that of the leading CNN-based GANs available during the study period. Additionally, the ViTGAN model exhibited constraints in terms of its limited resolution, confined to 64 x 64 pixels.

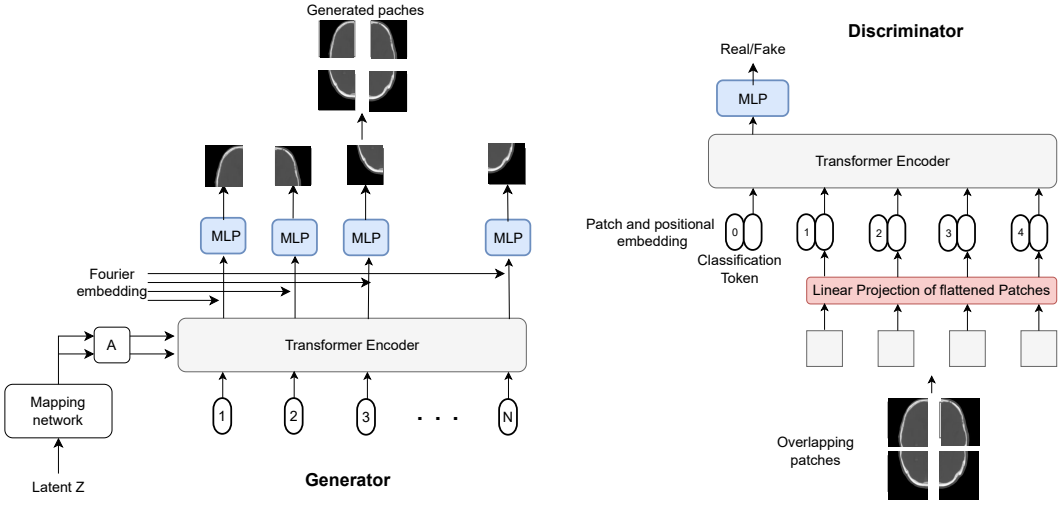


Figure 3.26: ViTGAN architecture. The figure is adapted from [65].

3.8 HiT-GAN

The utilization of vision transformers in GANs to generate high-resolution images poses several challenges, such as the scaling problem of more attention calculations as the resolution increase and the stability during the training phase. To address these issues, a model called HiT-GAN was proposed in the research paper titled "Improved Transformer for High-Resolution GANs" by Zhao et al. [4].

The HiT-GAN model incorporates a discriminator based on the StyleGAN architecture [67] "Progressive growing of GANs" from Section 3.2.5, and a generator that leverages

the ViT, as depicted in Figure 3.28. Given the quadratic scaling challenge posed by the Transformer's self-attention operation, as well as the greater need for spatial coherence in structure, color, and texture during image generation compared to discriminative tasks, the generator is partitioned into low-resolution stages and high-resolution stages. The low-resolution stage adopts the Nested Hierarchical Transformers design, introduced in [68]. The authors further enhance this design by introducing a method called "multi-axis blocked self-attention" that is more adept at capturing global information. This method will be expounded upon in Section 3.8.2 however, prior to its detailed exposition, fundamental techniques employed by the authors in the HiT-GAN approach will be elucidated.

3.8.1 HiT-GAN components

During the generation process, the input gets upsampled for each resolution stage, as depicted in Figure 3.30. The upsampling is done by a method called PixelShuffle. The PixelShuffle technique was first introduced in [69], serving as a convolutional neural network (CNN) operation applied in super-resolution models to achieve image upsampling. It employs sub-pixel convolutions, which possess a stride of 1/t, demonstrating efficiency. The procedure involves extracting feature maps within the low-resolution domain, subsequently utilizing an array of learned upscaling filters to upscale the feature maps to generate high-resolution output. According to the authors, this approach effectively reduces the computational complexity of the overall Super Resolution operation. This results in a rearrangement of the tensor shapes, transforming them from $(B, C * t^2, H, W)$ to $(B, C, H * t, W * t)$, as depicted in Figure 3.27. Where t represents the upscale factor, C is the number of channels, H is the height, and W is the width.

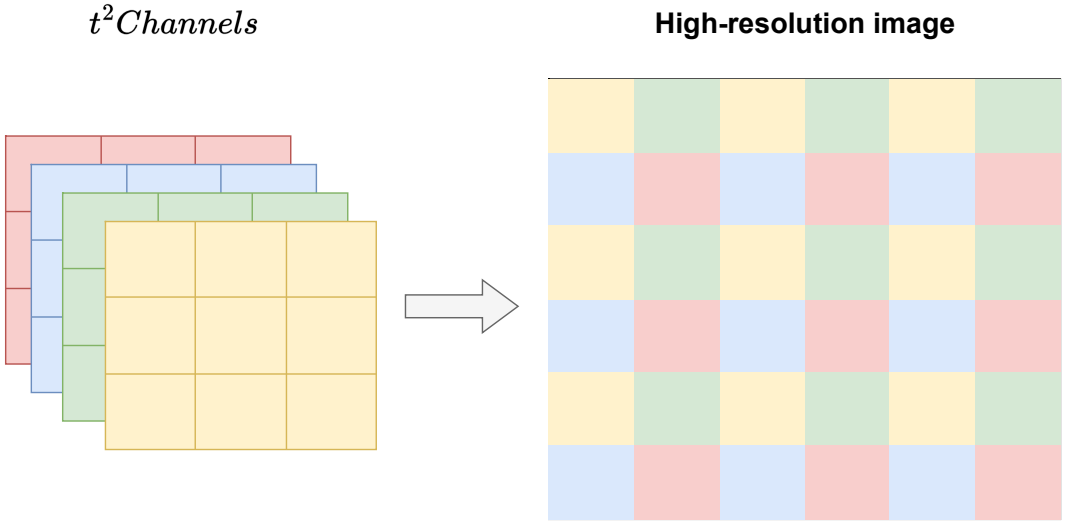


Figure 3.27: PixelShuffle upsampling illustrated using different colors for the channels. The figure is adapted from [70].

As outlined in Section 3.2.3, the training process of GAN models can exhibit instability. To deal with the problems of unstable training several different methods were utilized. First, the gradient penalty method used in the model is the R1 regularizer [71], shown in Equation (3.10). This approach ensures that the discriminator suffers a loss if it outputs a non-zero gradient orthogonal to the data manifold, when the generator generates images that the discriminator takes as a real image, giving 0 to the data manifold.

$$R_1(\psi) := \frac{\gamma}{2}\mathrm{E}_{p_{\mathcal{D}}(x)}\left[\|\nabla D_\psi(x)\|^2\right] \tag{3.10}$$

With the utilization of the R1 regularizer, the overall loss for the discriminator and generator is shown in Equation (3.11) and (3.12) respectively, where γ is the weight of the R1 gradient penalty.

$$\mathcal{L}_D = -\mathbb{E}_{x\sim P_x}[\log(D(x))] - \mathbb{E}_{z\sim P_z}[\log(1 - D(G(z)))] + \gamma \cdot \mathbb{E}_{x\sim P_x}\left[\|\nabla_x D(x)\|_2^2\right] \tag{3.11}$$

$$\mathcal{L}_G = -\mathbb{E}_{z\sim P_z}[\log(D(G(z)))] \tag{3.12}$$

Secondly, the HiT-GAN model integrates consistency regularization, as proposed in [72]. Consistency regulation uses the mean squared error in Equation (3.13) as a loss function. To enhance the effectiveness of this regularization technique, various augmentations such as color adjustments (random brightness, random saturation, and random contrast),

translation, and cutout are employed. The augmentation strategies, as described in [73], encompasses the application of specific transformations to the data, including color adjustments, translation, and cutout operations, thereby promoting robustness and generalization in the model training process.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2. \tag{3.13}$$

Thirdly, the utilization of non-saturating loss is employed to mitigate the adverse effects of the vanishing gradient problem, as expounded upon in Section 3.3.3. Instead of minimizing the log of the inverted discriminator probabilities for generated images, the non-saturating GAN loss maximizes the log of the discriminator probabilities for generated images, as shown in Equation (3.14). This results in the gradient information being better when the weights are updated, and leads to a more stable training [74].

$$generator : maximize(log(D(G(z)))) \tag{3.14}$$

Finally, an alternative to architectural components such as max pooling and strided-convolutions, which are used in DCGAN 3.2.4, the HiT-GAN model uses an approach called "blurpool" proposed in [75], for the downsampling in the discriminator. The "blurpool" introduces antialiasing by lowpass filtering, thus further increasing stability.

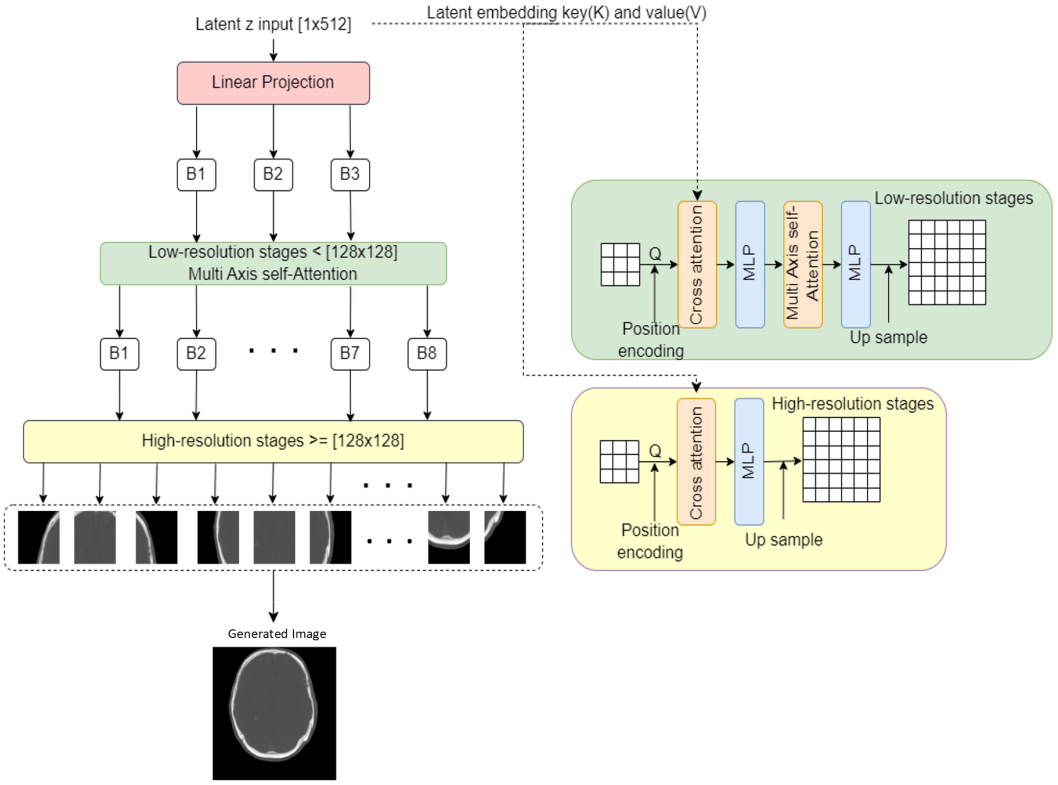


Figure 3.28: The HiT-GAN architecture showing the difference in the high-resolution and low-resolution stages [4]. The figure is adapted from [4].

3.8.2 Multi-axis blocked self-attention

The HiT-GAN model draws inspiration from the nested transformers framework [68], albeit with certain modifications. Notably, it introduces a novel approach that extends the concept of attention to multiple axes.

The multi-axis attention mechanism comprises two distinct forms of sparse self-attention: regional attention and dilated attention. Regional attention, inspired by previous works [76, 77], involves tokens attending to their non-overlapping neighboring blocks. On the other hand, dilated attention is employed to capture long-range dependencies across blocks, compensating for the absence of global attention.

Figure 3.29 illustrates the two types of attention, namely regional and dilated attention. These attention mechanisms are computed in parallel within a single layer, with each type being assigned half of the attention heads.

According to the authors, the adoption of a blocked structure exhibits a favorable inductive bias specifically suited for image-related tasks. To ensure balanced processing, each multi-axis blocked self-attention block operates on input sequences of similar lengths.

This balancing approach prevents a disproportionately sparse region by avoiding excessive attention focus from half of the attention heads.

Overall, the incorporation of regional and dilated attention in a parallel and blocked structure enhances the model's ability to capture local and long-range dependencies, resulting in improved performance for image generation tasks.

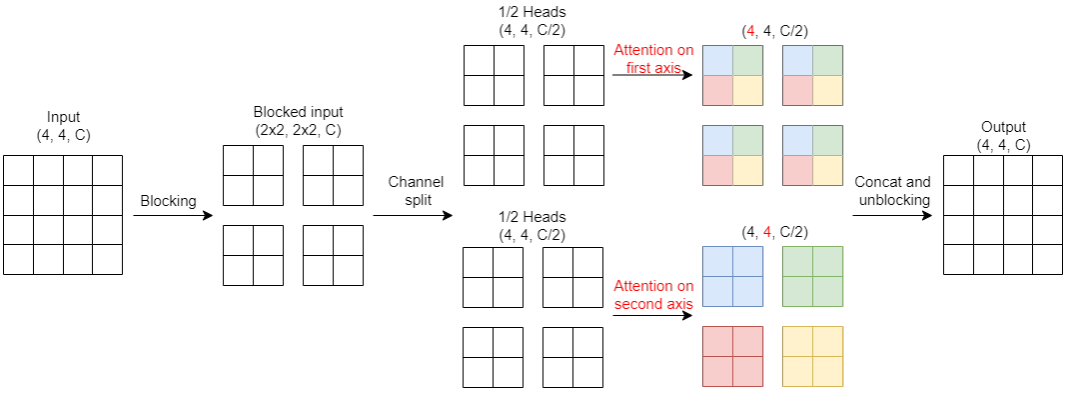


Figure 3.29: Multi-axis self-attention architecture from "Improved Transformer for High-Resolution GANs" [4]. [4,4,C] input with block size b=2. First, the input is blocked into 2x2 non-overlapping patches. Regional and dilated self-attention operations are computed for the patches along two axes. Each axis uses half of the attention heads. For each token, the attention operations are calculated in parallel, and their corresponding attention regions are illustrated with different colors. Then the tokens are turned into an output with the exact spatial dimensions as the input image. The figure is adapted with permission from Zhao et al. [4].

3.8.3 Cross-Attention for SelfModulation

The HiT-GAN model incorporates a mechanism called cross attention where the intermediate features directly attend to a small tensor derived from the input latent code. This process can be seen as a form of self-modulation [78], which plays a crucial role in stabilizing the generator and enhancing mode coverage. Moreover, in the absence of self-attention modules within high-resolution stages, directing attention towards the input latent code presents an alternative mechanism to capture global information during the generation of pixel-level details.

In contrast to ViTGAN [65], which relies on AdaIN and modulated layers and is limited to generating images up to a resolution of 64x64, the cross-attention operation employed in HiT-GAN demonstrates a linear computational complexity [4]. The inclusion of cross-attention facilitates effective information exchange between different stages and contributes to the generation of higher-resolution images beyond the limitations of ViTGAN.

3.8.4 Generator architecture

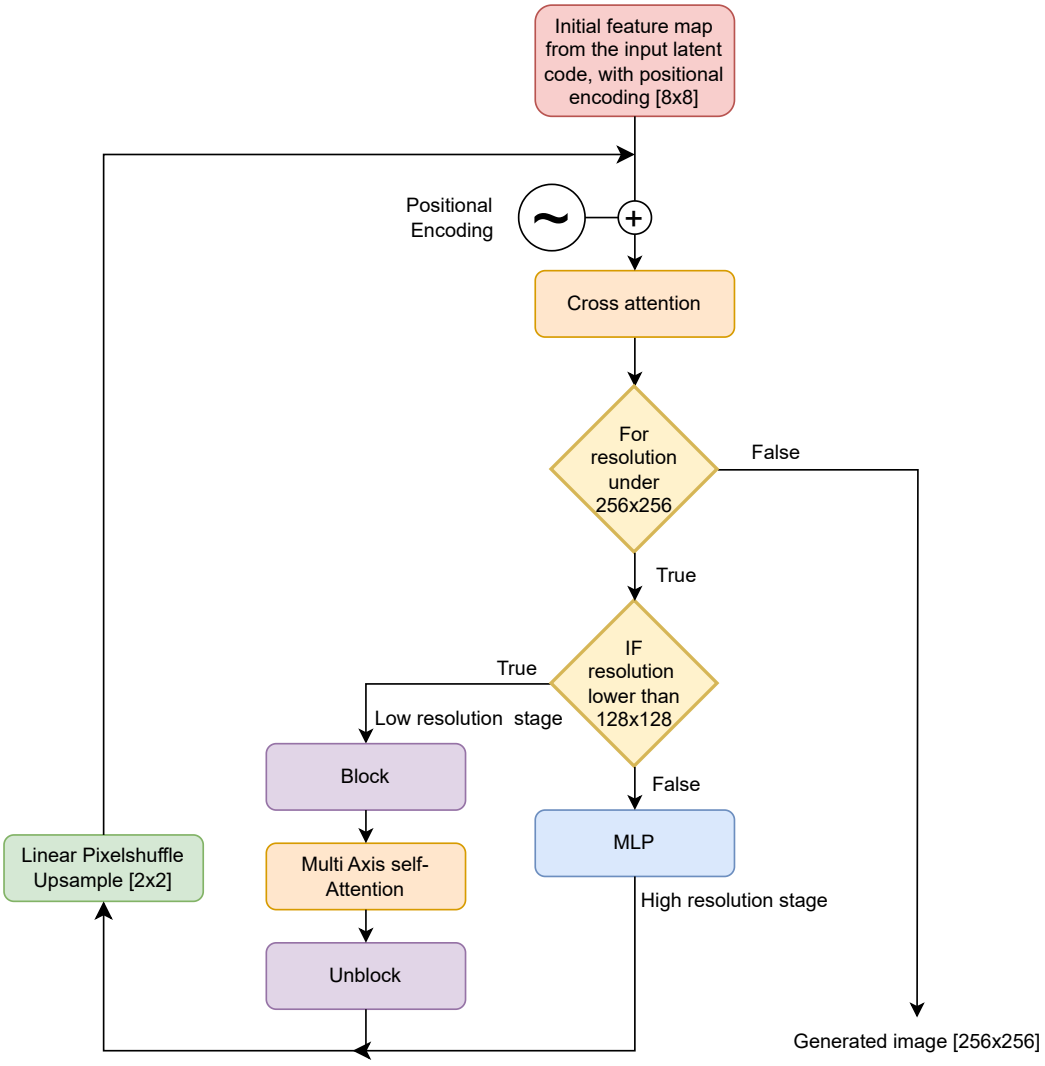


Figure 3.30: HiT-GAN generator architecture as a flow-chart.

The HiT-GAN model functions as an unconditional image generator (Section 3.2), indicating that it lacks control mechanisms over the generated output. It relies solely on a random latent variable z, with a dimension of 512, and learned weights to produce a random image of the target resolution 256 x 256 through a hierarchical structure, as shown in Figure 3.28.

The authors of the HiT-GAN model have employed TensorFlow's Keras Sequential to implement the model, incorporating GELU activation functions for the generator, leaky ReLU for the discriminator, and ReLU (all described in Section 3.1.2) for the MLP in the ViT.

During the image generation process, the model gradually increases the spatial dimensions of the future map while reducing the channel dimensions across multiple stages. To

Table 3.1: Resolution stages in the HiT-GAN model.

Stage nr.	Resolution	Resolution category
01	8x8	Low
02	16x16	Low
03	32x32	Low
04	64x64	Low
05	128x128	High
06	256x256	High

strike a balance between computational efficiency and feature dependence range during decoding, the image generation is divided into two stages: high-resolution (Section 3.8.6) and low-resolution (Section 3.8.5). Where the low-resolution stages include stages 1 to 4 and the high-resolution stages include stages 6 and 7 according to Table 3.1. This division allows for efficient processing while maintaining the ability to capture essential features. Both stages first undergo positional encoding and cross-attention and are upsampled using Pixelshuffle (3.27) for the next stage in the end. The generation process ends when the target resolution is reached, namely 256 x 256.

3.8.5 Low resolution stages

Within the low-resolution stages of the HiT-GAN model, an efficient attention mechanism enables the spatial mixing of information. The model's architectural design follows the decoder framework presented in the nested transformer approach [68].

Initially, the input feature is divided into non-overlapping patches, represented in the "block" block in Figure 3.30, with each patch representing a localized region within the input feature. This patch-based representation allows the model to capture local context and information. To incorporate positional information, the authors introduce a learnable position encoding mechanism. This encoding scheme enhances the model's understanding of spatial relationships and contextual dependencies among the patches, aiding in the effective representation of features.

Each patch subsequently undergoes independent processing through a shared attention module as shown in the "multi Axis self-attention" block in 3.30. This attention module, known as the multi-axis blocked self-attention, excels in capturing both local and global relations, thereby facilitating the creation of rich feature representations.

The final step in the low-resolution stage is shown in the "unblock" block in Figure 3.30, and refers to unblocking the image to the feature map.

3.8.6 High resolution stages

While the low-resolution stages of the HiT-GAN model primarily emphasize spatial dependency, the high-resolution stages prioritize the synthesis of pixel-level image details based on local features. In order to reduce computational complexity, all self-attention modules are eliminated within the high-resolution stages, and instead, MLPs with ReLU (described in Section 3.1.2) activation functions are employed.

In the absence of self-attention, only the cross-attention module is active in the high-resolution stage. This cross-attention mechanism facilitates direct conditioning of the network on the intermediate features and the initial input latent code. By doing so, it enhances the generative process, facilitates information flow between different stages, and contributes to the overall effectiveness of the model.

Chapter 4

Related work

In this chapter, we delve into the seminal research that serves as the foundation and inspiration for this thesis. Notable studies that have leveraged GANs and transformers in the context of medical image analysis are presented. Additionally, the integration of transformers in GANs for the synthesis of medical images is explored.

4.1 Introduction

Applying medical images to synthesis poses a few problems. First due to the detailed and complex nature of medical images, precision has to be in focus, more so than for regular images. Second, medical images are time-consuming to capture, and the process can be harmful to the patient, which is just some of the reasons why the datasets available are typically much smaller than the ones used in other vision tasks.

This research should give a perspective on what researchers have done in the field of medical images and the performance of existing models, and therefore give a good baseline for comparative analysis.

It should also be mentioned that the transformer-based models ViT, ViTGAN, and HiT-GAN from Sections 3.6, 3.7, and 3.8 are related works that built a foundation for this thesis, however, because of their technical relevance to this thesis's approach, were included in the technical background.

4.2 GANs for medical images

GANs have been employed for various purposes in medical applications: generating synthetic data, reconstruction, de-noising, detection, image translation, segmentation, and classification, are all examples of GANs usage on medical images [79]. GANs have demonstrated their utility in certain tasks, however, their effectiveness has been limited, as highlighted by the authors in the publication "GANs for Medical Image Analysis." *"While GANs show superior performance in many applications, they suffer from the same un-interpretability as other deep models. This is the main obstacle to their practical application in medical environments"* [79].

To gain a comprehensive understanding of the efficacy of GANs in the context of medical image applications, an examination of the research conducted in [80] is done. In this study, the authors evaluated synthesized images derived from three distinct datasets comprising medical images, using four distinct GAN models. The evaluation was performed based on the FID score (described in Section 3.4.2). The authors calculated the FID score for each GAN model individually, across all three datasets, using different hyperparameters in the evaluation.

The FID values calculated in [80] ranged from well above 150 to a bit under 100, as depicted in Table 4.1. The authors concluded with *"As a result, GANs effectiveness as a source of medical imaging data was found to be not always reliable, even if the produced images are nearly indistinguishable from real data."* Based on this result, this thesis will expand on the research by using vision transformers, known for having less inductive bias than the convolution-based models used in [80].

Table 4.1: Mean FID values obtained for different models on different datasets. The FID values acquired in the experiment, are approximated from Figure 3 in [80].

Dataset	DCGAN	LSGAN	WGAN	HingeGAN
ACDC	130	100	125	75
IDRID	175	185	100	100
SLiver07	80	75	145	80

4.2.1 DCGAN for generating synthetic CTP images

In Murat Korkmaz's master thesis from 2021, [3], DCGAN and MoCoGAN models were proposed and trained on the same preprocessed dataset, as described in 5.2. The goal of Korkmaz's work was to generate synthetic preprocessed CTP images in a 3D representation where the third dimension is images taken for one slice during one injection,

representing time. The way Korkmaz solved this was to include a separate discriminator to handle the slices in parallel to the discriminator that handles only one random image for each slice, thus capturing both the characteristics of the current slice and how the slice change during the injection.

In this thesis, the DCGAN model proposed by Korkmaz was adopted as a foundational framework due to its similarities with regard to dataset characteristics and the shared objective of generating CTP images. The DCGAN model was utilized in the approach in Chapter 6, to generate raw CTP images as described in Chapter 5.

4.3 Transformers on medical images

Due to the notable achievements of transformers in the domain of natural language processing, followed by their application in vision tasks, exemplified by the vision transformer model discussed in Section 3.6, researchers have recently displayed a growing interest in exploring the integration of transformers for medical image-related tasks [81].

Transformers have recently been used on segmentation [82–84] and classification [85–87] for medical images: The study conducted by Valanarasu et al. [84] employed transformer models to perform the segmentation of medical images across three distinct datasets, aiming to investigate the feasibility and efficacy of transformer-based approaches in the domain of medical image segmentation. The authors show that the transformer models achieve better performance than convolutional models. On the other hand, transformers were applied to the classification of stroke on CT scans in the model called StrokeViT [86] in combination with CNNs.

The benefits of including a transformer in these kinds of problems are to get rid of the locality of the convolution operation and exploit the long-range relationship in the data [87]. The downside is that to perform well, transformers require large-scale datasets, making the often small datasets in medical imaging not suitable for the task.

4.4 Combining transformers and GANs for synthetic medical images

Building upon the findings presented in the preceding sections, researchers have made attempts to combine Transformers and GANs to address the challenge of limited datasets in the field of medical imaging.

In the article "MedViTGAN: End-to-End Conditional GAN for Histopathology Image Augmentation with Vision Transformers", Li et al. [88] proposed the MedViTGAN model. This model combines GANs and transformers in an architecture based on TransGAN from [89], with the same transformer encoder architecture as in the Vision Transformer model from Section 3.6. The MedViTGAN model gave some promising results on histopathology images for data augmentation in an end-to-end manner. The FID scores achieved here were 57.8, 58.6, 118.4, and 126.2.

These results further motivate the primary objective of this thesis, which is to synthesize high-resolution CTP images using Vision Transformers.

Chapter 5

Dataset and preprosessing

This chapter provides a comprehensive description of the two datasets employed in this thesis: the raw dataset (see Section 5.1) and the preprocessed dataset (see Section 5.2). The preprocessing steps necessary to adapt the datasets for compatibility with the models are outlined in Section 5.3. Furthermore, various representations of the datasets are detailed in Sections 5.4, 5.5, and 5.6.

5.1 Raw CTP dataset

The dataset utilized in this thesis comprises a substantial collection of 70,050 CTP images, generously provided by the Stavanger University Hospital (SUH). These images have been obtained from a cohort of 157 individual patients between January 2014 and August 2020. Notably, the dataset encompasses a variable number of slices, ranging from 13 to 23, for each patient, accompanied by 30 distinct time points of image acquisition. How this is captured is illustrated in Figure 5.1. This extensive dataset facilitates comprehensive analysis and exploration of CTP data in the context of this research endeavor.

Within the dataset, the patients have been categorized into distinct groups according to the extent of the vessel occlusion. Specifically, there are 79 patients identified with large vessel occlusion (LVO), denoting a more severe condition, while 63 patients fall under the category of non-large vessel occlusion (non-LVO). Additionally, there are 15 patients who presented with stroke symptoms but were subsequently found to have no evidence of vessel occlusion following the diagnostic examination. This classification is demonstrated in Table 5.1 below.

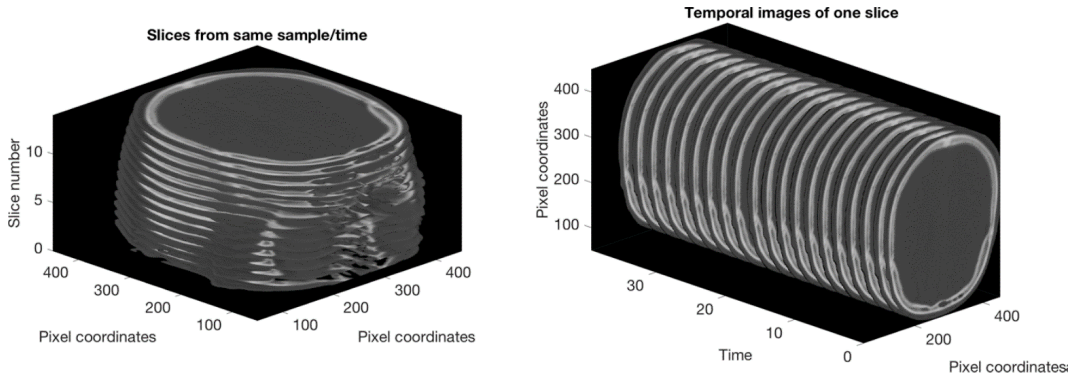


Figure 5.1: A representation of how each slice is captured in a 3D format, also denoting the fourth dimension on the right-hand side. Images are sourced from [90].

Table 5.1: Overview of patient groupings

Group code	Type	Num of patients
CTP_01	Patient with large vessel occlusion (LVO)	79
CTP_02	Patients with non-large vessel occlusion (non-LVO)	63
CTP_03	Patient with no vessel occlusion	15

All patients in this dataset went through an injection of 40 ml iodine-containing contrast agent (Omnipaque 350 mg/ml) and 40 ml isotonic saline in a cubital vein with a flow rate of 6 ml/s. This is to visualize the flow of blood in the brain. The delay for starting the scan acquisition was four seconds. Images were captured every 1s for the first 20s, and every 2s for the remaining 20s. The width and height of each image are 512×512 pixels with a resolution of 0.4258 mm/pixel, additionally, the slice thickness is 5mm. The same dataset was used in [2, 3, 61, 62, 91, 92].

5.2 Preprocessed CTP dataset

In addition to the collection of raw CTP data, a preprocessed CTP dataset was incorporated into the study. This dataset comprised 152 patients, encompassing a total of 67,530 CTP images. These images are structured identically to that of the raw dataset, however, prior to their inclusion, these images underwent a series of preprocessing steps, as outlined below:

1. Co-registration: The first time point in the 4D CTP dataset served as a reference for the co-registration process.

2. Encoding into Hounsfield Unit (HU) values: The CTP images were transformed into HU values, which are widely used in medical imaging for denoting radiodensity.

3. Brain extraction: An automated brain extraction method Najm et al. [93] was employed to isolate the brain region within the images.

4. Histogram equalization and gamma correction: Techniques such as histogram equalization and gamma correction were applied to enhance the image quality and improve visual contrast.

5. Standardization of the enhanced 4D tensor: The 4D tensor representing the enhanced CTP data was standardized to ensure consistent scaling and facilitate comparative analysis.

These preprocessing steps, originally introduced by Tomasetti et al., are discussed in greater detail in [91]. By implementing this preprocessed CTP dataset, the study aims to investigate the impact of these techniques on the subsequent analysis of data synthesis.

5.3 Preparation of data

In addition to the aforementioned preprocessing steps, a series of subsequent steps were performed on the raw CTP dataset to ensure its compatibility with the GAN models. Initially, the images were resized from their original resolution of 512 x 512 to a reduced size of 256 x 256. While it would be beneficial to explore the utilization of full-size images in future experiments to capture finer details in the data, this resizing step was necessary to ensure consistency and enable a fair comparative analysis between the models.

Additionally, the images in the dataset were initially in the TIFF file format. To align with the functionality of the implemented HiT-GAN model, which utilized a dataset function necessitating JPG format, the images were converted as such. It is important to note that the JPG format employs lossy compression methods, resulting in a loss of data during image compression, unlike the lossless methods used in the original TIFF format. This conversion may have introduced slight alterations to the base datasets, potentially impacting the generated images as well. Nevertheless, this conversion was necessary to incorporate the dataset into the architecture for further analysis and experimentation.

5.4 2D data

In this thesis, the CTP data is generated in the form of two-dimensional data referring to the x and y-axis of the image (Figure 5.2). This is referred to as a slice, and is the most conventional definition for images, however, when mentioning CTP data, there are several other dimensions to address as the 2D data does not capture the entire picture.

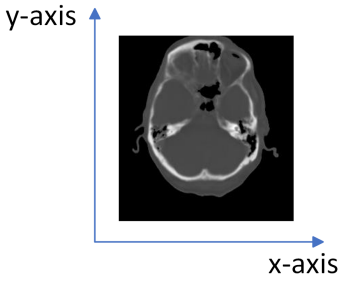


Figure 5.2: How the 2D data is represented in this thesis, using raw CTP image as an example.

5.5 3D data

When performing a CT scan, the idea is to capture a full 3D view of the patient. The way this is done is by capturing images as cross-sections or slices of the brain for different altitudes of the head, along the z-dimension. In this way, the brain can be illustrated as several slices ranging from the bottom of the head to the top (Figure 5.3).

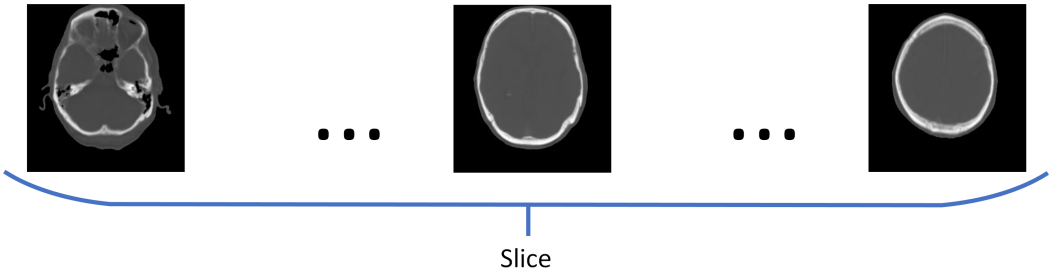


Figure 5.3: How the 3D data is represented in this thesis, using the raw CTP images as an example.

5.6 4D data

As explained in the medical background 2.2, CTP refers to the evaluation of a contrast agent's propagation through the bloodstream. It is therefore interesting to capture the time aspect of the perfusion, from the moment of injection until the wash-out of the contrast agent. This should be for each slice in the z-dimension. Combined, there now is a full 3D representation of the patient's brain for 30 time steps, completing the 4D representation (Figure 5.4).

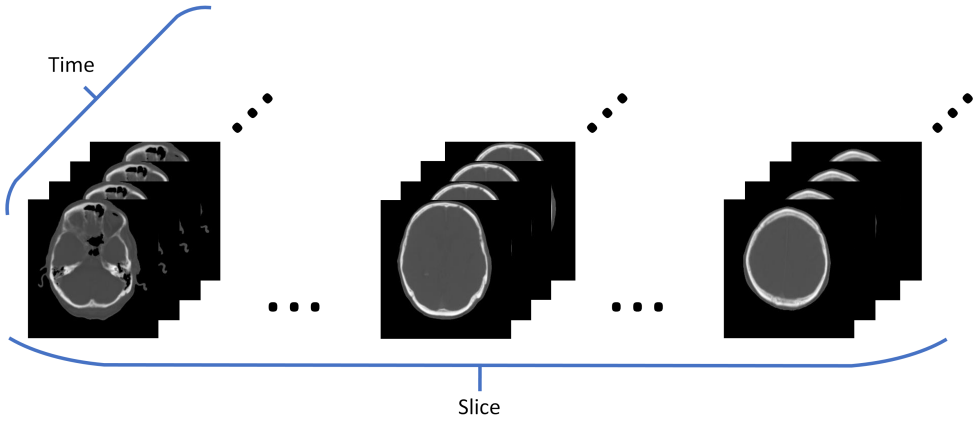


Figure 5.4: How the 4D data is represented in this thesis, using raw CTP images as an example.

Chapter 6

DCGAN approach

The following chapter details the experiments conducted in the context of this project, focusing on the DCGAN model [10]. Before specifying what the experiments themselves entail, the DCGAN model will be introduced shortly followed by what preparatory steps were taken prior to conducting the model training. Subsequently, the results obtained from each DCGAN experiment are presented and discussed, with a comprehensive analysis of the image quality, limitations, and potential areas for improvement. Finally, this chapter is concluded by summarizing the key findings and insights derived from the DCGAN experiments, setting the stage for the subsequent Chapter 7 which introduces the HiT-GAN model [4].

6.1 Introduction

As this thesis builds upon a previous master thesis by Murat Korkmaz in 2021 [3], it is only fitting that the first experiment utilizes the 3D DCGAN model presented in the said thesis. The motivation for why this model was implemented is threefold: i) to establish a knowledge basis, ii) to evaluate the baseline performance, and iii) to make a comparative analysis. This entails; firstly, the initial implementation of the DCGAN serves as a foundational step in understanding and gaining practical experience with the concept of GANs, their structures, limitations, and possibilities. Secondly, by implementing the DCGAN, a baseline performance can be established for generating CTP images and its limitations and potential can be assessed. Lastly, the results obtained from the DCGAN model will serve as a reference point for evaluating the advancements achieved by the HiT-GAN in terms of image quality and realism.

6.2 Experimental setup

While Korkmaz's thesis [3] primarily centered around preprocessed data, this thesis takes a different approach by primarily focusing on the utilization of raw data CTP images provided by SUS. The decision to prioritize raw data stems from its inclusion of supplementary information that adds valuable insights. Nonetheless, experiments involving preprocessed data have also been incorporated to widen the scope of comparison and analysis.

Since the architectural details of the DCGAN have been explained in Section 3.2.4, they will not be mentioned here, however, what is worth highlighting is the modifications made to the model to suit the specified research objectives. Upon reviewing the code comprising the model, certain improvements were deemed beneficial to be made. The original model would fetch an incomplete set of images from as many randomly chosen patients as the batch size specified, for each epoch. Another downside that followed from this logic, was that the model would likely not review all images for every epoch, only a fraction of them. In the modified version, the entire set of images the patient is composed of is fetched instead. Additionally, a list of patients was created, where every chosen patient was removed from said list, to emulate a random selection without replacement. This was to ensure every patient was visited for each epoch. The list would then be remade for every epoch.

By establishing this foundation, the experiments conducted can now be discussed, shedding light on the insights gained through this investigation.

6.3 Experiments

The model underwent over 20,000 training steps; however, for the purpose of these experiments, only the checkpoint at 20,000 steps will be referenced, as it represents the final saved weights. To remain consistent with the original code, the latent dimension was set at 180, and the discriminator and generator's learning rate at 0.0001. The model employs binary cross-entropy (BCE) loss, which was logged every 100 steps. The plotted losses illustrate the progression as a function of these 100-step intervals.

It is important to note that each step corresponds to the images from a single patient. Therefore, the number of steps required to complete one epoch is equivalent to the number of patients in the dataset. Consequently, the last checkpoint corresponds to a total of 127 completed epochs, indicating 127 complete iterations through the entire dataset.

6.3.1 Results on raw data

Figure 6.1 and 6.2 depicted below showcase the concurrent calculation of losses for the generator and discriminator during the training process, using the raw dataset. These losses and their optimal values are discussed in detail in Section 3.3. The plots reveal an initially erratic training phase, gradually converging towards a stable value with intermittent fluctuations leading to occasional spikes, which is as expected. Notably, the generator consistently maintains a higher value than what is ideal, while the discriminator exhibits overperformance. This discrepancy may stem from the discriminator becoming overtrained, disrupting the balance between the two convolutional networks and potentially contributing to suboptimal results. Of utmost significance, however, is the convergence of both networks, constituting a crucial aspect of the training process.

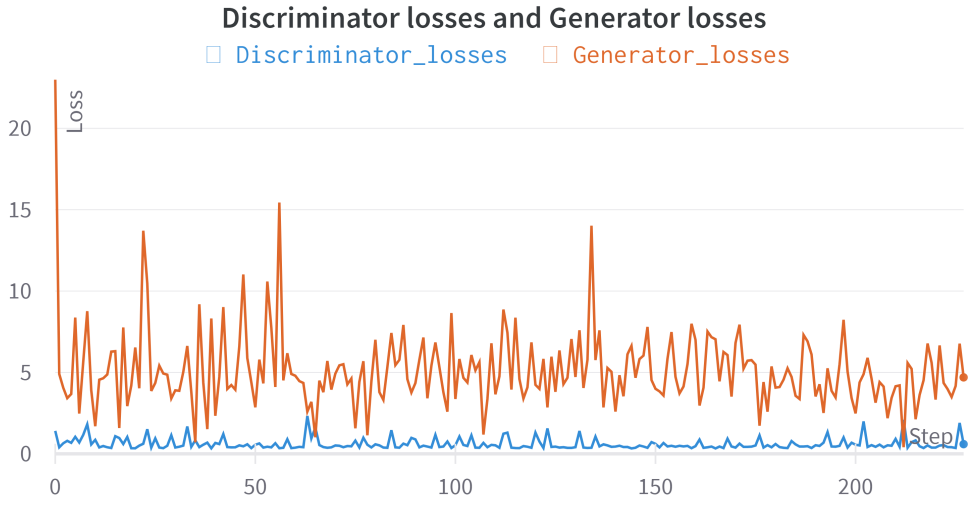


Figure 6.1: Generator and discriminator loss for the modified DCGAN.

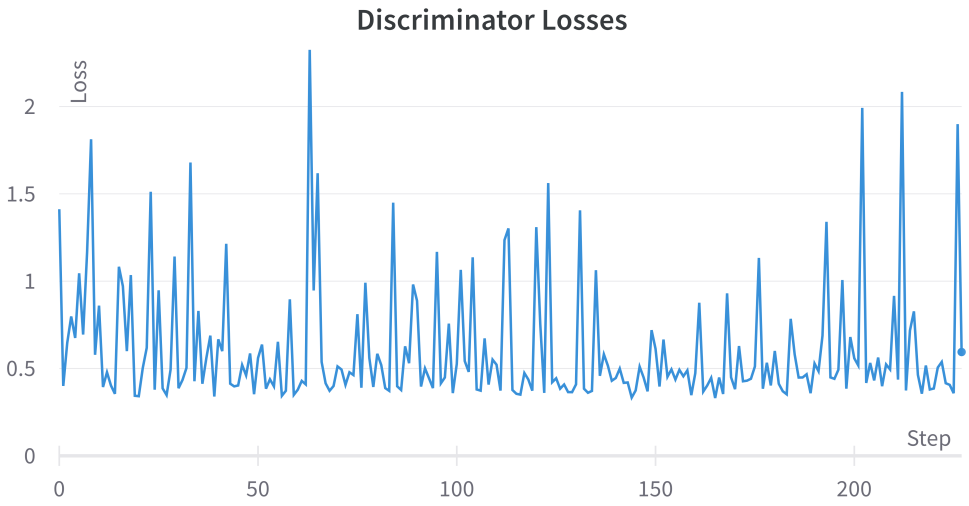


Figure 6.2: Magnified view of the discriminator loss for the modified DCGAN.

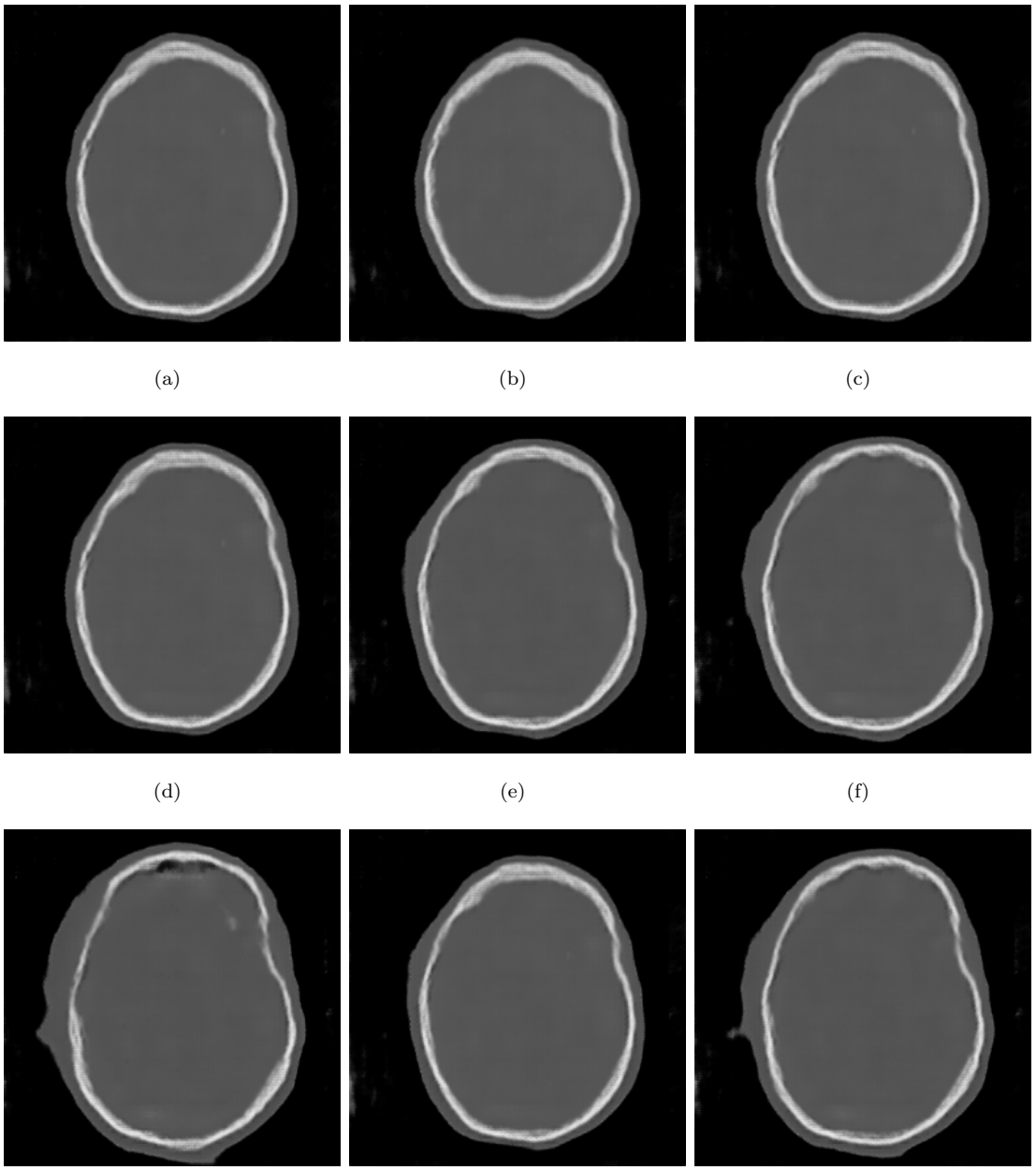
In order to establish a comparative framework for both HiT-GAN and DCGAN models, the FID was calculated for the DCGAN model as well. For this analysis, 5000 generated images from the DCGAN model were compared against 5000 randomly selected real images. It is important to note that the dataset was not partitioned into separate training and evaluation sets, thereby the model should yield images that more closely resemble those used for evaluating the model, this is discussed further in Section 6.3.3.

To ensure a more comprehensive and unbiased FID score, as the metric takes into account the arrangement of the images, a permutation-based evaluation approach was adopted. Specifically, the FID and IS metrics were computed 100 times, each time with a different ordering of the images. The resulting FID scores from these 100 calculations were then averaged to derive the final FID score. This evaluation methodology allows for a robust assessment of the model's performance, taking into account the potential influence of image ordering on the FID metric.

The mean scores of the FID and IS metrics were calculated to be 168.157, and 1.146, respectively. The best-recorded values from among the 100 were calculated to be 143.039 for the FID score, and 1.146 for the Inception score.

Figure 6.3 showcases the raw CTP data generated by the DCGAN model. A random selection of these images reveals visually appealing results in terms of gray-scale values, shape, and overall content. However, a noticeable limitation is the absence of discernible brain structures within the skull, indicating a failure to extract meaningful information from within the skull. Moreover, the lack of diversity among the generated images is evident, suggesting the presence of mode collapse. This could stem from the imbalance seen during the training phase, as highlighted by the aforementioned loss functions.

Despite these shortcomings, from a visual point of view, the generated data exhibits a reasonable resemblance to the training data.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 6.3: Generated raw CTP data from the DCGAN.

6.3.2 Result on preprocessed data

The second experiment conducted using the preprocessed dataset followed an identical parameter configuration and setup as the previous experiment. The average values of the FID and IS metrics were computed as 222.837 and 1.501, respectively. Among the

100 generated samples, the lowest recorded FID score was 200.239, indicating a slight improvement regarding the similarity between the generated and real data distributions, while the Inception score remained consistent at 1.501. These results provide insights into the quality of the generated samples, demonstrating the performance of the model.

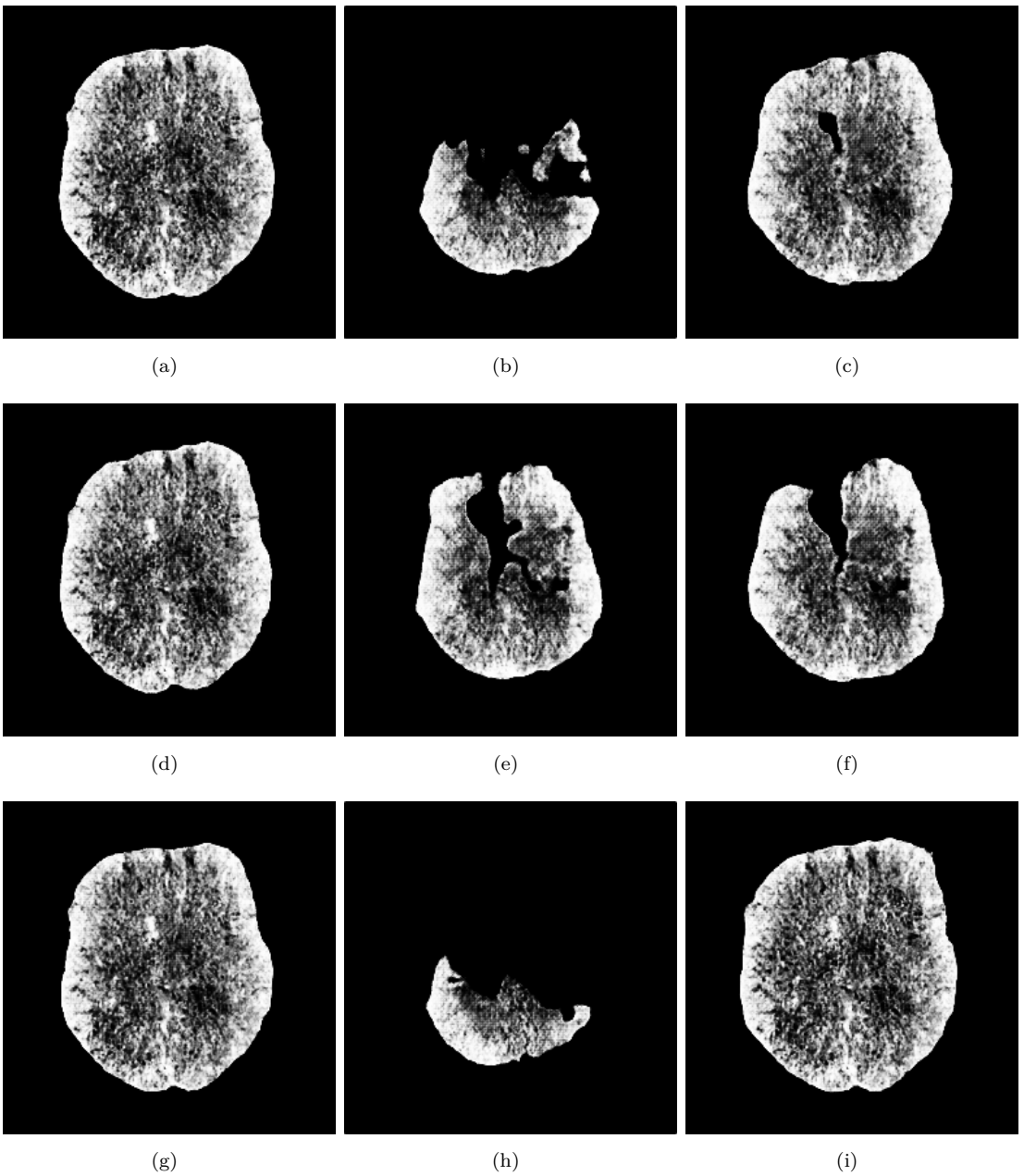


Figure 6.4: Generated preprocessed CTP data from the DCGAN.

Figure 6.4 showcases the generated preprocessed CTP data, which shares similarities with the observations made for the generated raw data. The increased contrast in these images reveals greater diversity in the brain patterns, yet there remains a noticeable lack of variability. This persistence of limited diversity echoes the discernible indications of mode collapse observed in the previous experiment. Additionally, as the model attempts

to capture the internal structures in the images, some information seems to be lost, as some images illustrate that parts of the brain are missing.

6.3.3 Discussion

For these experiments, the datasets were employed in their entirety to train the model with the anticipation that a sufficient amount of data would be available to facilitate the learning of a robust mapping from inputs to outputs. The decision to primarily focus on the training aspect of the experiment, rather than splitting the dataset for training and evaluation, was made strategically in order to allocate resources efficiently and optimize the research process. By directing attention toward training, the aim was to thoroughly analyze the performance and capabilities of the model under controlled conditions. This approach allowed for a deeper understanding of the training dynamics, fine-tuning of the model's parameters, and assessment of its potential for generating desired outputs. Although dataset splitting for evaluation purposes could provide additional insights, prioritizing the training phase enabled a more comprehensive and in-depth examination of the model's learning process.

Because of this decision, the FID measure is then expected to exhibit a bias toward the data generated by the DCGAN. Although this unconventional decision would not be replicated in future experiments, it is worth noting that, as the results will demonstrate, it did not significantly affect the comparison between the two models. It is also worth noting that the Inception-v3 model, which is employed for calculating the FID score, is pre-trained on the ImageNet dataset, which differs significantly from the dataset used in this thesis. Given this disparity, the achieved results can be considered noteworthy.

Despite its notable strengths, it is imperative to acknowledge the inherent limitations of the DCGAN model utilized in this study. One significant limitation pertains to its inability to generate meaningful 3D data, despite being specifically designed for this purpose. This limitation stems from the model's inherent definition of 3D data, where the third dimension represents time. Consequently, the generated data corresponds to a single random slice from a patient at different time steps, rather than comprehensive 3D representations. However, as the main focus of this thesis is to generate 2D data, this model serves as fine groundwork for these purposes.

Another noteworthy deficiency is the lack of diversity observed in the generated data, even when different latent vectors are provided as input. The generated data fails to capture the same diversity as the real-world data distribution, indicating a persistent mode collapse (Sec. 3.3.1). This phenomenon suggests that the generator consistently produces only from a small set of realistic samples that effectively deceive the discriminator.

In light of its limitations, the data generated by the DCGAN model serves as a valuable baseline for comparing the performance of subsequent models. As the model is already configured to fit the same task that the HiT-GAN is implemented for, it seamlessly handles the dataset employed in the subsequent experiments, thereby facilitating the experimental workflow. It enables comparisons not only in terms of image quality and evaluation metric values but also in terms of the proposed model's ability to generate more diverse data, addressing the shortcomings observed in the DCGAN outputs.

Chapter 7

Proposed approach

The forthcoming chapter aims to expound our approach, specifically centered on the utilization of vision transformers in GANs. The preparatory steps of the two experiments will be clarified before they are conducted in order to evaluate the efficacy and robustness of the HiT-GAN model. Furthermore, we will showcase generated images at various checkpoints including evaluation metrics for the images, and provide results that illustrate the overall progress of the training process, subsequently engaging in a comprehensive discussion of the obtained results.

7.1 Introduction

The primary aim of this study is to employ vision transformers in the generative process of GANs, with the objective of synthesizing artificial CTP images in 2D (5.4). The utilization of vision transformers is expected to minimize inductive bias, resulting in the generation of more realistic images [51, 65]. Furthermore, the synthetic images are intended to exhibit a level of detail comparable to that of the original CTP images. Consequently, an endeavor to preprocess the images will be undertaken to enhance their fidelity.

In order to accomplish this, a suitable model was required. Various models, including ViTGAN[1], TransGAN[2], and MedViTGAN, were evaluated. However, due to its remarkable ability to generate high-resolution images, the HiT-GAN model [4] described in Section 3.8 was ultimately chosen as the most appropriate candidate. It is worth noting that the HiT-GAN model, as described in the original paper, was trained on the CIFAR, ImageNet, and CelebHQ datasets, none of which specifically pertain to medical images.

[1] https://github.com/mlpc-ucsd/ViTGAN
[2] https://github.com/VITA-Group/TransGAN

The approach is divided into two separate experiments: The first experiment involves utilizing the raw CTP dataset, as detailed in Section 5.1, to train the HiT-GAN model. The second experiment involves employing the preprocessed dataset, as detailed in Section 5.2, for training the HiT-GAN model. However, prior to delving into the specifics of the experiment, a comprehensive description of the HiT-GAN model employed in the experiments will be provided.

7.2 Preparing the model

In order to align our dataset (described in Chapter 5) with the HiT-GAN model, several modifications were made to the original code[3]. The existing dataset builders were substituted with a function that facilitates the categorization of files within a specified directory into respective classes based on their sub-directories. Notably, the assignment of class values follows a sequential pattern, whereby the first sub-directory is assigned a class value of 0, the second sub-directory a class value of 1, and so forth. Additionally, image normalization procedures were applied. The training loop underwent a redesign to ensure that the dataset was iterated through a predetermined number of epochs, loading the dataset into memory before the training process starts. An additional operational mode, referred to as the "generate images" mode, was introduced alongside the existing train and evaluation modes. This newly incorporated mode facilitates the generation of a specified number of images using the saved generator model at a designated checkpoint.

The hyperparameters employed in this approach were those suggested by the creators of the HiT-GAN model [94], for execution on GPUs, despite the fact that the original code was executed on multiple TPUs. Wasserstein loss [95] and hinge gradient penalty [96] were options that could be employed in the HiT-GAN model. However, experimental results demonstrated that these methods exhibited inferior performance compared to the R1 and non-saturating loss functions (as detailed in Section 3.8.1) for this thesis's objectives.

On account of the low diversity between the images in our datasets, as well as the fact that it is pre-trained on Imagenet, it can be argued that the inception score metric is unsuitable for evaluating the images generated in this thesis. Despite this, the inception score is included in the experiments. The code used to calculate the IS in this thesis is taken from [97].

As a result of this, the FID score was chosen as the main evaluation metric in this thesis. That being said, also the FID score suffers from the disadvantage of utilizing Inception-v3

[3]https://github.com/google-research/hit-gan

which is pre-trained on Imagenet [98], making the FID score in this thesis less than ideal. Additionally, a noteworthy aspect affecting the evaluation of FID is the nature of the raw CTP images, which are gray-scale and exhibit substantial contrasts. For instance, the pixel values in the black background register as zero, while the tissue surrounding the head exhibits pixel values around 75, and certain regions of the skull can reach values of 225. Consequently, these significant gradients exert a discernible influence on the FID score.

7.3 Experiment 1

In the initial experiment, the raw CTP images were utilized as the data source. The dataset was partitioned into two distinct subsets: a training split encompassing 64,200 CTP images derived from 143 patients, and an independent evaluation split comprising 5,850 CTP images originating from 14 distinct patients. This partitioning strategy was employed to ensure that the evaluation process does not rely on the same data that was used for training our model. By segregating the dataset in this manner, the evaluation phase can be conducted using previously unseen data, thus providing a more robust assessment of the model's performance.

7.3.1 Results and discussion

The model underwent a total of 60 epochs of training on a single "Tesla v100-PCIe" GPU, with a batch size of 2, owing to the limitations imposed by the available 32510MiB of the GPUs. The latent dimension was set to 512, while both the discriminator and generator employed a learning rate of 0.00005. To enforce consistency and enhance training stability, consistency regularization techniques, and the R1 gradient penalty (detailed in Section 3.8.1) were incorporated into the training process.

During the training process, the loss and FID values are plotted based on the number of steps taken. Considering the batch size of 2 as utilized in our approach, each step involves the presentation of two real images to the model. Consequently, the total number of steps required to complete a single epoch can be calculated as the dataset size divided by the batch size, resulting in 32100 steps.

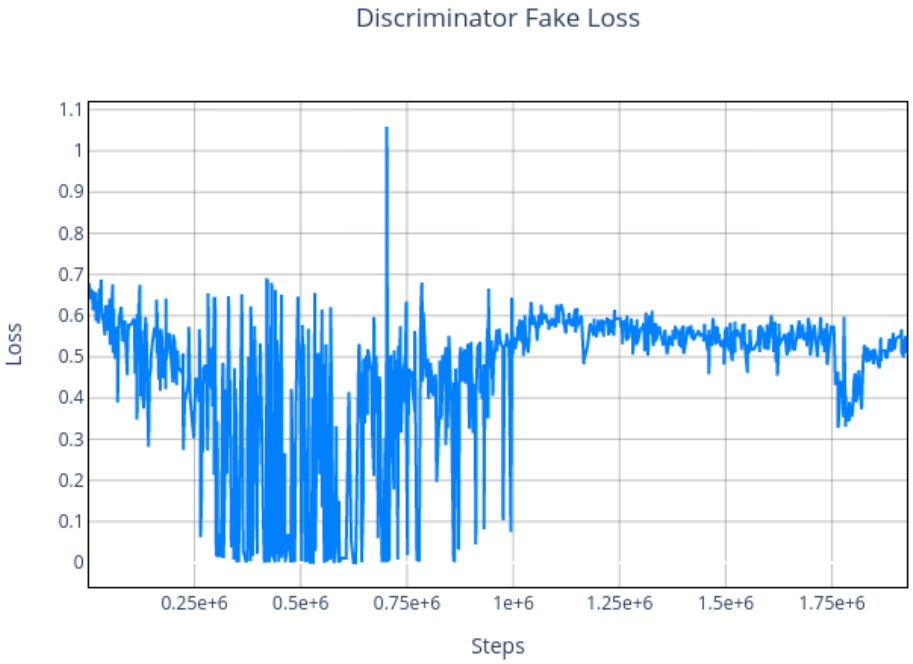


Figure 7.1: The "discriminator fake loss", as expounded upon in Section 3.3, was evaluated over a span of 60 epochs during the training of the HiT-GAN model.

The graphical representation of the discriminator's fake loss, as depicted in Figure 7.1, demonstrates initial instability during the early stages of training. However, notable stabilization occurs approximately after step 1e+6 or around epoch 31, with the fake loss converging consistently between the range of 0.5 and 0.6. These values align favorably with the optimal performance benchmarks established for the discriminator, as discussed in Section 3.4. This stabilization in the discriminator's fake loss serves as evidence of the model's enhanced capacity to discriminate between real and generated samples, affirming the robustness of the discriminator's discriminatory ability as the training progresses.

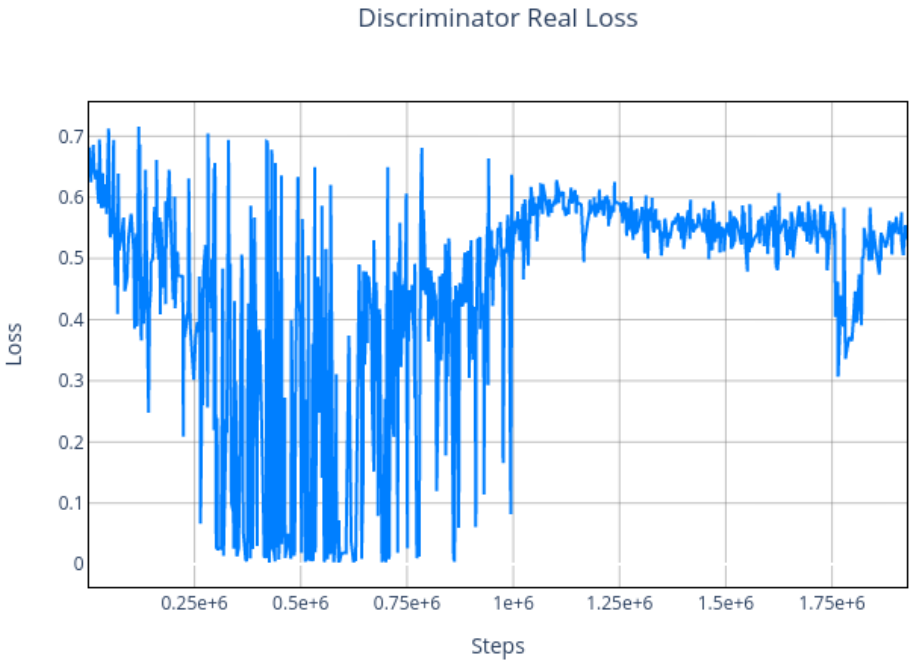


Figure 7.2: The "discriminator real loss", as expounded upon in Section 3.3, was evaluated over a span of 60 epochs during the training of the HiT-GAN model.

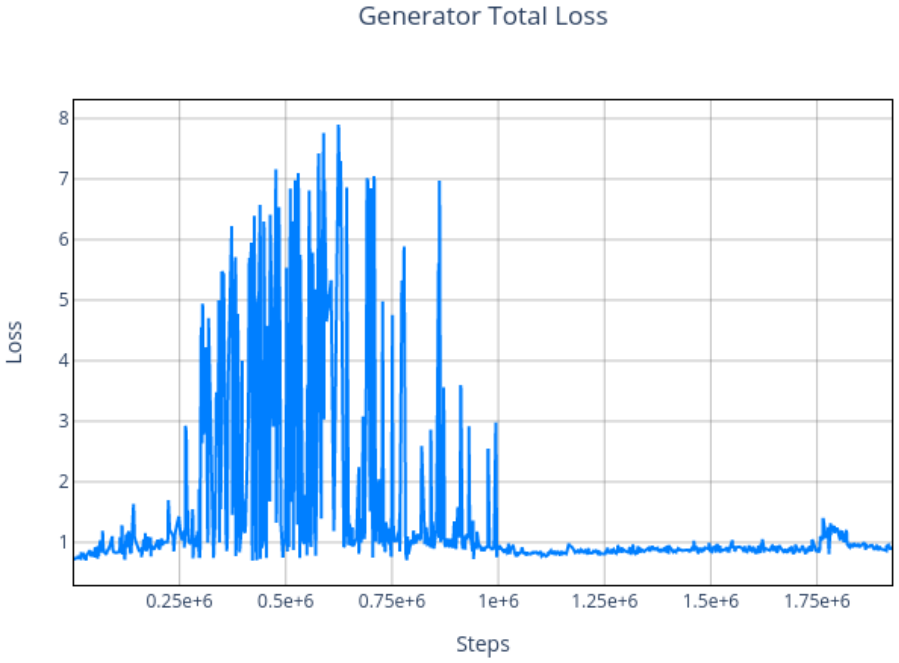


Figure 7.3: The "generator loss", as expounded upon in Section 3.3, was evaluated over a span of 60 epochs during the training of the HiT-GAN model.

The generator loss, depicted in Figure 7.3, demonstrates a stabilization trend around the value of one after 1e+6 steps. As elaborated in Section 3.3, this particular value signifies a desirable stabilization point, taking into account the corresponding discriminator loss values illustrated in Figure 7.1 and 7.2. This convergence point establishes a satisfactory equilibrium between the generator and discriminator components, ensuring a balanced training process. Studying the loss data, there is nothing to suggest any complications have arisen during the training, however, consulting the resulting image data will give a more conclusive verification.

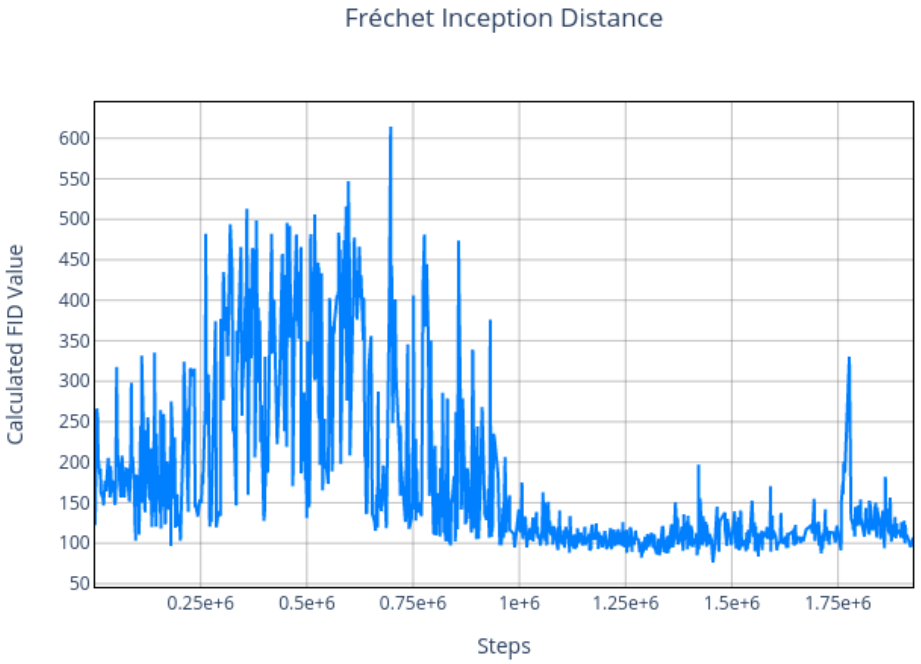


Figure 7.4: The FID score as expounded upon on in Section 3.4.2 calculated for every checkpoint during the training of the HiT-GAN training.

Concurrently with the training process, an evaluation model is employed to compute the FID score for each saved checkpoint. To accommodate GPU limitations, this evaluation model is executed on a separate GPU. The FID scores are visualized in the plot presented in Figure 7.4, where it can be observed that the FID score stabilizes around 100 after step 1e+6. It is noteworthy that the Inception-v3 model, employed for calculating the FID score, is pre-trained on the ImageNet dataset, which is distinctly different from the dataset utilized in this thesis. Considering this dissimilarity, the achieved results can be considered commendable.

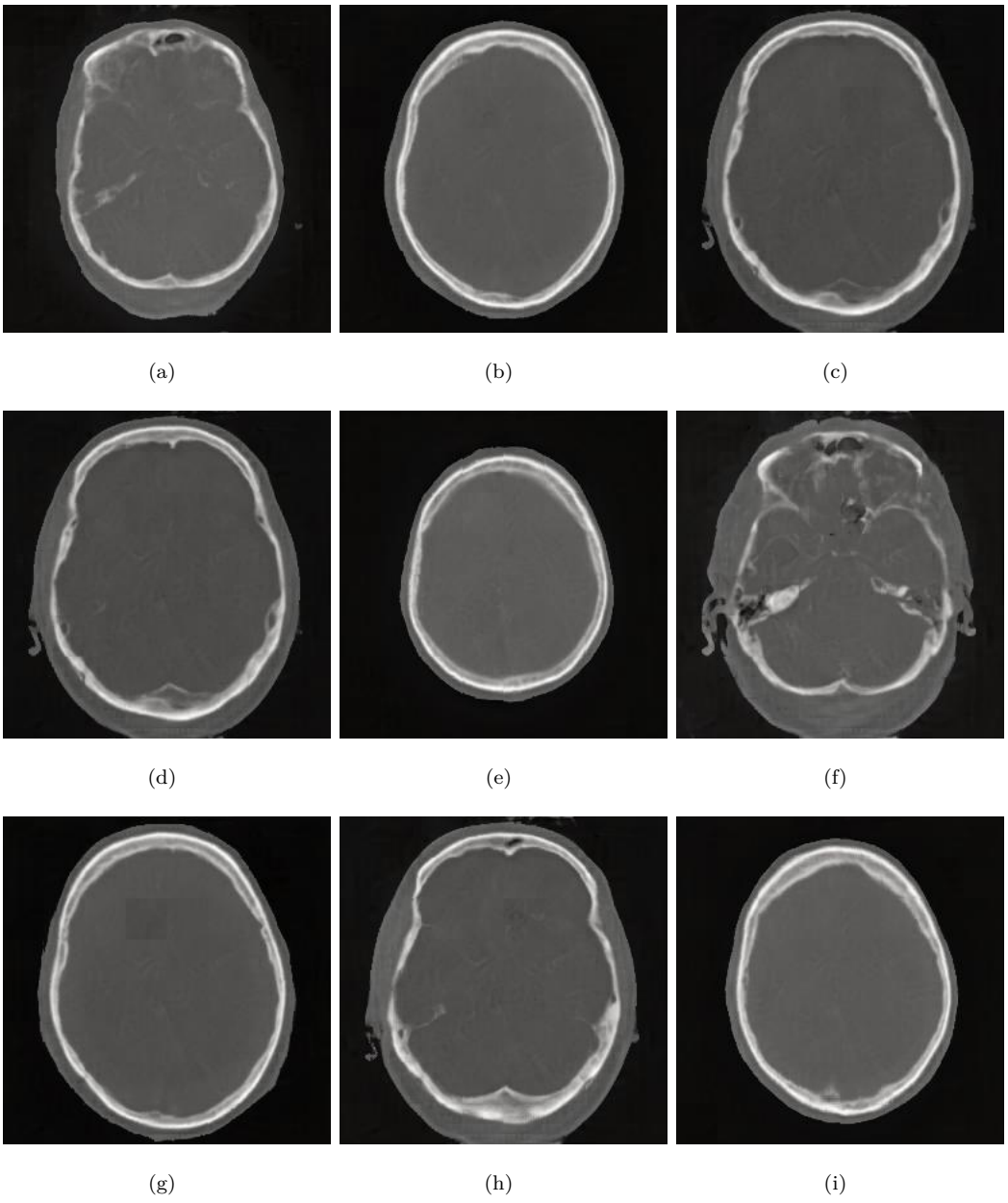
In conjunction with the calculation of the FID for each checkpoint, as illustrated in the plot presented in Figure 7.4, a distinct FID and IS evaluation was conducted for every saved

checkpoint. This additional evaluation was necessary due to the inherent dependency of the FID score on the ordering of the images. To obtain a more representative and generalized FID score, the FID and IS metrics were computed 100 times for 5000 generated images and 5000 images from the evaluation split, each with a different ordering of the images. The resulting FID scores from these 100 calculations were then averaged to obtain the final FID and IS scores. This evaluation approach enables a more robust assessment of the model's performance, accounting for the potential influence of image ordering on the FID metric. The checkpoints presented in this section are the checkpoints with the best FID score, according to Figure 7.4, in different parts of the training process.

First checkpoint

The first checkpoint in this experiment was captured from step 1455628 corresponding to epoch 45. Notably, this checkpoint yielded a commendable FID score of 80.121, accompanied by an inception score of 1.578, with a standard deviation of 0.025. Additionally, the mean FID score amounted to 93.790, while the mean inception score reached 1.577, with a standard deviation of 0.032.

Upon examination of the generated images presented in Figure 7.5, it becomes evident that the diversity between the images has improved greatly compared to the DCGAN data presented in Chapter 6. Upon closer examination, images (c) and (d) exhibit significant similarities. While some differences, particularly in the skull's shape, can be discerned, this may still suggest the occurrence of a mode collapse. That being said, on account of the dataset containing groups of nearly identical images (that being the time steps of the perfusion), it might not be surprising seeing some similarities in the produced data. Further examining the images, intricate details within the skull can be observed, which could potentially be better elucidated through preprocessing techniques.



(a) (b) (c)

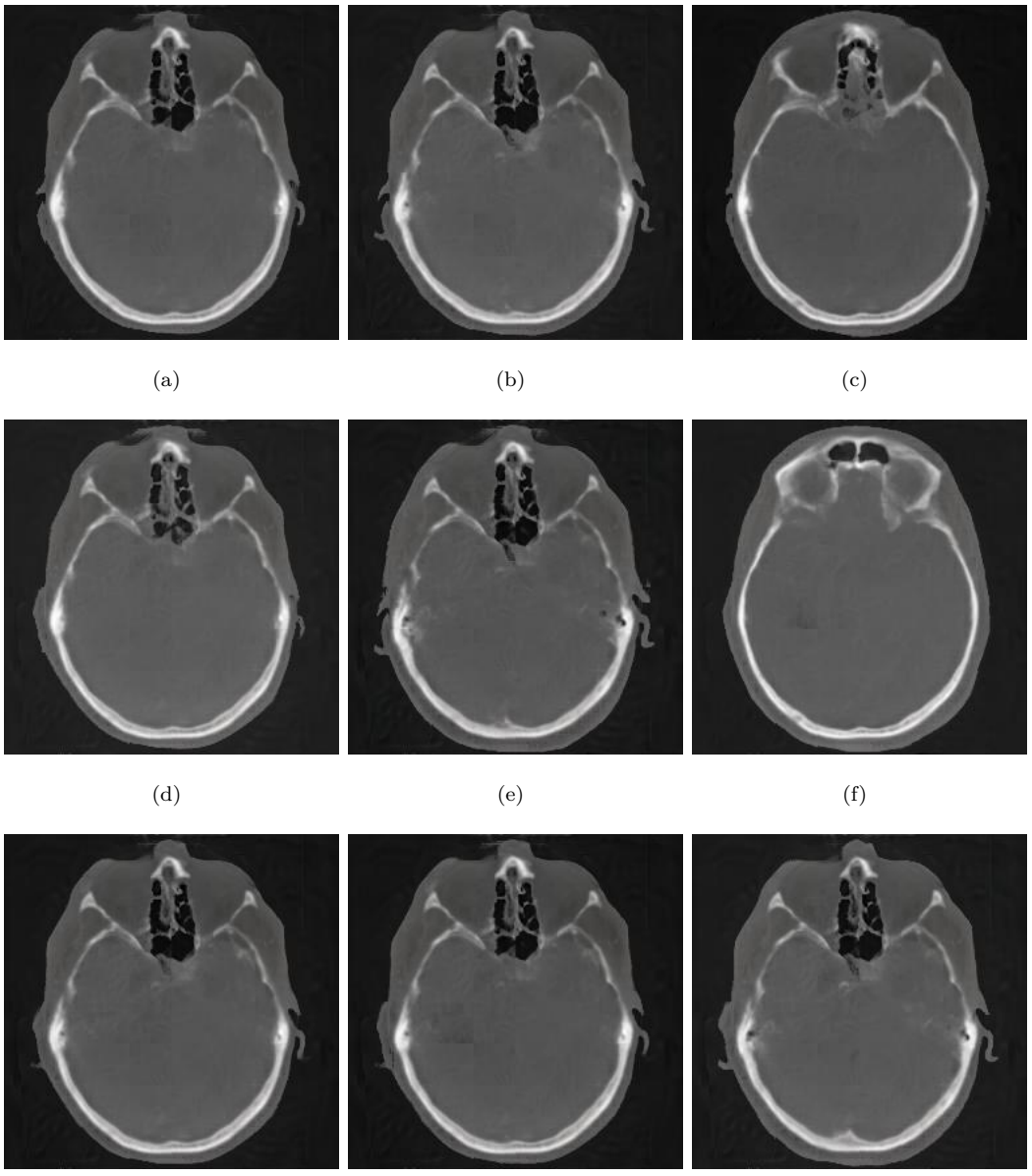
(d) (e) (f)

(g) (h) (i)

Figure 7.5: First set of images generated at step 1455628, given a latent z input.

Second checkpoint

At step 1604572, or epoch 50, the second checkpoint was obtained. This checkpoint demonstrated an FID score of 126.303 and an inception score value of 1.330 with a standard deviation of 0.010. The mean FID score was calculated to be 153.825, while the inception score maintained an average value of 1.330 with a standard deviation of 0.012. These results indicate a moderate decrease in performance during the training, but as the training fluctuates, this is to be expected.



(a) (b) (c)

(d) (e) (f)

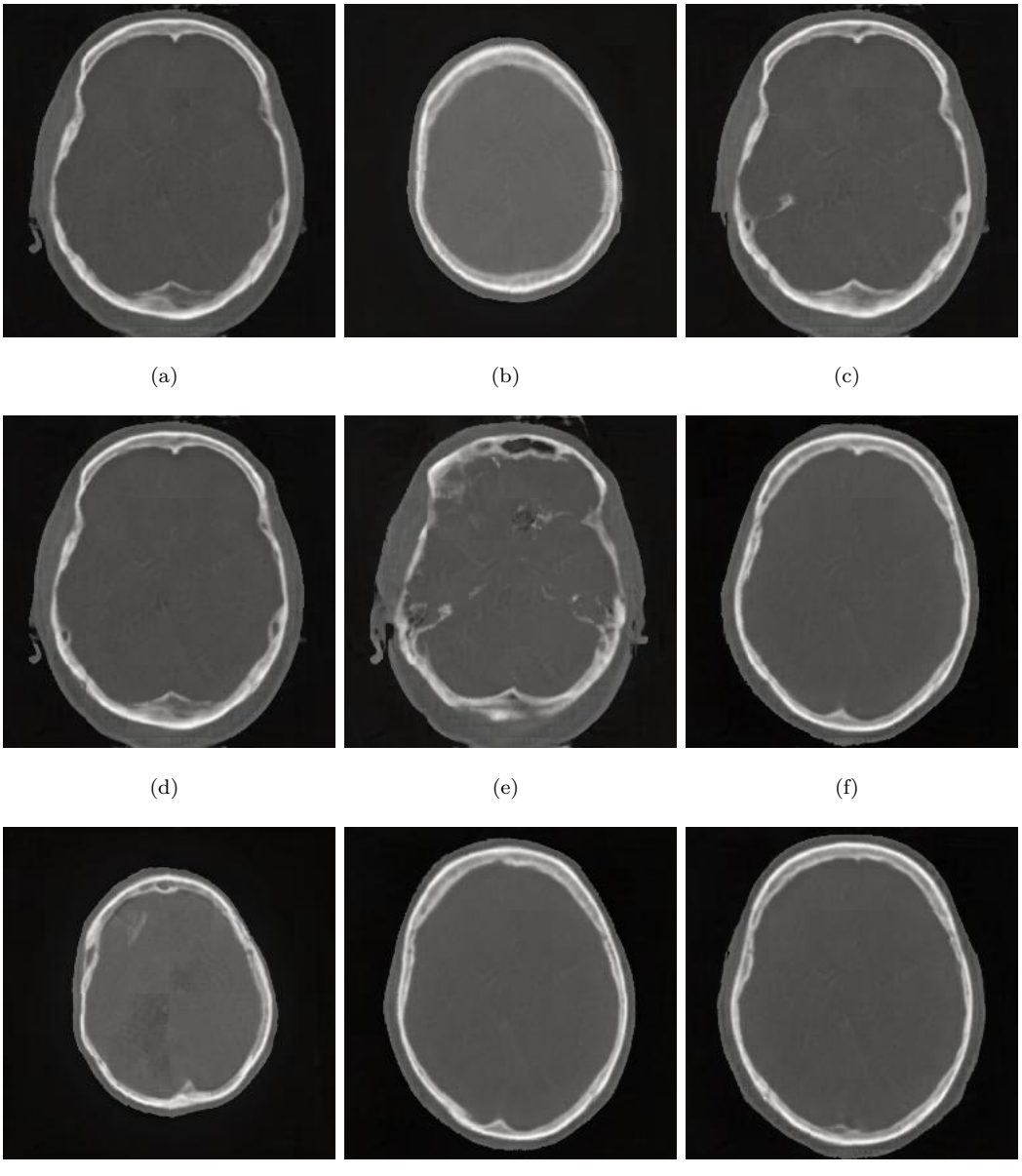
(g) (h) (i)

Figure 7.6: First set of images generated at step 1604572, given a latent z input.

Figure 7.6 presents a selection of images generated at the checkpoint corresponding to step 1604572. It is evident that these generated images exhibit limited diversity, more so favoring a particular mode than the previous iteration. This observation is consistent with the evaluation metrics discussed earlier and raises the issue of whether this might be a mode collapse, as discussed in Section 3.3.1, or potential overfitting. The evaluation of the generator and discriminator losses, as depicted in the plots in Figure 7.1 and 7.2, does not provide substantial evidence indicative of any complications. However, from reviewing the generated data, we can infer that the observed limitations in image diversity are likely attributed to either mode collapse or overfitting.

Last checkpoint

The final checkpoint of the model demonstrated impressive performance, with a mean FID score of 91.887 and a corresponding mean inception score of 1.334, with a standard deviation of 0.017. Notably, the best recorded FID score was 77.406, while the best inception score remained consistent at 1.334. These results indicate that the model has reached a highly satisfactory state, suggesting convergence to desirable values. Further evaluation of the generated images will provide valuable insights into the overall performance of the model at this stage.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 7.7: Images generated at the last checkpoint corresponding to 60 epochs.

The analysis of the images depicted in Figure 7.7 reveals several noteworthy observations similar to those discussed regarding the images generated in the first checkpoint, as shown in Figure 7.5. While certain pairs of images exhibit distinct similarities, the model consistently demonstrates its capacity to generate diverse data encompassing the entire brain. To ascertain whether the generated CTP images capture sufficient details from within the brain structure comparable to real CTP data, the images from the last checkpoint underwent the same preprocessing steps as the original dataset (described in Section 5.2). Initially, the brain extraction step of the preprocessing process was applied to the images in Figure 7.7. The resulting evaluation is visually presented in Figure 7.8.

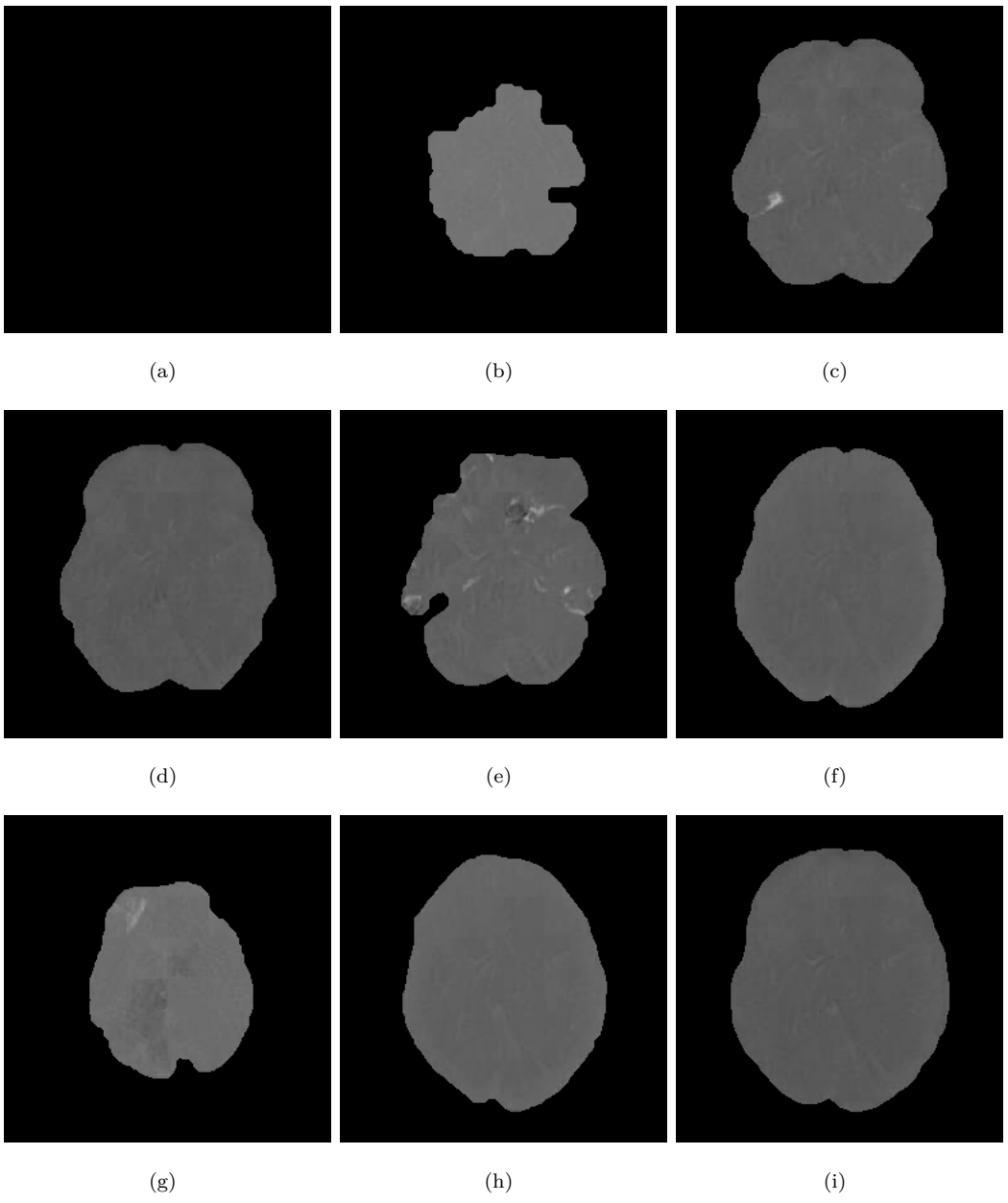
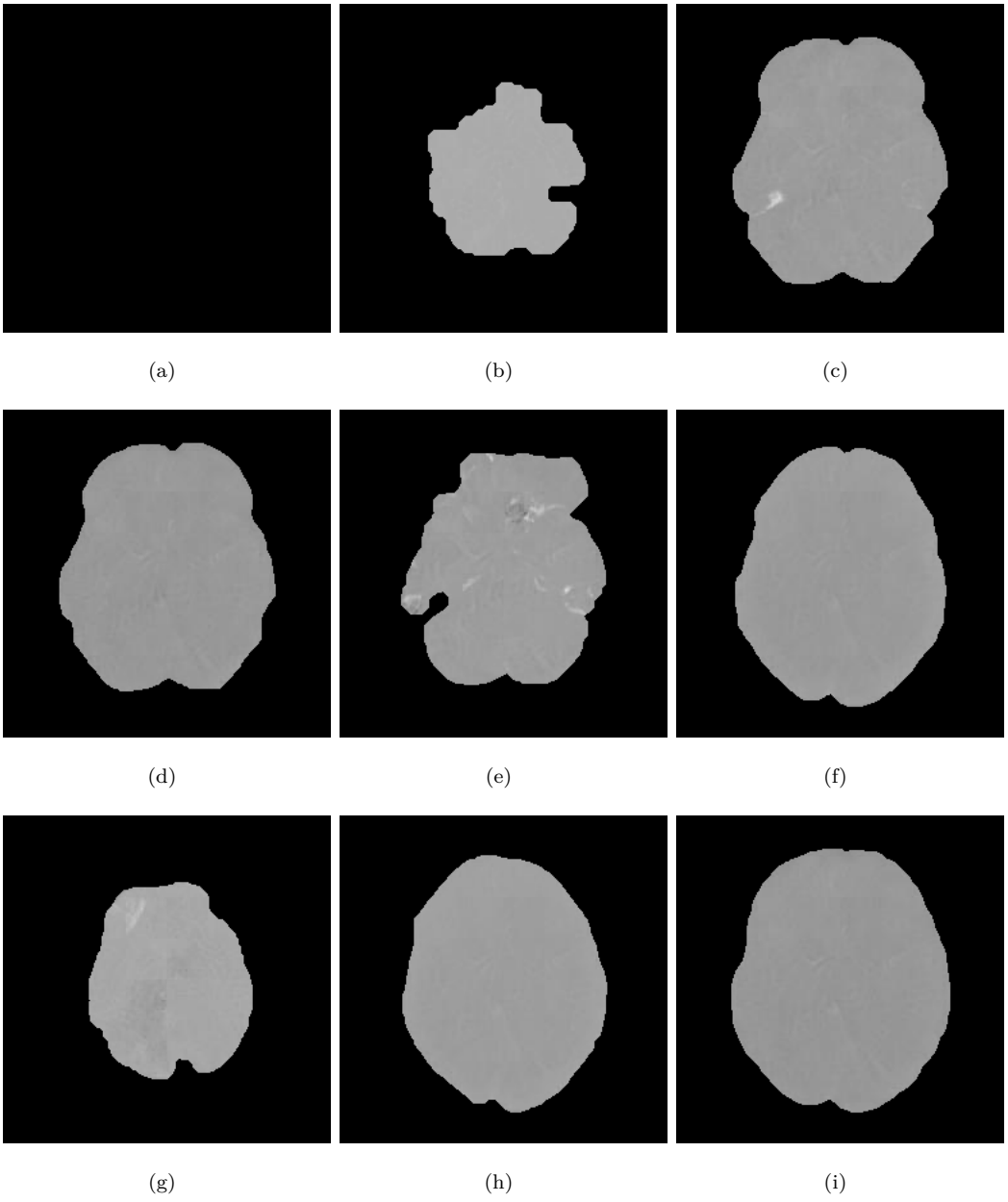


Figure 7.8: The same images as in Figure 7.7, preprocessed using step 3 in 5.2.

As seen in Figure 7.7, the majority of images exhibit a clear gradient between the white part of the skull and the gray matter of the brain, which can be easily detected by the preprocessing algorithm, making the skull removable. However, for image (a) in Figure 7.8, the algorithm might have been unable to detect the skull, possibly due to slightly higher pixel values in the corresponding image (a) in Figure 7.7, explaining why the brain is missing entirely in this image. This is merely conjecture, however, as the preprocessing algorithm is something briefly adopted in this thesis.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 7.9: The same images as in Figure 7.8, but proprocessed with step 4 in 5.2.

After completing step 3 of the preprocessing process as outlined in Section 5.2, the brain-extracted images from Figure 7.8 were further processed to enhance the visibility

of the brain's internal structure, corresponding to step 4 in the preprocessing process. Although some structure is visible, much of it appears to be uniform, resulting in poor image quality. However, a few images in Figure 7.9 exhibit an enhanced vein structure (as in image (e)), which holds the potential for improving the model in the future. While the images generated are currently unsuitable for medical purposes, with further refinement, the model may be able to produce valuable data.

7.4 Experiment 2

Due to the intricate nature of the raw dataset, the model encounters challenges in generating medically applicable images. Consequently, the objective of this experiment is to assess the model's performance on a preprocessed dataset, characterized by reduced complexity and increased image diversity.

The preprocessed dataset employed in this experiment comprises a cohort of 152 patients, yielding a total of 67,530 preprocessed images. The dataset is divided into two distinct subsets for training and evaluation purposes. The training split encompasses 138 patients, contributing a total of 61,680 images. Conversely, the evaluation split encompasses 14 patients, with a total of 5,850 images reserved for evaluation and validation. As the dataset has already undergone the preprocessing process outlined in Section 5.2, no further preprocessing is required for this experiment.

7.4.1 Results and discussion

This experiment underwent a total of 93 epochs with the same hyperparameters and setup as experiment 1. The decision to train the model for a greater number of epochs, in comparison to the previous experiment, was motivated by the lack of convergence during training. The extended duration aimed to investigate the possibility of achieving convergence over time. Regrettably, as illustrated in Figure 7.11, 7.12, and 7.13, the model did not converge despite the prolonged training duration.

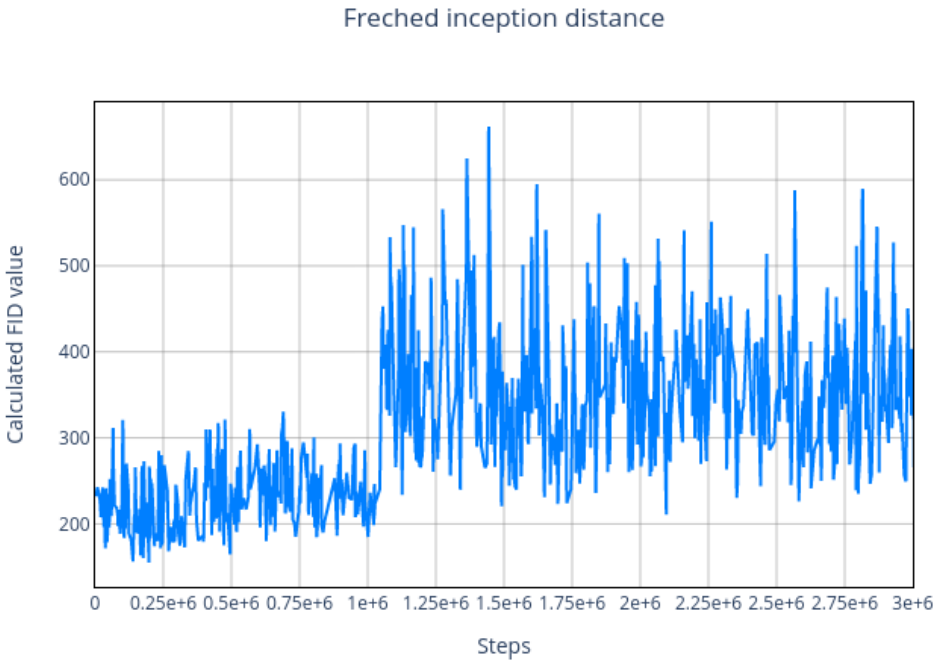


Figure 7.10: The FID score as expounded upon on in Section 3.4.2 calculated for every checkpoint during the training of the HiT-GAN training.

During the initial stages of training, the FID exhibited a range of values between 100 and 300, as observed in Figure 7.10, which is not that far from the patterns observed in the previous section's raw data experiment, as shown in Figure 7.4. This range persisted until approximately step 1e+6 or epoch 32, at which point the FID score became notably more volatile and deteriorated significantly, fluctuating between 200 and 600. For this reason, the presented results are segregated into two subsets: the first set represents the training progress prior to reaching the 1e+6 checkpoint, while the last set signifies the training progress beyond this threshold. Also for this experiment, the same FID and IS calculations as described in the previous section were employed in addition to the FID scores attained in Figure 7.10.

The losses in Figure 7.11, 7.12 and 7.13 also show a sudden change in stability in the training at around the same time as stated for the FID.

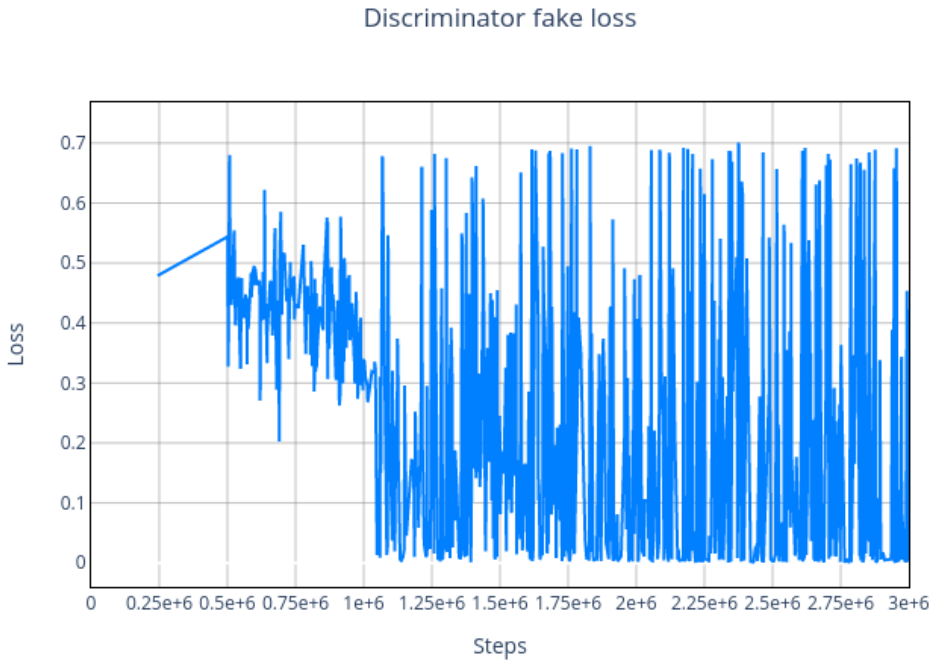


Figure 7.11: The "discriminator fake loss", as expounded upon in Section 3.3, was evaluated over a span of 93 epochs during the training of the HiT-GAN model.

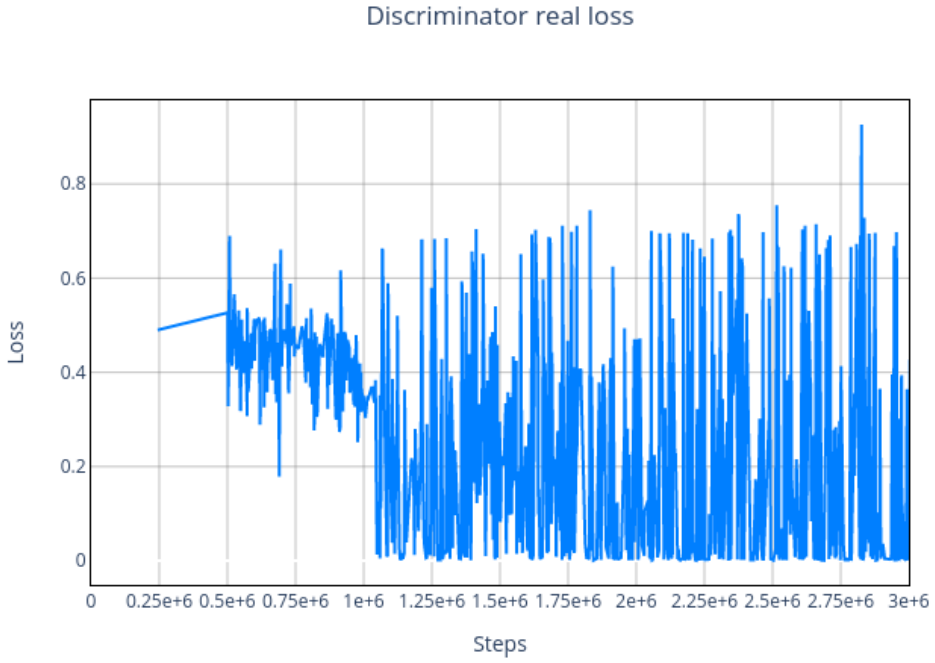


Figure 7.12: The "discriminator real loss", as expounded upon in Section 3.3, was evaluated over a span of 93 epochs during the training of the HiT-GAN model.

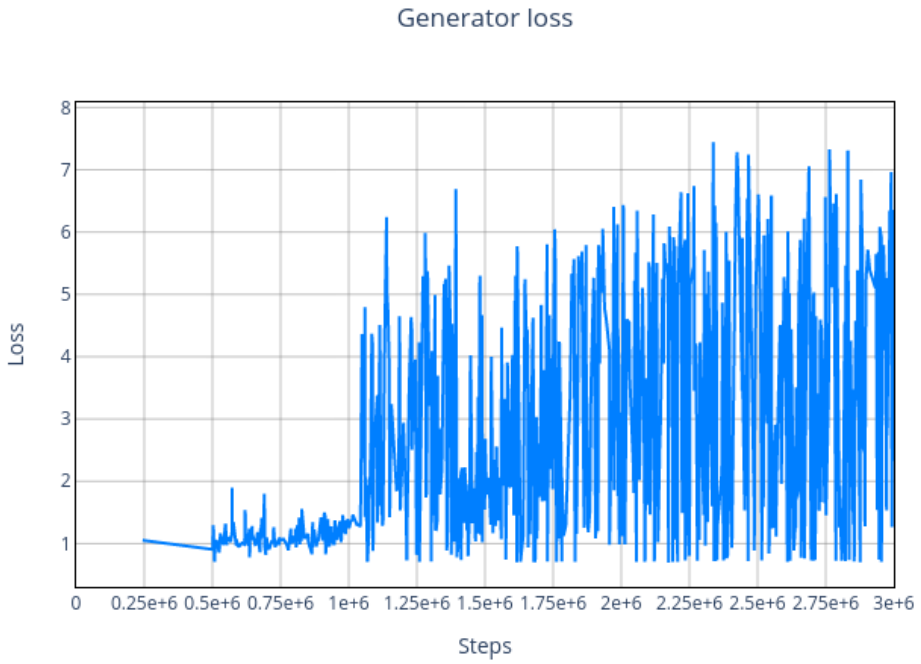


Figure 7.13: The "generator loss", as expounded upon in Section 3.3, was evaluated over a span of 93 epochs during the training of the HiT-GAN model.

To get a better understanding of what happened in the training process in this experiment, the losses in Figure 7.11, 7.12 and 7.13 have been smoothed using moving averages [99] with a window size of 100 in Figure 7.15, 7.14 and 7.16. Smoothing the data grants a better opportunity to see what directions the losses are headed in, or rather where they are converging to.

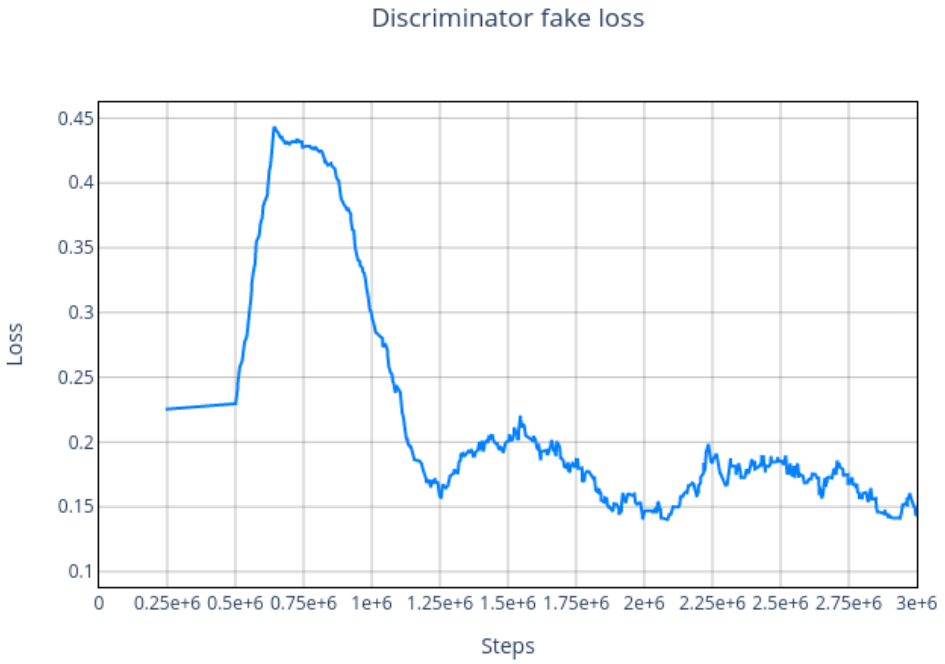


Figure 7.14: The "discriminator fake loss", as expounded upon in Section 3.3, was evaluated over a span of 93 epochs during the training of the HiT-GAN model. The plot is smoothed using the moving average method with a window size of 100.

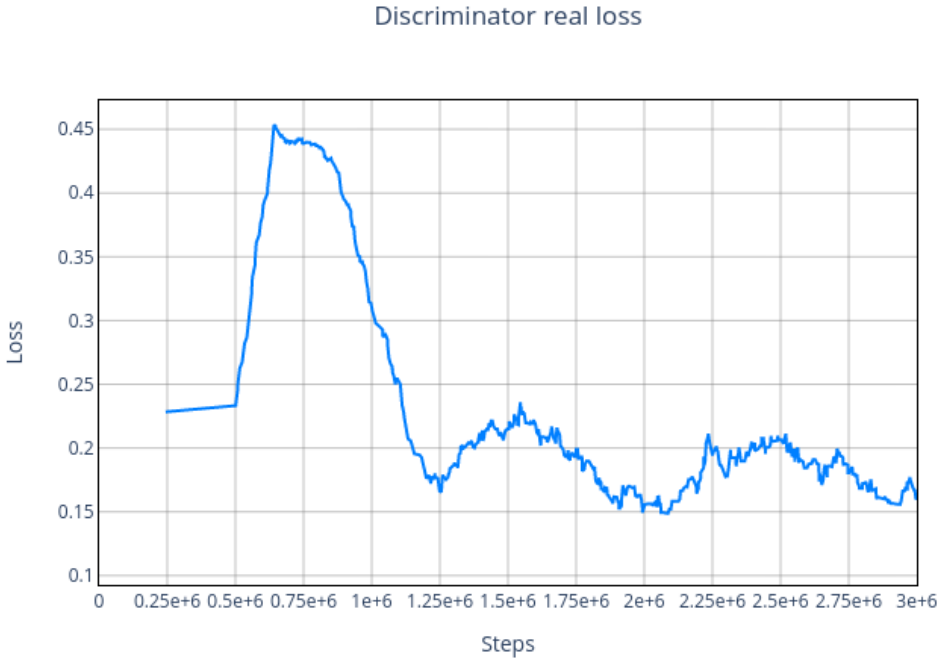


Figure 7.15: The "discriminator real loss", as expounded upon in Section 3.3, was evaluated over a span of 93 epochs during the training of the HiT-GAN model. The plot is smoothed using the moving average method with a window size of 100.

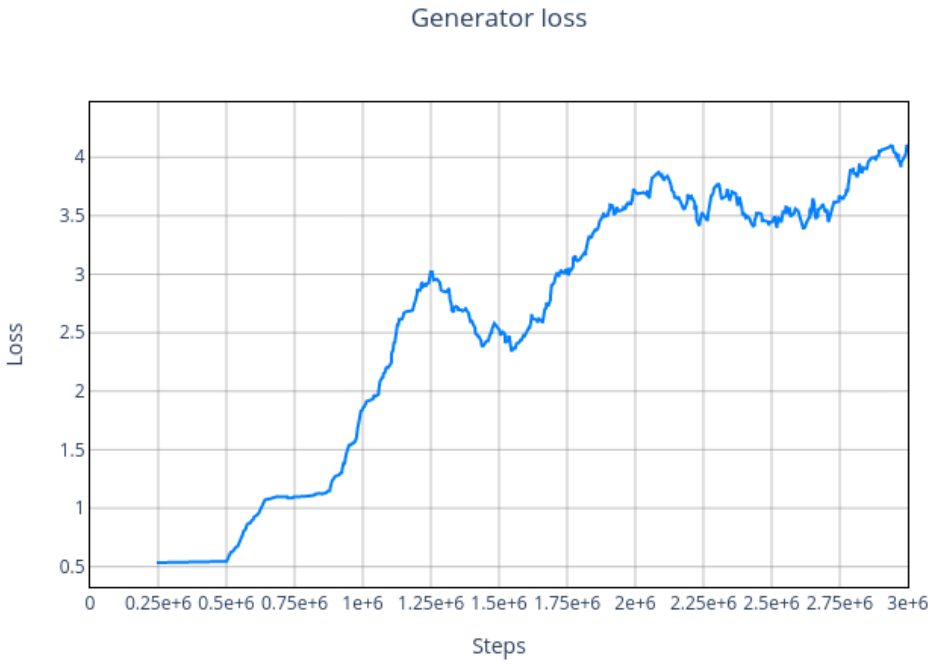


Figure 7.16: The "generator loss", as expounded upon in Section 3.3, was evaluated over a span of 93 epochs during the training of the HiT-GAN model. The plot is smoothed using the moving average method with a window size of 100.

Studying the plots in Figure 7.15, 7.14, and 7.16 the discriminator losses are decreasing over time and the generator loss is increasing over time, similar to the plots in Figure 3.19. Given this trend, it is safe to say the losses will not converge toward their ideal values. To the best of our knowledge, this could be the indication of a convergence failure as described in 3.3.2.

First checkpoint

This particular checkpoint corresponds to step 965280 or epoch 31 and is included because it is the only checkpoint that was saved before step 6e+10. The average FID and IS values recorded were 269,902 and 1,207 respectively, however, the best scores achieved were 246,030 and 1,207. These values are quite bad compared to the values attained by using the raw dataset, which was suspected as this is quite early in the training process and by the unstable training shown in the loss plots. Figure 7.17 displays the images generated at this checkpoint.

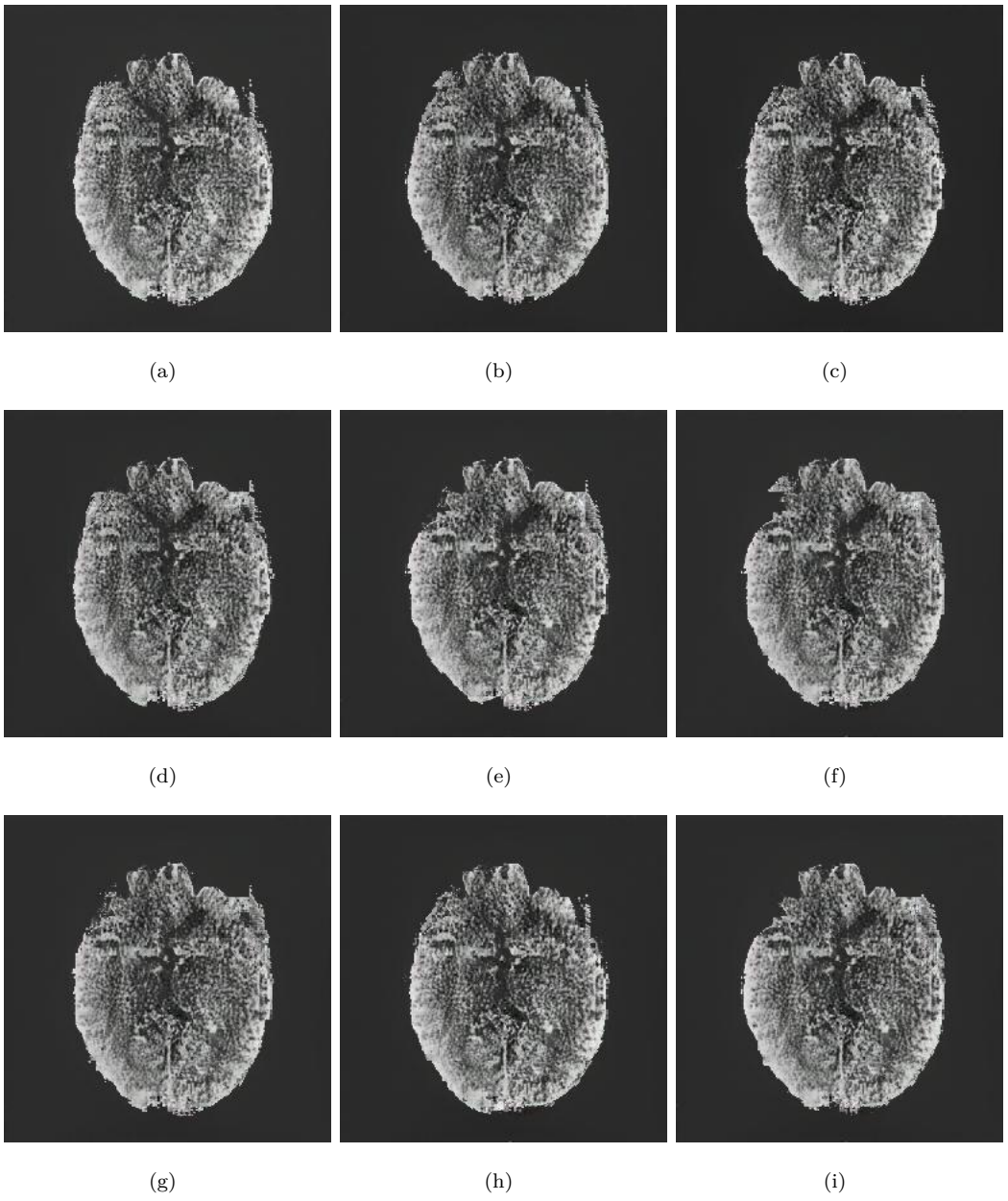
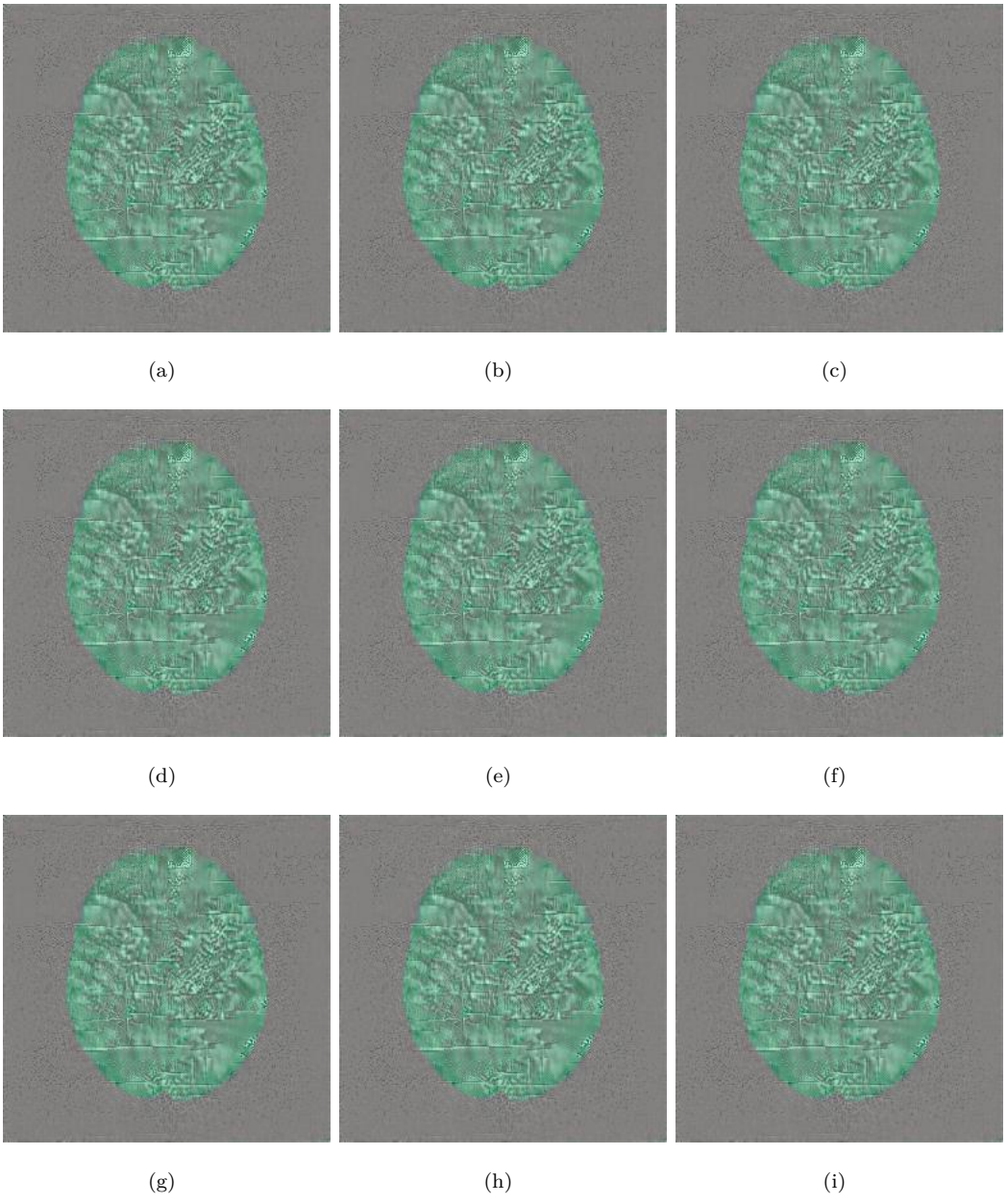


Figure 7.17: Generated images from checkpoint 965280 corresponding to epoch 31.

The analysis of the FID score at this checkpoint indicates that the generated images exhibit subpar quality. However, upon closer examination, it is evident that the model is capable of capturing certain aspects of the brain and vessel structures, with a relatively higher resolution compared to previous checkpoints. Nevertheless, the generated data appears to be affected by some artifacts. In terms of diversity, the images demonstrate similar characteristics as observed in earlier checkpoints, wherein only a limited number of distinct images are generated with some minor variations.

Last checkpoint

This checkpoint is the last conducted step in the training process, corresponding to step 3e+6 or epoch 97. The average FID and IS values achieved by this checkpoint are 482.964 and 1.003, respectively, with the best values obtained at 445.504 and 1.003. Unfortunately, this checkpoint is the worst-performing one in this study, indicating that our experiment might have encountered a convergence failure, as discussed earlier in this experiment.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 7.18: Generated images from the last checkpoint in this experiment, corresponding to epoch 97.

According to the high FID score, it is evident that the images produced for this particular checkpoint (as depicted in Figure 7.18) exhibit substandard quality and significant levels of noise, resembling the instance of convergence failure exemplified in Figure 3.20. This further supports our assertion that the observed outcome is indeed a manifestation of convergence failure. Moreover, the images also demonstrate the HiT-GAN model's capability of generating images with three color channels (RGB). This reveals that the model has previously learned the correct combination of color values needed to produce gray-scale images.

Chapter 8

Comparative analysis

The following chapter will give an overview of this thesis. The objectives of the thesis that were introduced in Chapter 1 will be discussed based on what was presented in Chapters 2-7. Additionally, the results that were obtained in this study will be presented, giving a comprehensive comparison. Furthermore, a variety of the findings and challenges in this study will be discussed, including a comparative analysis of the results acquired from the DCGAN and HiT-GAN models as well as some proposed improvements to some of the methods conducted.

In this thesis, the DCGAN model proposed by Korkmaz [3] has been employed to synthesize both raw and preprocessed CTP data. The generated images from both experiments were deemed acceptable representations of their respective datasets, despite having higher FID values than expected. These images served as a reasonable baseline for comparison with the HiT-GAN model, particularly in terms of visual quality. Regarding the HiT-GAN results, initially, the generated images exhibited high quality, capturing the necessary details to undergo preprocessing using the same method as the preprocessed dataset. However, these results fell short when compared to the real samples from the preprocessed dataset. Unfortunately, the second experiment encountered convergence failure, rendering the obtained results mostly unusable. Nonetheless, considering the promising outcomes from other HiT-GAN runs, with additional efforts, this experiment could potentially yield commendable results. Overall, the HiT-GAN model outperformed expectations, displaying FID values comparable to the best results achieved by models working on similar datasets. Moreover, the model generated high-quality CTP images and produced slices from different elevations of the head, showcasing its superiority over the alternative model, despite its own limitations.

The evaluation metrics and the obtained results are further discussed in Section 8.1, providing a comprehensive analysis of the outcomes from both models.

8.1 DCGAN versus HiT-GAN

The FID results presented in Table 8.1, along with the visual analysis of the generated data shown in Figure 6.3 and 7.7, demonstrate that the HiT-GAN model outperforms the DCGAN model proposed by Korkmaz et al. [3] in terms of performance, especially for the raw dataset. A careful examination of the outputs from the last checkpoint, as depicted in Figure 7.7, reveals that the HiT-GAN model generates data with enhanced diversity and finer details compared to the DCGAN outputs shown in Figure 6.3. These findings highlight the potential benefits of incorporating transformers within GAN models, even in the domain of medical image data.

Table 8.1: All FID and IS results obtained in this study. The best-registered performance is denoted in **bold text**.

Model	Dataset	Epoch	Best FID	Best IS	Mean FID	Mean IS
DCGAN	Raw	127	143,039	1,146	168,157	1,146
DCGAN	Preprocessed	127	200,239	1,501	222,837	1,501
HiT-GAN	Raw	45	80,121	1,578	93,790	1,577
HiT-GAN	Raw	50	126,303	1,330	153,825	1,330
HiT-GAN	**Raw**	**60**	**77,406**	**1,334**	**91,887**	**1,334**
HiT-GAN	Preprocessed	31	246,030	1,207	269,902	1,207
HiT-GAN	Preprocessed	97	445,504	1,003	482,964	1,003

The results obtained from the second experiment, which involved the use of the preprocessed dataset, reveal that the DCGAN model outperforms the HiT-GAN model. This performance difference can be attributed to the failure of the HiT-GAN model to converge. By examining the HiT-GAN model's outcomes of the initial checkpoint for the generated preprocessed data in Figure 7.17, along with the clear results observed for the raw dataset in Figure 7.7, one can infer that the HiT-GAN model would have outperformed the DCGAN model had it successfully converged. Analyzing the generated data from both models in Figures 6.4 and 7.17, it is evident that the DCGAN model captures a greater variety of brain shapes for the preprocessed dataset; however, some data loss is observed in the process. Conversely, the output of the HiT-GAN model, as depicted in Figure 7.17, exhibits more intricate details and structures within the brain images, albeit with a limited diversity among the generated samples.

The obtained IS values in this thesis are suboptimal and do not align with the evaluation criteria established for this study, as elaborated in Section 7.2. However, it is worth noting that a lower IS value can indicate a lower level of diversity among the generated images [44]. Although the medical datasets, in general, are less diverse than Imagenet or similar datasets, the declining IS values for the HiT-GAN model on the raw dataset, presented in Table 8.1 could indicate a decreasing trend in diversity throughout the training process for the output generated by the HiT-GAN model.

For the raw dataset, the DCGAN model underwent a total training time of 195 hours, approximately spanning 8 days. Conversely, the HiT-GAN model was trained for 186 hours, also equating to nearly 8 days. Despite the comparable training durations, the DCGAN completed more than double the number of epochs compared to the HiT-GAN model. It is worth noting that the HiT-GAN model requires fewer epochs to achieve satisfactory results, as evidenced by the poor output quality of the DCGAN model, particularly up until epoch 127.

Furthermore, the HiT-GAN model exhibits high parallelizability, enabling training across multiple GPUs concurrently. This aspect raises the question of available computing power as a determining factor in utilizing the model effectively.

8.2 Comparison with related work

As outlined in Chapter 7, it should be noted that the FID metric utilized for evaluating the models in this study is pre-trained on the ImageNet dataset. Consequently, the FID scores obtained for the generated images in this thesis may appear comparatively higher. Therefore, it is imperative to compare the FID scores with those of GAN models trained on similar datasets.

Table 4.1 presents the FID values for various GAN models trained on comparable medical datasets. Considering the values provided in that table, the FID scores achieved by the HiT-GAN model on the raw dataset are deemed satisfactory.

8.3 Improvements

First, it is noteworthy that the original images in the dataset were converted from the lossless TIFF format to the lossy JPG format to accommodate the model's requirements. While there is no explicit evidence suggesting that this conversion adversely affects image quality, it may be prudent to consider utilizing a lossless format such as PNG to mitigate any potential loss of information.

Additionally, it is important to note that the original design of the HiT-GAN model by the authors was intended for execution on multiple TPUs. However, due to the constraints of this study, which involved the utilization of only two GPUs, one for training and one for evaluation during training, the batch size employed in this thesis was limited to two. This is in contrast to the authors' recommendation of a batch size of 256 [94].

Also, the computation of the FID and IS metrics for the two models examined in this thesis could incorporate the standard deviation for the 100 iterations (as described in Section 7.3.1), thereby yielding more precise representations.

The original images, as detailed in Section 5.3, possess a resolution of 512 x 512 pixels. However, in this study, a resolution of 256 x 256 pixels was adopted due to limitations inherent in the HiT-GAN model, which supports only resolutions of 128 x 128, 256 x 256, or 1024 x 1024 pixels. Extending the model to incorporate a resolution of 512 x 512 pixels would mitigate the loss of information incurred by downsampling the images to a lower resolution.

The findings presented in this thesis highlight the challenges associated with synthesizing CTP data, considering the limited availability of diverse datasets in the medical domain. Despite these challenges, it is demonstrated that it is possible to generate images that possess certain properties resembling the original CTP images. The dataset utilized in this study consists of data obtained from a 4D study, where the temporal changes in the images are minute. As a result, it is reasonable to assume that the 30 images in time for each slice are nearly identical. This inherent lack of diversity in the dataset poses a hindrance, particularly in the context of GAN training, which is known for its instability [10]. One could surmise that given the groups of nearly identical images which make up the dataset, the supposed mode collapse which is experienced in the results is simply the model trying to capture different time instances of the perfusion. Meaning that given a dataset of purely dissimilar images, the model would output exclusively unique images, with no major resemblances between them. To enhance the training process this way, it could be beneficial to employ a dataset characterized by greater diversity, as it would contribute to a more robust and effective training of the GAN model.

8.4 ViTGAN approach

ViTGAN was also selected as a model to be utilized for CTP image synthesis. This was because the model is essentially the most rudimentary, baseline combination of vision transformers and GANs, and would serve as an interesting model for comparison. Unfortunately, after spending time implementing it and making it fit the designated dataset, the model would not produce any valuable output. After running for a period of 10-30 minutes, the losses would lock onto a value of 0.693, in a way resembling convergence failure. This value, however, relates to the loss function's (BCE) input value of 0.5, which might suggest that for whatever reason, the discriminator outputted a static 0.5 probability of the input image being real, which did not give the generator any input to improve itself from. As a result, the generator would output pure noise. Given

this undiagnosed issue and the lack of meaningful output data in addition to the time pressure related to the project itself, it was decided that the main focus would lie on the HiT-GAN model which did work, and there would be no reason to include the ViTGAN with its own chapter.

Chapter 9

Conclusions

9.1 Conclusion

This thesis delved into the potential applications of vision transformers as alternatives to the conventional employment of CNNs in GANs, within the domain of synthesizing CTP images. The primary objective was to augment the limited scale of typical medical datasets, thus providing a sufficient quantity of training data for medical machine-learning tasks.

Initially, the DCGAN model, as proposed by Korkmaz, served as the foundation and starting point for this thesis. Subsequent to its adoption, certain modifications were introduced, and image generation and evaluation were conducted on both the raw dataset and the preprocessed dataset.

Subsequently, vision transformers were integrated into the GAN generation process through the introduction of the HiT-GAN model. This model underwent training on both datasets, and the generated images underwent evaluation through the FID metric. The findings demonstrated that although the generated images captured specific aspects of the brain structure, they exhibited limited diversity within their visual characteristics. This observation led to the conclusion that this outcome could be attributed to either a mode collapse phenomenon or the inherent limitations of the low-diversity dataset employed in this study.

This work might serve as a part of the baseline for a data synthesis model that will be able to generate artificial CTP data that captures every minute detail present in the real-world data. A breakthrough in this sector would provide applied algorithms with all the training data they would need, possibly leading to a revolution within the medical sector with regard to diagnostics, not to mention the response time for treatments. The

different methods explored in this project may aid other researchers considering similar approaches.

9.2 Future directions

Adapting the HiT-GAN model to incorporate conditional capabilities (as described in Section 3.2) would allow for the utilization of 3D data (5.5). By treating slices as classes, it becomes possible to represent an entire patient using 14 slices. This approach provides a more comprehensive representation compared to generating a single random 2D (5.4) slice for each patient. To achieve this, the dataset needs to be restructured into folders of slices instead of patients. Each folder would be assigned the same label, with Slice 01 assigned label 0, Slice 02 assigned label 1, and so forth.

In order to accommodate multiple inputs and outputs for the additional label channel, the HiT-GAN model would need to be rewritten using the functional API (as documented in [100]) instead of the Sequential API. Both the generator and discriminator components would need to incorporate label embedding into the image, resulting in an extra channel of the same size as the image. These label embeddings would then be concatenated to the image. This can also extend to 4D data (5.6) resulting in the ultimate CTP data representation.

The HiT-GAN model only includes ViT in the generative process. In future works the implementation of a ViT-based discriminator, as in [65], could be studied. This might reduce inductive bias from the discriminating process and improve the quality of the generated images by allowing the discriminator to better understand the content of the images.

Appendix A

Appendix contents

The code used in this thesis, as well as the accompanying README.md, can be found on https://github.com/orjanvier/master_hitgan

The following pdfs are included in this chapter:

- Master thesis poster
- NOBIM conference abstract
- NOBIM presentation

Using vision transformer to synthesize computed tomography perfusion images in ischemic stroke patients

Ørjan Vier[1], Carl Henrik Hovland Christiansen[1], Luca Tomasetti[1], Mahdieh Khanmohammadi[1] and Kathinka Dæhli Kurz[2]

[1] Department of Electrical Engineering and Computer Science, University of Stavanger, Norway
[2] Department of Radiology, Stavanger University Hospital, Norway

MOTIVATION & BACKGROUND

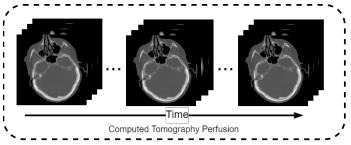
- **Computed tomography perfusion (CTP)** imaging is routinely used for diagnosing cerebral stroke and determining the extent of damage to the brain [1].
- Therefore, automatic segmentation [2] of the dead tissue **(ischemic core)** and salvageable tissue **(penumbra)** is needed.
- This is as opposed to the time-consuming and imperfect process of manually examining parametric maps derived from the **CTP** images.
- However, automatic techniques based on neural networks require **immense amounts** of labeled data to return promising results.
- We propose to tailor a High-resolution Transformer-based Generative Adversarial Network **(HiT-GAN)** [3] that employs a hierarchical structure with multiple generative stages in combination with a discriminator to generate **CTP** data.

DATA MATERIALS

Image generated using *Simplified*

CTP scans from 156 patients divided:

1. Large Vessel Occlusion (LVO)
2. Non LVO
3. Without IS (WIS)



APPROACH & EXPERIMENTS

- Generated synthetic CTP images extending the Deep Convolutional GAN (DCGAN) model developed in [4], serving as a baseline for evaluation.
- Generated synthetic CTP images utilizing the newly implemented HiT-GAN [3] model, thereby introducing vision transformers into the generative process.
- Evaluated the data generated by each model individually, comparing it against the original training data, to assess its quality and fidelity using the Fréchet Inception Distance (FID) metric.
- Conducted a comparative analysis between the generated CTP data from DCGAN and HiT-GAN, examining their respective characteristics, strengths, and weaknesses.

Stage nr.	Resolution	Resolution category
01	8x8	Low
02	16x16	Low
03	32x32	Low
04	64x64	Low
05	128x128	High
06	256x256	High

Table. 1. Different resolutions in the HiT-GAN model.

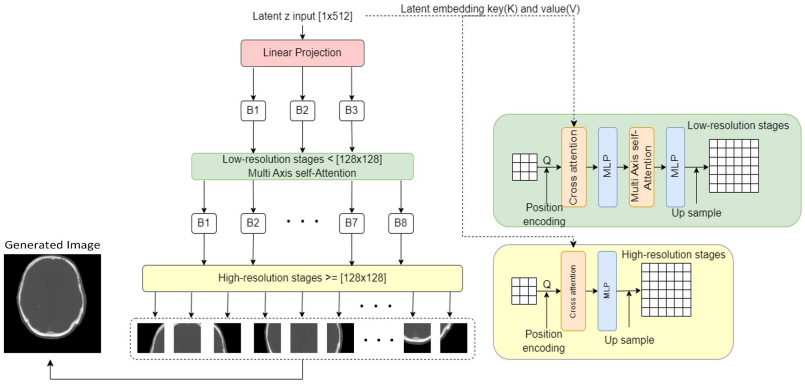


Fig. 1. General overview of the various resolution stages involved

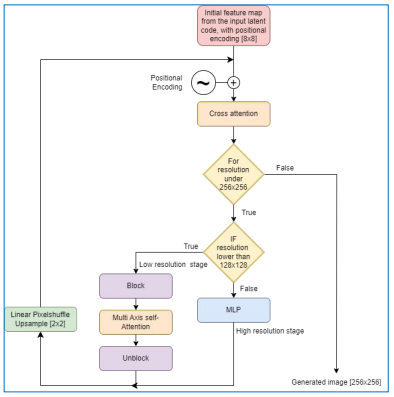


Fig. 2. Visual overview of the HiT-GAN architecture.

RESULTS AND CONCLUSION

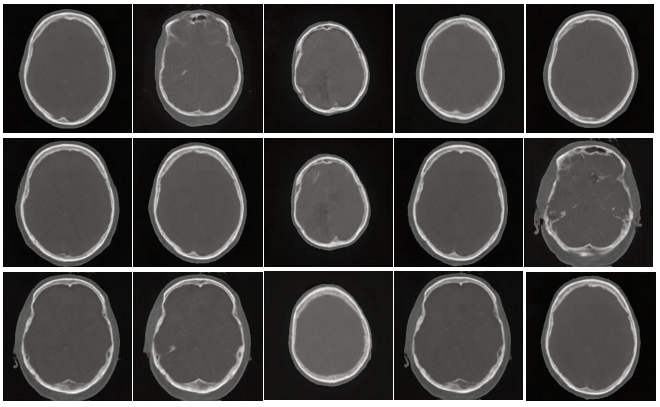
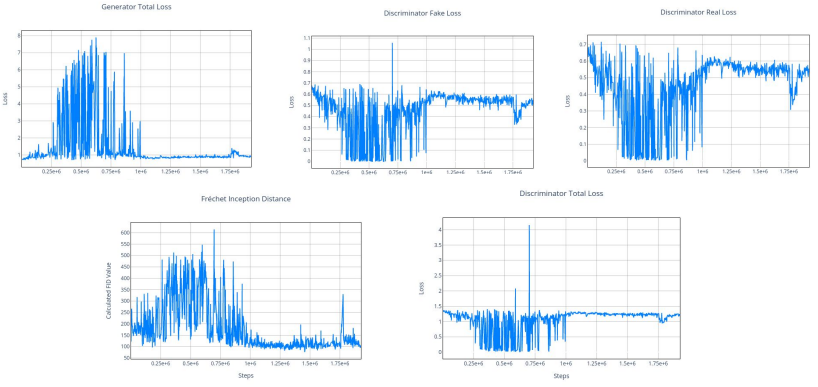


Fig. 3. Images generated by the HiTGAN model



Checkpoint	Best FID	Best IS	Mean FID	Mean IS
1455628	0.121	1.578	93.790	1.577
1604572	126.303	1.330	153.825	1.330
Last	77.406	1.334	91.887	1.334

Table 2. Best & Mean (+ standard deviation) test results.

CONCLUSION

- We proposed an implementation of a **transformer-based GAN** for the synthesis of raw CTP data.
- The proposed implementation demonstrated benefits regarding **image quality and diversity**. An average FID score of 91,887 is achieved for the last generated images.
- This study is a first step of exploring vision transformer-based models for CTP synthesis; further research is required to further improve this implementation

REFERENCES

[1] Heit JJ, Wintermark M. "Perfusion computed tomography for the evaluation of acute ischemic stroke". Stroke 2016: 47(4): 1153–1158.
[2] Tomasetti L, Hansen S, Khanmohammadi M, Engan K, Hallesli LJ, Kurz KD, Kampffmeyer M. Self-supervised fewshot learning for ischemic stroke lesion segmentation, 2023.
[3] Zhao L, Zhang Z, Chen T, Metaxas DN, Zhang H. "Improved transformer for high-resolution gans", 2021.
[4] Korkmaz M. Synthesising training data with generative adversarial networks (GANs) in computed tomography perfusion. Master's thesis, University of Stavanger, Norway, 2021.

Using vision transformer to synthesize computed tomography perfusion images in ischemic stroke patients

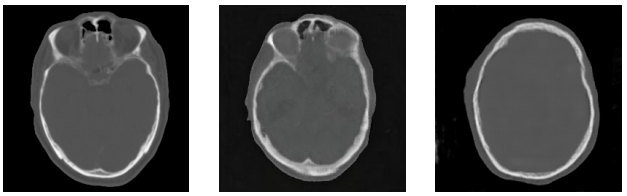
Ørjan Vier[1], Carl H. Christiansen[1], Luca Tomasetti[1], Mahdieh Khanmohammadi[1] and Kathinka Kurz[2]
[1]*Department of Electrical Engineering and Computer Science, University of Stavanger*
[2]*Department of Radiology, Stavanger University Hospital*

Computed tomography perfusion (CTP) imaging is routinely used for diagnosing cerebral stroke and determining the extent of damage to the brain [1]. It guides treatment decisions, which is why timely identification of the affected area is crucial in ischemic stroke patients [2]. Therefore, automatic segmentation of the dead tissue (ischemic core) and salvageable tissue (penumbra) is needed. This is as opposed to the time-consuming and imperfect process of manually examining parametric maps derived from the CTP images. Automatic techniques based on neural networks require immense amounts of labeled data to return promising results, however, and obtaining ground truth in medical images is notoriously cumbersome. Regardless, self-supervised segmentation to separate the ischemic core from the penumbra has been the center of attention currently [3].

Self-supervised segmentation requires a large training set we propose to obtain by synthesizing CTP images. Previous projects [4] employed deep convolutional generative adversarial networks (DCGANs) consisting of a generator network and a discriminator network. The generator network usually comprises transposed convolutional layers, while the discriminator network uses standard convolutional layers. Although DCGANs can generate visually appealing and realistic images, synthesizing CTP images needs a network that can produce high-resolution images with fine details.

We propose to tailor a High-resolution Transformer-based Generative Adversarial Network (HiT-GAN) [5] that employs a hierarchical structure with multiple generative stages in combination with a discriminator to generate CTP data. HiT-GAN is based on vision transformers [6] and uses self-attention mechanisms to capture long-range dependencies in the image at different scales. Through an attention-based learning approach, vision transformers have been shown to outperform convolutional neural networks (CNNs) regarding both computational efficiency and accuracy [6]. HiT-GAN uses multi-axis blocked self-attention that captures local and global dependencies within non-overlapping image blocks in parallel. It allows the model to capture both global and local image features to improve the quality of the generated images.

To train our model we used CTP images from 156 patients acquired by the Stavanger university hospital. These patients are separated into three groups based on vessel occlusion: 1) patients with large vessel occlusion (LVO) n = 78, 2) patients with non-large vessel occlusion (non-LVO) n = 63, and 3) patients with no vessel occlusion n = 15. In total 68 130 raw-data images of size 512 x 512 were included in this study. These images have been normalized to include grayscale values between 0-255 and downsampled to size 256 x 256. The results of our experiments presented Fréchet Inception Distance (FID) scores of 77.406 for the HiT-GAN, compared to 143.039 for the DCGAN, evaluated on a set of 5 000 images each. The resulting generated images are compared to a real sample in Figure 1. The HiT-GAN model showed promising results in generating two-dimensional images. Generating 3D CTP data by conditioning the model with labeled brain slices is our future focus.



(a) Real image (b) HiT-GAN (c) DCGAN

Figure 1: Comparison of generated images to an original. The first two images, 1a and 1b, capture eye sockets and the nasal region, while the last image 1c is purely of the skull.

References

[1] Heit JJ, Wintermark M. Perfusion computed tomography for the evaluation of acute ischemic stroke. Stroke 2016; 47(4):1153–1158.

[2] Demeestere J, Wouters A, Christensen S, Lemmens R, Lansberg MG. Review of perfusion imaging in acute ischemic stroke. Stroke 2020;51(3):1017–1024.

[3] Tomasetti L, Hansen S, Khanmohammadi M, Engan K, Høllesli LJ, Kurz KD, Kampffmeyer M. Self-supervised few-shot learning for ischemic stroke lesion segmentation, 2023.

[4] Korkmaz M. Synthesising training data with generative adversarial networks (GANs) in computed tomography perfusion. Master's thesis, University of Stavanger, Norway, 2021.

[5] Zhao L, Zhang Z, Chen T, Metaxas DN, Zhang H. Improved transformer for high-resolution gans, 2021.

[6] Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, Dehghani M, Minderer M, Heigold G, Gelly S, Uszkoreit J, Houlsby N. An image is worth 16x16 words: Transformers for image recognition at scale. CoRR 2020; abs/2010.11929. URL https://arxiv.org/abs/2010.11929.

Using vision transformer to synthesize computed tomography perfusion images in ischemic stroke patients

Carl Henrik Hovland Christiansen[1], **Ørjan Vier**[1], Luca Tomasetti[1], Mahdieh Khanmohammadi[1], Kathinka Dæhli Kurz[2]

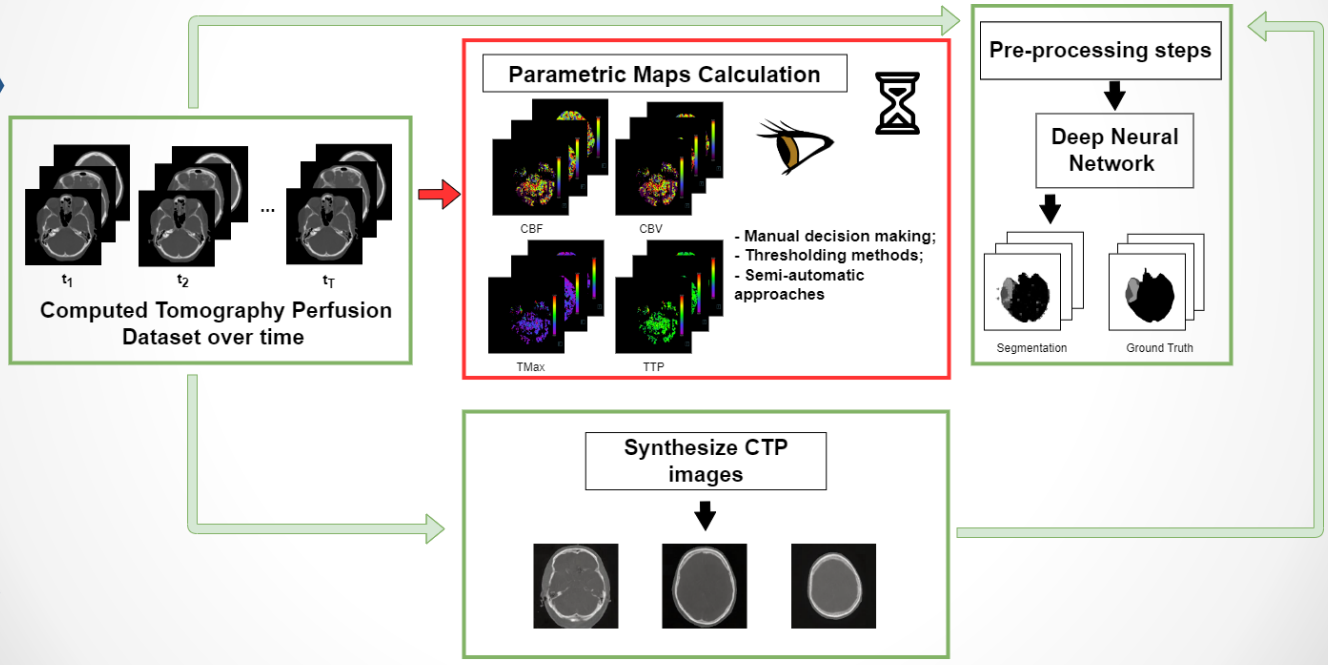
[1] Department of Electrical Engineering and Computer Science, University of Stavanger, Norway
[2] Department of Radiology, Stavanger University Hospital, Norway

University of Stavanger
uis.no

6/5/2023

University of Stavanger

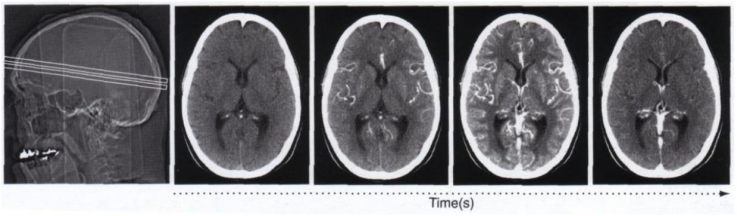
Overview



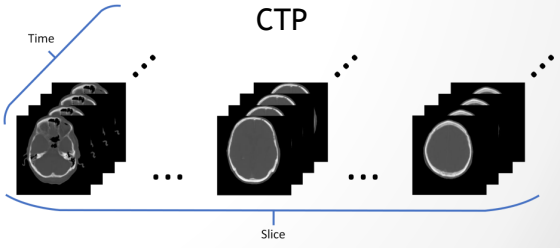
University of Stavanger

Problem definition

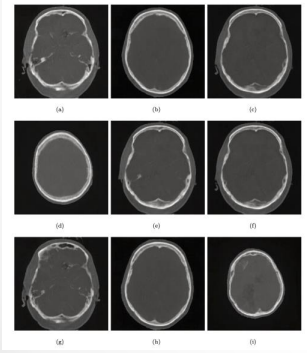
Patient admission



Angiography process



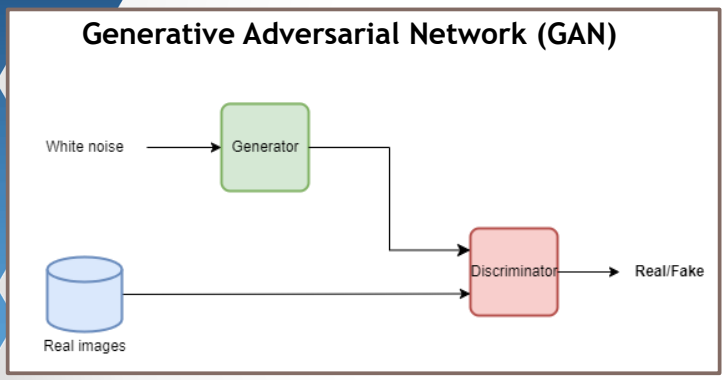
GANs

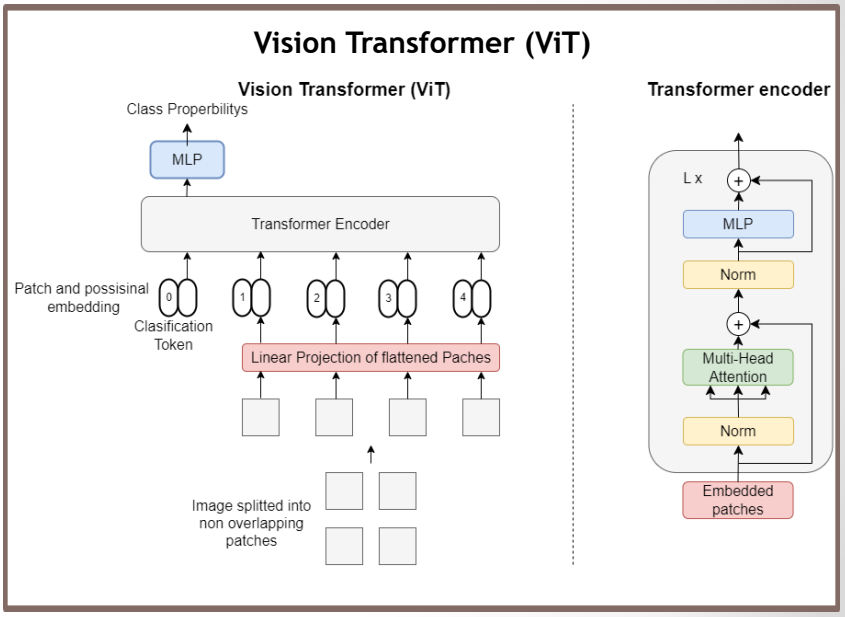


Generated data

University of Stavanger

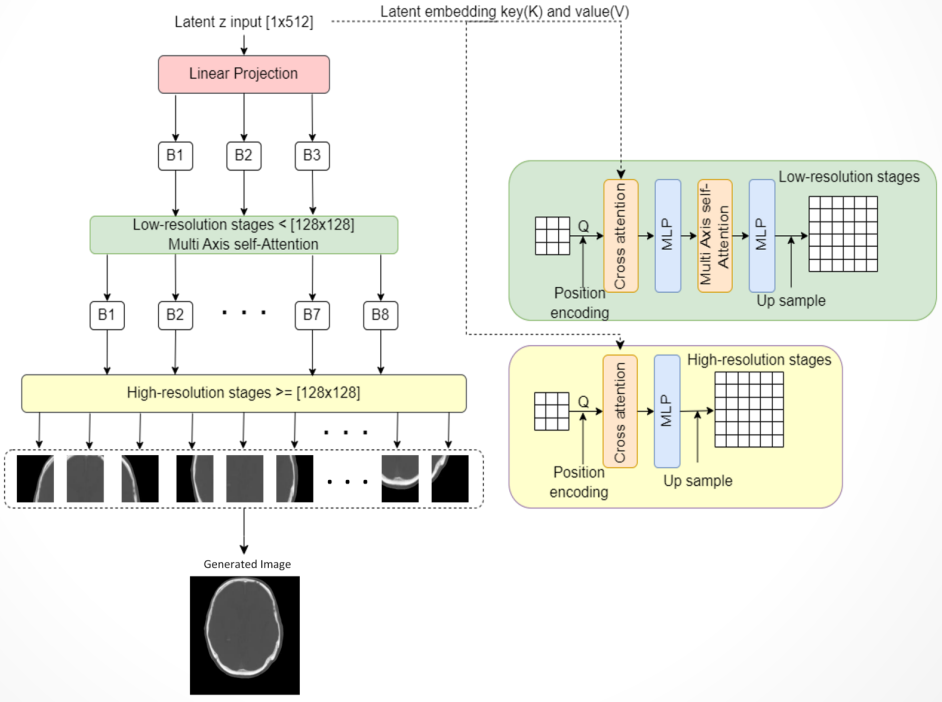
Approach





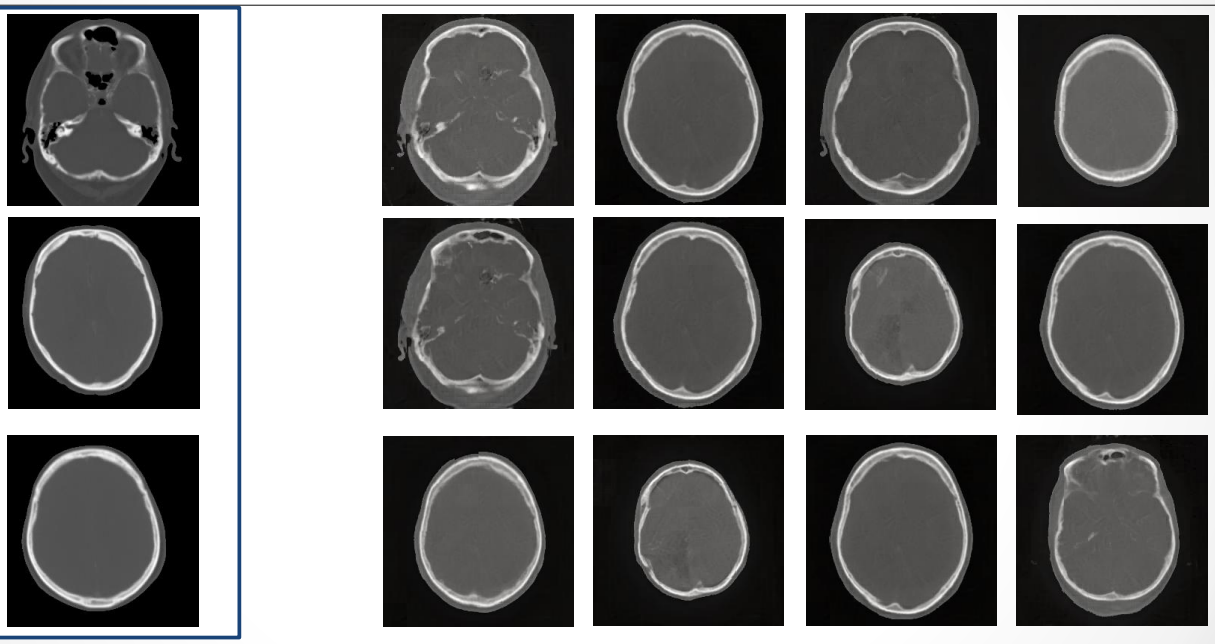
University of Stavanger

HiT-GAN architecture



University of Stavanger

Results and discussions



Real CTP data

Generated CTP data

University of Stavanger

References

Mode collapse?

- [1] Heit JJ, Wintermark M. “Perfusion computed tomography for the evaluation of acute ischemic stroke”. Stroke 2016; 47(4): 1153–1158.
- [2] Tomasetti L, Hansen S, Khanmohammadi M, Engan K, Høllesli LJ, Kurz KD, Kampffmeyer M. Self-supervised fewshot learning for ischemic stroke lesion segmentation, 2023.
- [3] Zhao L, Zhang Z, Chen T, Metaxas DN, Zhang H. “Improved transformer for high-resolution gans”, 2021.

University of Stavanger

Bibliography

[1] Jeremy J. Heit and Max Wintermark. Perfusion computed tomography for the evaluation of acute ischemic stroke. *Stroke*, 47(4):1153–1158, 2016. doi: 10.1161/STROKEAHA.116.011873.

[2] Luca Tomasetti, Stine Hansen, Mahdieh Khanmohammadi, Kjersti Engan, Liv Jorunn Høllesli, Kathinka Dæhli Kurz, and Michael Kampffmeyer. Self-supervised few-shot learning for ischemic stroke lesion segmentation, 2023.

[3] Murat Korkmaz. Synthesising training data with generative adversarial networks (GANs) in computed tomography perfusion. Master's thesis, University of Stavanger, Norway, 2021.

[4] Long Zhao, Zizhao Zhang, Ting Chen, Dimitris N. Metaxas, and Han Zhang. Improved transformer for high-resolution gans. 6 2021. URL http://arxiv.org/abs/2106.07631.

[5] CP Warlow. Epidemiology of stroke. *The Lancet*, 352:S1–S4, 1998. ISSN 0140-6736. doi: https://doi.org/10.1016/S0140-6736(98)90086-1. URL https://www.sciencedirect.com/science/article/pii/S0140673698900861. Stroke.

[6] Valery L Feigin et al. Global, regional, and national burden of stroke and its risk factors, 1990–2019: a systematic analysis for the global burden of disease study 2019. *The Lancet Neurology*, 20(10):795–820, 2021. ISSN 1474-4422. doi: https://doi.org/10.1016/S1474-4422(21)00252-0. URL https://www.sciencedirect.com/science/article/pii/S1474442221002520.

[7] World Health Organization et al. Optimizing brain health across the life course: Who position paper. 2022.

[8] Martin W Kurz, Johanna Maria Ospel, Kathinka Daehli Kurz, and Mayank Goyal. Improving stroke care in times of the covid-19 pandemic through simulation: practice your protocols! *Stroke*, 51(7):2273–2275, 2020.

[9] KD Kurz, G Ringstad, A Odland, R Advani, E Farbu, and MW Kurz. Radiological imaging in acute ischaemic stroke. *European journal of neurology*, 23:8–17, 2016.

[10] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. 11 2016.

[11] Vincent Dumoulin and Francesco Visin. A guide to convolution arithmetic for deep learning. 03 2016.

[12] Tommaso Cillotto, Alex Buoite Stella Giovanni Furlanis, Paola Caruso Carlo Lugnan, Marcello Naccarato Roberta Pozzi Mucelli, and Paolo Manganotti. Focusing on single ct perfusion quantitative maps: Percheron's artery stroke detection in an emergency setting. *Journal of the Neurological Sciences*, 429, 2021. doi: https://doi.org/10.1016/j.jns.2021.118742. URL https://www.sciencedirect.com/science/article/pii/S0022510X21014386.

[13] Bianca de Haan, Philipp Clas, Hendrik Juenger, Marko Wilke, and Hans-Otto Karnath. Fast semi-automated lesion demarcation in stroke. *NeuroImage Clin.*, 9: 69–74, July 2015.

[14] Anthony J Winder, Susanne Siemonsen, Fabian Flottmann, Götz Thomalla, Jens Fiehler, and Nils D Forkert. Technical considerations of multi-parametric tissue outcome prediction methods in acute ischemic stroke patients. *Sci. Rep.*, 9(1): 13208, September 2019.

[15] Salwa El Tawil and Keith W Muir. Thrombolysis and thrombectomy for acute ischaemic stroke. *Clin. Med.*, 17(2):161–165, April 2017.

[16] J. Claude Hemphill, Steven M. Greenberg, Craig S. Anderson, Kyra Becker, Bernard R. Bendok, Mary Cushman, Gordon L. Fung, Joshua N. Goldstein, R. Loch Macdonald, Pamela H. Mitchell, Phillip A. Scott, Magdy H. Selim, and Daniel Woo. Guidelines for the management of spontaneous intracerebral hemorrhage. *Stroke*, 46(7):2032–2060, 2015. doi: 10.1161/STR.0000000000000069. URL https://www.ahajournals.org/doi/abs/10.1161/STR.0000000000000069.

[17] Ajaya Kumar A. Unnithan, Joe M Das, and Parth Mehta. *Hemorrhagic Stroke*. StatPearls Publishing, Treasure Island (FL), 2022. URL http://europepmc.org/books/NBK559173.

[18] Seyedhossein Ojaghihaghighi, Samad Shams Vahdati, Akram Mikaeilpour, and Ali Ramouz. Comparison of neurological clinical manifestation in patients with hemorrhagic and ischemic stroke. *World journal of emergency medicine*, 8(1):34, 2017.

[19] David M Yousem, Robert D Zimmerman, Robert I Grossman, and Rohini Nadgir. *Neuroradiology: The Requisites E-Book*. Elsevier Health Sciences, 2010.

[20] Robert C Rennert, Arvin R Wali, Jeffrey A Steinberg, David R Santiago-Dieppa, Scott E Olson, J Scott Pannell, and Alexander A Khalessi. Epidemiology, natural history, and clinical presentation of large vessel ischemic stroke. *Neurosurgery*, 85 (Suppl 1):S4, 2019.

[21] Patti L. Hui C, Tadi P. Ischemic stroke, Updated 2022 Jun 2. URL https://www.ncbi.nlm.nih.gov/books/NBK499997/. Treasure Island (FL): StatPearls Publishing.

[22] Smith W. S., English J. D. Lev M. H., Chou M. Camargo E. C., Gonzalez G. Johnston S. C., Dillon W. P. Schaefer P. W., Koroshetz W. J., and Furie K. L. Significance of large vessel intracranial occlusion causing acute ischemic stroke and tia. *Stroke*, 40:3834–3840, 2009. doi: https://doi.org/10.1161/STROKEAHA.109.561787.

[23] Nikita Lakomkin, Mandip Dhamoon, Kirsten Carroll, Inder Paul Singh, Stanley Tuhrim, Joyce Lee, Johanna T Fifi, and J Mocco. Prevalence of large vessel occlusion in patients presenting with acute ischemic stroke: a 10-year systematic review of the literature. *Journal of neurointerventional surgery*, 11(3):241–245, 2019.

[24] Kathleen R. Tozer Fink and James R. Fink. 4 - principles of modern neuroimaging. In Richard G. Ellenbogen, Laligam N. Sekhar, Neil D. Kitchen, and Harley Brito da Silva, editors, *Principles of Neurological Surgery (Fourth Edition)*, pages 62–86.e2. Elsevier, Philadelphia, fourth edition edition, 2018. ISBN 978-0-323-43140-8. doi: https://doi.org/10.1016/B978-0-323-43140-8.00004-4. URL https://www.sciencedirect.com/science/article/pii/B9780323431408000044.

[25] Basemah Alshemali and Jugal Kalita. Improving the reliability of deep neural networks in nlp: A review. *Knowledge-Based Systems*, 191:105210, 2020. ISSN 0950-7051. doi: https://doi.org/10.1016/j.knosys.2019.105210. URL https://www.sciencedirect.com/science/article/pii/S0950705119305428.

[26] Boukaye Boubacar Traore, Bernard Kamsu-Foguem, and Fana Tangara. Deep convolution neural network for image recognition. *Ecological Informatics*, 48:257–268, 2018. ISSN 1574-9541. doi: https://doi.org/10.1016/j.ecoinf.2018.10.002. URL https://www.sciencedirect.com/science/article/pii/S1574954118302140.

[27] Li Deng, Geoffrey Hinton, and Brian Kingsbury. New types of deep neural network learning for speech recognition and related applications: an overview. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8599–8603, 2013. doi: 10.1109/ICASSP.2013.6639344.

[28] Y. Chauvin and D.E. Rumelhart. *Backpropagation: Theory, Architectures, and Applications*. Developments in Connectionist Theory Series. Taylor & Francis, 2013. ISBN 9781134775811. URL https://books.google.no/books?id=B71nu3LDpREC.

[29] Jason Brownlee. A gentle introduction to sigmoid function, from machine learning mastery, August 25, 2021. URL https://machinelearningmastery.com/a-gentle-introduction-to-sigmoid-function/. 24.05, 2023.

[30] Jason Brownlee. Softmax activation function with python, from machine learning mastery, October 19, 2020. URL https://machinelearningmastery.com/softmax-activation-function-with-python/. 24.05, 2023.

[31] Jason Brownlee. A gentle introduction to the rectified linear unit (relu), from machine learning mastery, January 9, 2019. URL https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/. 24.05, 2023.

[32] Vinod Nair and Geoffrey Hinton. Rectified linear units improve restricted boltzmann machines vinod nair. volume 27, pages 807–814, 06 2010.

[33] Andrew L. Maas. Rectifier nonlinearities improve neural network acoustic models. 2013.

[34] Dan Hendrycks and Kevin Gimpel. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. *CoRR*, abs/1606.08415, 2016. URL http://arxiv.org/abs/1606.08415.

[35] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. 6 2014. URL http://arxiv.org/abs/1406.2661.

[36] Karim Armanious, Chenming Jiang, Marc Fischer, Thomas Küstner, Tobias Hepp, Konstantin Nikolaou, Sergios Gatidis, and Bin Yang. Medgan: Medical image translation using gans. *Computerized Medical Imaging and Graphics*, 79:101684, 2020. ISSN 0895-6111. doi: https://doi.org/10.1016/j.compmedimag.2019.101684. URL https://www.sciencedirect.com/science/article/pii/S0895611119300990.

[37] Federico Di Mattia, Paolo Galeone, Michele De Simoni, and Emanuele Ghelfi. A survey on gans for anomaly detection. *CoRR*, abs/1906.11632, 2019. URL http://arxiv.org/abs/1906.11632.

[38] J.T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. Striving for simplicity: The all convolutional net. In *ICLR (workshop track)*, 2015. URL http://lmb.informatik.uni-freiburg.de/Publications/2015/DB15a.

[39] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 448–456. JMLR.org, 2015.

[40] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation, 2018.

[41] Jason Brownlee. How to identify and diagnose gan failure modes, 2019. URL https://machinelearningmastery.com/practical-guide-to-gan-failure-modes/. 24.03, 2023.

[42] Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *CoRR*, abs/1606.03498, 2016. URL http://arxiv.org/abs/1606.03498.

[43] Ali Borji. Pros and cons of gan evaluation measures, 2018.

[44] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. 6 2016. URL http://arxiv.org/abs/1606.03498.

[45] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. volume 2016-December, pages 2818–2826. IEEE Computer Society, 12 2016. ISBN 9781467388504. doi: 10.1109/CVPR.2016.308.

[46] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. 6 2017. URL http://arxiv.org/abs/1706.08500.

[47] Jason Brownlee. How to implement the frechet inception distance (fid) for evaluating gans, August 30, 2019. URL https://machinelearningmastery.com/how-to-implement-the-frechet-inception-distance-fid-from-scratch/. 19.04, 2023.

[48] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. volume 2017-December, 2017.

[49] Alex Sherstinsky. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *Physica D: Nonlinear Phenomena*, 404:132306, 2020. ISSN 0167-2789. doi: https://doi.org/10.1016/j.physd.2019.132306. URL https://www.sciencedirect.com/science/article/pii/S0167278919305974.

[50] Chenguang Wang, Mu Li, and Alexander J. Smola. Language models with transformers, 2019.

[51] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *The International Conference on Learning Representations*, 2021.

[52] Nanxin Chen, Shinji Watanabe, Jesús Villalba, Piotr Żelasko, and Najim Dehak. Non-autoregressive transformer for speech recognition. *IEEE Signal Processing Letters*, 28:121–125, 2021. doi: 10.1109/LSP.2020.3044547.

[53] Witold Wydmanski. What's the difference between self-attention and attention in transformer architecture?, from medium.com, Dec 3, 2022. URL https://medium.com/mlearning-ai/whats-the-difference-between-self-attention-and-attention-in-transformer-architecture-3780404382f3. 26.05, 2023.

[54] Ketan Doshi. Transformers explained visually — not just how, but why they work so well, Jun 2, 2021. URL https://towardsdatascience.com/transformers-explained-visually-not-just-how-but-why-they-work-so-well-d840bd61a9d3. 08.02, 2023.

[55] Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *CoRR*, abs/1607.06450, 2016. URL http://arxiv.org/abs/1607.06450.

[56] G. Bebis and M. Georgiopoulos. Feed-forward neural networks. *IEEE Potentials*, 13(4):27–31, 1994. doi: 10.1109/45.329294.

[57] Ketan Doshi. Transformers explained visually (part 2): How it works, step-by-step, Jan 2, 2021. URL https://towardsdatascience.com/transformers-explained-visually-part-2-how-it-works-step-by-step-b49fa4a64f34. 07.02, 2023.

[58] Ketan Doshi. Transformers explained visually (part 1): Overview of functionality, Dec 13, 2020. URL https://towardsdatascience.com/transformers-explained-visually-part-1-overview-of-functionality-95a6dd460452. 07.02, 2023.

[59] Saul Fuster, Farbod Khoraminia, Umay Kiraz, Neel Kanwal, Vebjørn Kvikstad, Trygve Eftestøl, Tahlita C.M. Zuiverloon, Emiel A.M. Janssen, and Kjersti Engan. Invasive cancerous area detection in non-muscle invasive bladder cancer whole slide images. In *2022 IEEE 14th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP)*, pages 1–5, 2022. doi: 10.1109/IVMSP54334.2022.9816352.

[60] Neel Kanwal, Roger Amundsen, Helga Hardardottir, Emiel AM Janssen, and Kjersti Engan. Detection and localization of melanoma skin cancer in histopathological whole slide images. *arXiv preprint arXiv:2302.03014*, 2023.

[61] Luca Tomasetti. Segmentation of infarcted regions in Perfusion CT images by 3D deep learning. Master's thesis, University of Stavanger, Norway, 2019. URL http://hdl.handle.net/11250/2620505.

[62] Luca Tomasetti, Mahdieh Khanmohammadi, Kjersti Engan, Liv Jorunn Høllesli, and Kathinka Dæhli Kurz. Multi-input segmentation of damaged brain in acute ischemic stroke patients using slow fusion with skip connection. *arXiv preprint arXiv:2203.10039*, 2022.

[63] Øyvind Meinich-Bache, Kjersti Engan, Ivar Austvoll, Trygve Eftestøl, Helge Myklebust, Ladislaus Blacy Yarrot, Hussein Kidanto, and Hege Ersdal. Object detection during newborn resuscitation activities. *IEEE journal of biomedical and health informatics*, 24(3):796–803, 2019.

[64] Øyvind Meinich-Bache, Simon Lennart Austnes, Kjersti Engan, Ivar Austvoll, Trygve Eftestøl, Helge Myklebust, Simeon Kusulla, Hussein Kidanto, and Hege Ersdal. Activity recognition from newborn resuscitation videos. *IEEE journal of biomedical and health informatics*, 24(11):3258–3267, 2020.

[65] Kwonjoon Lee, Huiwen Chang, Lu Jiang, Han Zhang, Zhuowen Tu, and Ce Liu. Vitgan: Training gans with vision transformers. 7 2021. URL http://arxiv.org/abs/2107.04589.

[66] Hyunjik Kim, George Papamakarios, and Andriy Mnih. The lipschitz constant of self-attention, 2021.

[67] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. 12 2019. URL http://arxiv.org/abs/1912.04958.

[68] Zizhao Zhang, Han Zhang, Long Zhao, Ting Chen, Sercan O. Arik, and Tomas Pfister. Nested hierarchical transformer: Towards accurate, data-efficient and interpretable visual understanding. 5 2021. URL http://arxiv.org/abs/2105.12723.

[69] W. Shi, J. Caballero, F. Huszar, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1874–1883, Los Alamitos, CA, USA, jun 2016. IEEE Computer Society. doi: 10.1109/CVPR.2016.207. URL https://doi.ieeecomputersociety.org/10.1109/CVPR.2016.207.

[70] Remote sensing single-image resolution improvement using a deep gradient-aware network with image-specific enhancement - scientific figure on researchgate. URL https://www.researchgate.net/figure/The-pixel-shuffle-layer-transforms-feature-maps-from-the-LR-domain-to-the-HR-image_fig3_339531308. accessed 29 May 2023.

[71] Kevin Roth, Aurelien Lucchi, Sebastian Nowozin, and Thomas Hofmann. Stabilizing training of generative adversarial networks through regularization. 05 2017.

[72] Han Zhang, Zizhao Zhang, Augustus Odena, and Honglak Lee. Consistency regularization for generative adversarial networks. 10 2019. URL http://arxiv.org/abs/1910.12027.

[73] Shengyu Zhao, Zhijian Liu, Ji Lin, Jun-Yan Zhu, and Song Han. Differentiable augmentation for data-efficient gan training. 6 2020. URL http://arxiv.org/abs/2006.10738.

[74] Jason Brownlee. A gentle introduction to generative adversarial network loss functions, September 2, 2019. URL https://machinelearningmastery.com/generative-adversarial-network-loss-functions/. 19.05, 2023.

[75] Richard Zhang. Making convolutional networks shift-invariant again. 4 2019. URL http://arxiv.org/abs/1904.11486.

[76] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. 3 2021. URL http://arxiv.org/abs/2103.14030.

[77] Ryan Steed and Aylin Caliskan. Image representations learned with unsupervised pre-training contain human-like biases. pages 701–713. Association for Computing Machinery, Inc, 3 2021. ISBN 9781450383097. doi: 10.1145/3442188.3445932.

[78] Ting Chen, Mario Lucic, Neil Houlsby, and Sylvain Gelly. On self modulation for generative adversarial networks. 10 2018. URL http://arxiv.org/abs/1810.01365.

[79] Salome Kazeminia, Christoph Baur, Arjan Kuijper, Bram van Ginneken, Nassir Navab, Shadi Albarqouni, and Anirban Mukhopadhyay. Gans for medical image analysis. *Artificial Intelligence in Medicine*, 109:101938, 2020. ISSN 0933-3657. doi: https://doi.org/10.1016/j.artmed.2020.101938. URL https://www.sciencedirect.com/science/article/pii/S0933365719311510.

[80] Youssef Skandarani, Pierre-Marc Jodoin, and Alain Lalande. Gans for medical image synthesis: An empirical study. *Journal of Imaging*, 9(3):69, 2023.

[81] Fahad Shamshad, Salman Khan, Syed Waqas Zamir, Muhammad Haris Khan, Munawar Hayat, Fahad Shahbaz Khan, and Huazhu Fu. Transformers in medical imaging: A survey. *Medical Image Analysis*, page 102802, 2023. ISSN 1361-8415. doi: https://doi.org/10.1016/j.media.2023.102802. URL https://www.sciencedirect.com/science/article/pii/S1361841523000634.

[82] Hu Cao, Yueyue Wang, Joy Chen, Dongsheng Jiang, Xiaopeng Zhang, Qi Tian, and Manning Wang. Swin-unet: Unet-like pure transformer for medical image segmentation. 5 2021. URL http://arxiv.org/abs/2105.05537.

[83] Lucas De Vries, Bart Emmer, Charles Majoie, C B Majoie@amsterdamumc Nl, Henk Marquering, H A Marquering@amsterdamumc Nl, and Efstratios Gavves. Transformers for ischemic stroke infarct core segmentation from spatio-temporal ct perfusion scans. *Medical Imaging with Deep Learning-Under Review*, 2021.

[84] Jeya Maria Jose Valanarasu, Poojan Oza, Ilker Hacihaliloglu, and Vishal M. Patel. Medical transformer: Gated axial-attention for medical image segmentation. volume 12901 LNCS, pages 36–46. Springer Science and Business Media Deutschland GmbH, 2021. ISBN 9783030871925. doi: 10.1007/978-3-030-87193-2_4.

[85] Xiaohong Gao, Yu Qian, and Alice Gao. Covid-vit: Classification of covid-19 from ct chest images based on vision transformer models. URL https://github.com/xiaohong1/COVID-ViT.

[86] Rishi Raj, Jimson Mathew, Santhosh Kumar Kannath, and Jeny Rajan. Strokevit with automl for brain stroke classification. *Engineering Applications of Artificial Intelligence*, 119, 3 2023. ISSN 09521976. doi: 10.1016/j.engappai.2022.105772.

[87] Yin Dai, Yifan Gao, and Fayu Liu. Transmed: Transformers advance multimodal medical image classification. *Diagnostics*, 11, 8 2021. ISSN 20754418. doi: 10.3390/diagnostics11081384.

[88] Meng Li, Chaoyi Li, Peter Hobson, Tony Jennings, and Brian C Lovell. Medvitgan: End-to-end conditional gan for histopathology image augmentation with vision transformers. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 4406–4413. IEEE, 2022.

[89] Yifan Jiang, Shiyu Chang, and Zhangyang Wang. Transgan: Two pure transformers can make one strong gan, and that can scale up. *Advances in Neural Information Processing Systems*, 34:14745–14758, 2021.

[90] Sigurd Myklebust. Cerebral vessel segmentation in contrast ct images. Master's thesis, University of Stavanger, Norway, 2018.

[91] Luca Tomasetti, Kjersti Engan, Liv Jorunn Høllesli, Kathinka Dæhli Kurz, and Mahdieh Khanmohammadi. Exploiting 4d ct perfusion for segmenting infarcted areas in patients with suspected acute ischemic stroke. *ArXiv*, abs/2303.08757, 2023.

[92] Luca Tomasetti, Liv Jorunn Høllesli, Kjersti Engan, Kathinka Dæhli Kurz, Martin Wilhelm Kurz, and Mahdieh Khanmohammadi. Machine learning algorithms versus thresholding to segment ischemic regions in patients with acute ischemic stroke. *IEEE Journal of Biomedical and Health Informatics*, 26(2):660–672, 2021.

[93] Mohamed Najm, Hulin Kuang, Alyssa Federico, Uzair Jogiat, Mayank Goyal, Michael D Hill, Andrew Demchuk, Bijoy K Menon, and Wu Qiu. Automated brain extraction from head ct and cta images using convex optimization with shape propagation. *Computer Methods and Programs in Biomedicine*, 176:1–8, 2019.

[94] Hit-gan official tensorflow implementation. URL https://github.com/google-research/hit-gan. accessed 2 June 2023.

[95] Charlie Frogner, Chiyuan Zhang, Hossein Mobahi, Mauricio Araya, and Tomaso A Poggio. Learning with a wasserstein loss. *Advances in neural information processing systems*, 28, 2015.

[96] Claudio Gentile and Manfred KK Warmuth. Linear hinge loss and average margin. *Advances in neural information processing systems*, 11, 1998.

[97] Jason Brownlee. How to implement the inception score (is) for evaluating gans, October 11, 2019. URL https://machinelearningmastery.com/how-to-implement-the-inception-score-from-scratch-for-evaluating-generated-images/. 03.05, 2023.

[98] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

[99] Rob J Hyndman. Moving averages., 2011.

[100] Functional api. URL https://github.com/keras-team/keras-io/blob/master/guides/functional_api.py. accessed 4 June 2023.