Franz Thaler, BSc

Sparse-View Computed Tomography Reconstruction Using Wasserstein Generative Adversarial Networks

MASTER'S THESIS

to achieve the university degree of
Diplom-Ingenieur

Master's degree programme
Software Engineering and Management

submitted to

Graz University of Technology

Supervisor

Univ.-Prof. Dipl.-Ing. Dr.techn. Horst Bischof
Institute of Computer Graphics and Vision, Graz University of Technology

Advisor

Dipl.-Ing. Dr.techn. Darko Štern
Ludwig Boltzmann Institute for Clinical Forensic Imaging, Graz

Graz, Austria, March 2019

Abstract

While Computed Tomography (CT) is a widely used and well established technique to non-invasively visualize the patient's interior body structure, it also exposes the patient to ionizing radiation and increases the risk to develop cancer. Directly reducing the amount of ionizing radiation exposure and violating the Nyquist-Shannon sampling theorem degrades the quality of the reconstructed CT image by heavily introducing artifacts. Different approaches aiming to improve the image quality from undersampled data have been introduced of which many recent approaches rely on Convolutional Neural Networks (CNNs). In this thesis we propose a sparse-view CT reconstruction method from a reduced set of equidistant projection views around the axial plane of the patient. To allow data augmentation and experimentation at will, we simulate the necessary projection data by generating it from 10 already reconstructed normal-dose CT images which we separated into a training and test set. Our experiments include the utilization of three dimensional (3D) CT volumes respectively two dimensional (2D) CT slices used to train individual CNNs optimized on $\mathcal{L}_1$ loss. We conducted additional experiments on the 2D data where a combined loss function is used that consists of a content $\mathcal{L}_1$ and an adversarial $\mathcal{L}_{wGAN}$ loss coming from a Generative Adversarial Network (GAN). We show that CNNs represent a viable option to reduce the amount of ionizing radiation exposure to the patient while still achieving good reconstruction results compared to the well established Filtered Backprojection (FBP) method and also, that GAN based methods can be used to further optimize the visual quality of the reconstructed images. However, the results generated by these CNN based methods have to be treated with care especially when huge undersampling rates are used, since the correspondence to the patient is gradually decreased which can lead to the introduction of artifacts that look similar to anatomical structures. As such, while the use in diagnostic clinical practice remains questionable, we see potential applications in cases where reducing the amount of ionizing radiation exposure to the patient is favorable over image quality, e.g. during minimally invasive and image guided surgeries.

Kurzfassung

Während Computertomographie (CT) ein weit verbreitetes und etabliertes Verfahren darstellt, um die inneren Körperstrukturen eines/einer Patienten/Patientin zu visualisieren, so setzt sie den/die Patienten/Patientin auch einer Belastung durch ionisierende Strahlung aus, die das Risiko erhöht Krebs zu entwickeln. Eine direkte Reduzierung der ionisierenden Strahlenbelastung und eine Verletzung des Nyquist-Shannon-Abtasttheorems führen zu einer verringerten Qualität der rekonstruierten CT Bilder aufgrund einer vermehrten Einführung von Artefakten. Verschiedene Ansätze existieren, die darauf abzielen die Bildqualität von unterabgetasteten Daten zu verbessern, von welchen sich viele kürzlich publizierte Arbeiten auf Convolutional Neural Networks (CNNs) verlassen. In dieser Diplomarbeit schlagen wir eine CT Rekonstruktionsmethode von einer verringerten Anzahl von gleichabständigen Projektionsbildern des/der Patienten/Patientin von verschiedenen Winkeln auf der Transversalebene vor. Um eine weitreichende Datenaugmentierung sowie die Durchführung beliebiger Experimente zu ermöglichen, simulieren wir die notwendigen Projektionsdaten, indem wir diese von 10 bereits rekonstruierten normal dosierten CT Bildern simulieren, welche wir in ein Trainings- und Testset aufteilen. In unseren Experimenten nutzen wir dreidimensionale (3D) CT Volumen bzw. zweidimensionale (2D) CT Scheiben, die jeweils dazu genutzt werden, individuelle CNNs zu trainieren, welche mithilfe der $\mathcal{L}_1$ Verlustfunktion optimiert werden. Des Weiteren haben wir zusätzliche Experimente auf den 2D Daten durchgeführt, bei denen wir eine kombinierte Verlustfunktion verwenden, die die $\mathcal{L}_1$ Verlustfunktion mit einer von einem Generative Adversarial Network (GAN) stammenden $\mathcal{L}_{wGAN}$ Verlustfunktion kombiniert. Wir zeigen, dass CNNs eine Möglichkeit darstellen, um die Belastung durch ionisierende Strahlung des/der Patienten/Patientin zu reduzieren, während gleichzeitig bessere Rekonstruktionsergebnisse erzielt werden als durch die gefilterte Rückprojektion (FBP) und weiters, dass GAN basierte Methoden genutzt werden können, um eine weitere

Verbesserung der visuellen Qualität der rekonstruierten Bilder zu erreichen. Jedoch sind Ergebnisse, die mithilfe von CNN basierten Methoden generiert werden mit Vorsicht zu behandeln, dies gilt insbesondere für den Fall einer hohen Unterbeabtastung. Da die Übereinstimmung des rekonstruierten Bildes mit dem/der Patienten/Patientin graduell reduziert wird, kann dies zu der Einführung von Artefakten führen, die echten anatomischen Strukturen ähneln. Während die Anwendung von CNN basierten Methoden zur Diagnose in der klinischen Praxis fragwürdig bleibt, sehen wir potenzielle Anwendungen in Fällen, in denen eine Reduzierung der ionisierenden Strahlenbelastung des/der Patienten/Patientin gegenüber einer optimalen Bildqualität bevorzugt wird, ein Beispiel hierfür stellen minimal invasive und bildgestützte chirurgische Eingriffe dar.

Affidavit

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used.

The text document uploaded to TUGRAZonline is identical to the present master's thesis dissertation.

Date	Place	Signature

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

Das in TUGRAZonline hochgeladene Textdokument ist mit der vorliegenden Masterarbeit identisch.

Ort	Datum	Unterschrift

Contents

1 Overview **1**

2 Medical Image Reconstruction **5**

2.1 Imaging Modalities in Medicine 6
  2.1.1 Motivation 7
2.2 Inverse Problems 7
  2.2.1 Image Processing and Signal Processing 7
  2.2.2 Undersampling 8
  2.2.3 Nyquist-Shannon Sampling Theorem 8
  2.2.4 Compressed Sensing 9
  2.2.5 Computed Tomography as an Inverse Problem 9
2.3 Computed Tomography 10
  2.3.1 Schematics of a Computed Tomography Scanner 10
  2.3.2 Types of Computed Tomography Scanners 11
2.4 Image Acquisition in Computed Tomography 12
  2.4.1 Ionizing Radiation 13
  2.4.2 Fundamental Photon Attenuation Law 13
  2.4.3 Projection View Generation using Line Integrals 14
  2.4.4 Hounsfield Unit Scale 16
  2.4.5 Sinogram Generation 16
2.5 Image Reconstruction in Computed Tomography 20
  2.5.1 Backprojection 20
  2.5.2 Projection-Slice Theorem 21
  2.5.3 Filtered Backprojection 23
2.6 Limitations of Classical Reconstruction Approaches 24

3 Machine Learning **27**
3.1 Supervised Learning . . . 28
3.1.1 Discriminative Models . . . 28
3.1.2 Generative Models . . . 29
3.2 Deep Learning . . . 29
3.2.1 Artificial Neurons . . . 29
3.2.2 Feedforward Neural Networks . . . 31
3.3 Optimization . . . 32
3.3.1 Loss Function . . . 32
3.3.2 Optimizer . . . 33
3.4 Convolutional Neural Networks . . . 35
3.4.1 Convolution . . . 35
3.4.2 Pooling . . . 36
3.4.3 Activation Functions . . . 36
3.5 Architectures of Neural Networks . . . 37
3.5.1 Encoder . . . 38
3.5.1.1 Classification Neural Network . . . 38
3.5.2 Decoder . . . 38
3.5.2.1 Image-to-Image Neural Network . . . 39
3.6 Generative Adversarial Networks . . . 40
3.6.1 Wasserstein Generative Adversarial Networks . . . 43
3.6.2 Wasserstein Generative Adversarial Networks with Gradient Penalty 44
3.6.3 Extending Generative Adversarial Networks with a Prior . . . 45
3.7 Training Neural Networks . . . 46
3.7.1 Underfitting and Overfitting . . . 46
3.7.2 Early Stopping . . . 47
3.7.3 Data Augmentation . . . 47

4 Related Work **49**
4.1 Different Strategies to Apply Deep Learning . . . 50
4.1.1 Sinogram-to-Sinogram Reconstruction . . . 50
4.1.2 Image-to-Image Reconstruction . . . 51
4.1.3 Sinogram-to-Sinogram Reconstruction and then Image-to-Image Reconstruction . . . 52
4.1.4 Sinogram-to-Image Reconstruction . . . 52
4.2 Tube Current Reduction based Approaches . . . 53
4.2.1 Tube Current Reduced Approaches . . . 54
4.2.2 Approaches Using Generative Adversarial Networks . . . 54
4.3 Beam Blocking based Approaches . . . 55
4.3.1 Observations on Physical Properties of Beam Blockers . . . 56
4.3.2 Beam Blocking in Cone-Beam Computed Tomography . . . 56

4.3.3 Beam Blocking in Fan-Beam Computed Tomography 56
4.4 Sparse-View based Approaches 57
4.4.1 Sparse Sinogram-to-Sinogram Reconstruction based Approaches . . 57
4.4.2 Image-to-Image Reconstruction based Approaches 58
4.4.3 Sparse Sinogram-to-Image Learning based Approaches 59
4.4.4 Limited Angle Computed Tomography 59
4.4.5 Approaches Using Generative Adversarial Networks 60
4.5 Accelerated Magnetic Resonance Imaging 60
4.5.1 Typical Accelerated Magnetic Resonance Imaging Approaches . . . 60
4.5.2 Generative Adversarial Networks in Magnetic Resonance Imaging . . 61

5 Method 63
5.1 Preprocessing of the Data 64
5.1.1 2D Computed Tomography Slice Extraction for our 2D Pipeline . . 64
5.1.2 3D Vertebra Extraction for our 3D Pipeline 64
5.2 On-the-fly Projection Data Generation 66
5.2.1 Augmentation 67
5.2.2 Masking 67
5.2.3 Forward Projection 68
5.2.4 Back Projection 70
5.3 Learned Computed Tomography Reconstruction 70
5.3.1 Generator Network Architecture 71
5.3.2 Discriminator Network Architecture 73
5.3.3 Loss Function 73
5.4 Experimental Setup 74
5.4.1 Material 74
5.4.2 Setup Parameters for All Experiments 75
5.4.3 Setup Parameters Specific to 3D Experiments 75
5.4.4 Setup Parameters Specific to 2D Experiments 75

6 Results 77
6.1 Evaluation of 3D Experiments 77
6.1.1 Quantitative Evaluation 77
6.1.2 Qualitative Evaluation 79
6.1.2.1 Overview of a Selected Volume Used for Evaluation 79
6.1.2.2 Varying the Number of Projection Views for a Selected Volume 80
6.1.2.3 Evaluation of Additional Slices 80
6.2 Evaluation of 2D Experiments 83
6.2.1 Quantitative Evaluation 83
6.2.2 Qualitative Evaluation 84

6.2.2.1 Overview of a Selected Slice Used for Evaluation 86
6.2.2.2 Varying the Number of Projection Views for a Selected Slice 86
6.2.2.3 Varying the Weight λ between Losses for a Selected Slice . 91
6.2.2.4 Evaluation of Additional Slices 92
6.2.2.5 Evaluation of the Coronal and Sagittal View 93
6.2.2.6 Evaluation of a Site Unseen During Training 95

7 Discussion **99**
7.1 Comparison to Related Work 99
7.1.1 Strategies to Apply Deep Learning 99
7.1.2 Types of Approaches to Reduce Ionizing Radiation 100
7.2 Discussion of Our Method 101
7.2.1 3D Experiments 101
7.2.2 2D Experiments 102
7.3 Future Work 104

8 Conclusion **105**

A List of Acronyms **107**

B List of Publications **109**
B.1 2018 109
B.2 2018 110

Bibliography **111**

List of Figures

2.1 A radiography and a Computed Tomography image. 6
2.2 A signal that was undersampled and reconstructed incorrectly. 9
2.3 A Computed Tomography scanner. 10
2.4 Different geometries of Computed Tomography scanners. 11
2.5 The image acquisition process in Computed Tomography. 13
2.6 The attenuation process of photons. 14
2.7 The correlation between the photon energy and the linear attenuation coefficient. 15
2.8 Projection view generation using line integrals. 16
2.9 Some Hounsfield Unit numbers related to medical imaging. 17
2.10 Imaging planes of a body in medicine. 17
2.11 The forward projection procedure. 18
2.12 A line integral on a 2D plane using a δ function. 19
2.13 The sinogram generation procedure. 19
2.14 The backprojection procedure. 21
2.15 A sequential visual demonstration of the backprojection procedure. 21
2.16 The projection-slice theorem. 22
2.17 A sequential visual demonstration of the Filtered Backprojection method. 23
2.18 Different reconstruction results using a reduced number of projection views generated by the Filtered Backprojection method. 25

3.1 Supervised learning. 28
3.2 A biological neuron. 30
3.3 An artificial neuron. 30
3.4 A Feedforward Neural Network. 31
3.5 A convolution kernel. 36

3.6 A pooling layer. 36
3.7 Important activation functions. 37
3.8 An encoder neural network. 38
3.9 A classification neural network. 39
3.10 A decoder neural network. 39
3.11 The U-Net. 40
3.12 Schematic of a Generative Adversarial Network. 41
3.13 Schematic of an extension to Generative Adversarial Networks. 45
3.14 Early stopping. 47
3.15 Different data augmentation operations. 48

4.1 Sinogram-to-sinogram reconstruction procedure. 51
4.2 Image-to-image reconstruction procedure. 51
4.3 Sinogram-to-sinogram reconstruction procedure followed by image-to-image reconstruction procedure. 52
4.4 Sinogram-to-image reconstruction procedure. 53
4.5 Schematic of tube current based approaches. 53
4.6 Schematic of beam blocking based approaches. 55
4.7 Schematic of sparse-view based approaches. 57

5.1 The data extraction procedures. 65
5.2 The analogy between 3D Computed Tomography images and 2D Computed Tomography slices. 66
5.3 Difficulties encountered when generating projection data. 68
5.4 An original image, a mask and a masked image. 68
5.5 Discrete forward projection generation. 69
5.6 Discrete back projection procedure. 70
5.7 Projection image generation from a target image. 71
5.8 Schematic of our 3D method. 72
5.9 Schematic of our two 2D methods. 72

6.1 Quantitative results evaluating the Mean Absolute Error of our 3D method compared to the Filtered Backprojection method. 78
6.2 Important structures considered in the qualitative evaluation of our 3D method. 79
6.3 Qualtitative results of our 3D method and the Filtered Backprojection method for a different number of views. 81
6.4 Qualtitative results of our 3D method showing additional slices for eight and 30 views. 82
6.5 Quantitative results evaluating the Mean Absolute Error of our 2D methods compared to the Filtered Backprojection method. 84

6.6 Quantitative results evaluating the Structural Similarity Index Measure of our 2D methods compared to the Filtered Backprojection method. 85
6.7 Important structures considered in the qualitative evaluation of our 2D method. 86
6.8 Qualtitative results of our 2D methods and the Filtered Backprojection method for one and two views. 88
6.9 Qualtitative results of our 2D methods and the Filtered Backprojection method for four and six views. 88
6.10 Qualtitative results of our 2D methods and the Filtered Backprojection method for eight and 15 views. 90
6.11 Qualtitative results of our 2D methods and the Filtered Backprojection method for 30 and 60 views. 90
6.12 Qualtitative results of the Filtered Backprojection method for 120, 180 and 360 views. 91
6.13 Qualtitative results of our 2D-$\mathcal{L}_1$-$\mathcal{L}_{wGAN}$ method generated using different weights between the loss functions. 92
6.14 Qualtitative results of our 2D methods showing additional slices for eight and 30 views. 94
6.15 Qualtitative results of our 2D methods when showing the coronal and sagittal view of the stacked 2D results using eight views. 95
6.16 Qualtitative results of our 2D methods when showing the coronal and sagittal view of the stacked 2D results using 30 views. 95
6.17 Qualtitative results of our 2D methods when evaluating a site unseen during traing using a different number of views. 96
6.18 Qualtitative results of our 2D methods when evaluating a site unseen during traing using a different number of views. 97

List of Tables

6.1 Quantitative results evaluating the Mean Absolute Error of our 3D method compared to the Filtered Backprojection method. 78
6.2 Quantitative results evaluating the Mean Absolute Error of our 2D methods compared to the Filtered Backprojection method. 83
6.3 Quantitative results evaluating the Structural Similarity Index Measure of our 2D methods compared to the Filtered Backprojection method. 85

1

Overview

Computed Tomography (CT) is a well established and widely used technique to visualize the interior body structure of a patient non-invasively. Radiologists and physicians use the information they gain from *CT* images to treat patients by diagnosing injuries and diseases as well as monitoring their progress. *CT* imaging requires a set of X-ray projection views that are acquired from different angles on the axial plane of the patient and then used for *CT* image reconstruction. However, since *CT* imaging is based on X-ray technology, the patient is exposed to ionizing radiation during image acquisition and potentially harmed in the long term by introducing cancer [11]. Directly reducing the amount of ionizing radiation exposed to the patient leads to undersampling, i.e. a violation of the Nyquist-Shannon sampling theorem and consequently to a degraded image quality that increasingly suffers from artifacts especially when classical reconstruction techniques like the Filtered Backprojection (FBP) method are used for image reconstruction. To preserve a better image quality, many recent contributions to the field of *CT* image reconstruction from undersampled data rely on machine learning and especially Convolutional Neural Networks (CNNs), where knowledge about the data distribution learned during training is exploited to compensate the reduced amount of projection data available for reconstruction. Additionally, literature provides different approaches to acquire undersampled projection data in the first place, which we separated into three groups, namely tube current reduction based, beam blocking based and sparse-view based approaches. Tube current reduction based approaches reduce the tube current that leads to a lower number of emitted X-rays utilized for image acquisition, while beam blocking based methods reduce the amount of radiation by using physical beam blockers to partially shield the patient from X-rays. Lastly, sparse-view based methods reduce the number of projection views that are acquired and used for *CT* image reconstruction.

In this thesis we propose a *CNN* based sparse-view *CT* reconstruction approach that is motivated by reducing the amount of ionizing radiation exposure to the patient, where we aim to improve the quality of reconstructed *CT* images from undersampled projection data. Many recent learning based approaches utilize already reconstructed low-dose

CT images that have been reconstructed using classical techniques like the *FBP* method and learn to improve the quality of these images by removing the undersampling artifacts introduced by the utilized classical reconstruction technique. In contrast to that, our sparse-view *CT* reconstruction method does not rely on any classical reconstruction technique and instead replaces it, by directly optimizing a *CNN* to learn *CT* image reconstruction from a reduced number of projection views. As such, the *CNN* used in our approach directly receives a set of equidistant projection views acquired from the axial plane of the patient as an input, while the corresponding normal-dose *CT* image is used as the target to optimize the *CNN*. Since most publicly available data sets in this field only provide already reconstructed *CT* images without the original projection data used for reconstruction, we decided to simulate the projection data from the *CT* images. The simulation of the projection data results in two important advantages: First, simulation allowed us to augment the *CT* images before generating the projection data at will, which would corrupt the correspondence to the original projection data in most cases and second, it also made it possible to freely experiment with the number of projection views as well as with the angles from which they are acquired. As such, projection image simulation enabled us to proof our conecpt and focus on the feasibility of our method as well as on the various experiments we wanted to conduct to get a feeling of the performance of our method.

We implemented our method as a two dimensional (2D) and a three dimensional (3D) pipeline, where the *2D* pipeline learns to independently reconstruct *2D* *CT* slices, while the *3D* pipeline directly reconstructs the whole *CT* volume at once. Due to the vast execution time and huge memory requirements when training the *CNN* on the *3D* data, it was necessary to use a rather small image size to sufficiently conduct the *3D* experiments, whereas the lower dimensionality of our *2D* pipeline allowed us to use a larger image size and additional experiments. We optimized our *CNNs* by minimizing the $\mathcal{L}_1$ loss function, however, we also experimented with a combined loss function on our *2D* data that consists of a content $\mathcal{L}_1$ loss and an adversarial $\mathcal{L}_{wGAN}$ loss that comes from a Generative Adversarial Network (GAN) [26]. The intuition behind this combined loss function is, that while a content $\mathcal{L}_1$ loss promotes consistency in anatomy, an adversarial $\mathcal{L}_{wGAN}$ loss promotes sharper looking results. The results generated by our approach are evaluated quantitatively and qualitatively by comparing them to the results of the well known *FBP* method which we used as a baseline.

We show that the results generated by our approach are significantly better than the results of the *FBP* method when undersampled projection data is utilized for *CT* image reconstruction. While the *FBP* method increasingly suffers from the introduction of streaking artifacts when decreasing the number of projection views, our approach is able to result in images that convey more information. Comparing the results optimized on $\mathcal{L}_1$ loss to the results generated by our *GAN* based method *2D*-$\mathcal{L}_1$-$\mathcal{L}_{wGAN}$ shows, that the $\mathcal{L}_1$-only results look increasingly blurry, while *2D*-$\mathcal{L}_1$-$\mathcal{L}_{wGAN}$ is able to achieve sharper reconstruction results that look more realistic, but are not necessarily anatomical consistent to the patient. This observation leads to the conclusion that, while the results generated by our *2D*-$\mathcal{L}_1$-$\mathcal{L}_{wGAN}$ method look visually better compared to the results of our *2D*-$\mathcal{L}_1$-only method, they have to be treated with caution when heavily undersampled projection data is used for image reconstruction. We show that tweaking the weight hyperparameter of the combined loss function as such that the contribution of the adversarial $\mathcal{L}_{wGAN}$ loss is reduced, is a possibility to increase the anatomical consistency to the patient when using heavily undersampled data while still achieving sharper images. Due to these difficulties it remains an open question whether or not our method can be utilized in diagnostic clinical practice, however, we see potential applications in scenarios where a reduction of ionizing radiation exposure to the patient outweighs and a degraded image quality is feasible, a possible example is image registration during minimally invasive and image guided surgeries to precisely locate the surgical instruments inside of the patient.

To provide a theoretical background, Chapter 2 is dedicated to *CT* imaging, where we give insight into *CT* image acquisition as well as *CT* image reconstruction, while Chapter 3 focuses on machine and especially deep learning. The related work will be described in Chapter 4 which we separated into groups depending on how the ionizing radiation dose reduction is achieved as well as the strategy that was used to improve the reconstruction results. Chapter 5 is dedicated to our *CNN* based sparse-view *CT* reconstruction method to improve the quality of the reconstruction results generated from undersampled projection data. The results of our method are presented in Chapter 6, while a discussion is given in Chapter 7. Finally, we conclude our work in Chapter 8.

2

Medical Image Reconstruction

Contents

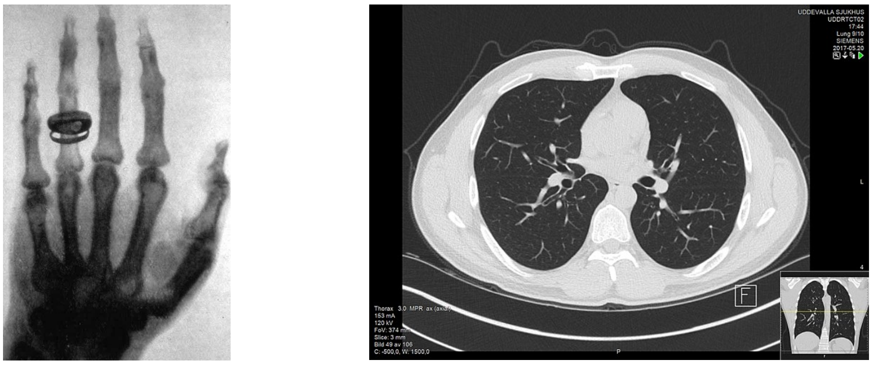
2.1 Imaging Modalities in Medicine **6**
2.2 Inverse Problems **7**
2.3 Computed Tomography **10**
2.4 Image Acquisition in Computed Tomography **12**
2.5 Image Reconstruction in Computed Tomography **20**
2.6 Limitations of Classical Reconstruction Approaches **24**

Medical imaging refers to a set of techniques that have revolutionized medicine by allowing physicians to safely and non-invasively visualize the interior of a body keeping it fully intact. By observing the region of interest, the physician gains insights that make diagnosis of diseases and injuries possible that in the past would only be found at surgery or autopsy. Nowadays, medical imaging supports diagnostic radiologists in their decision process on a daily basis making it a well-established and integral part of healthcare. As such it is no surprise that the invention of various important technologies that lead to medical imaging as it is known today has been awarded with a nobel price, amongst them is the discovery of X-rays by Wilhelm Röntgen [73], the invention of Computed Tomography (CT) by Allan Cormack and Godfrey Hounsfield [3, 34] as well as the invention of Magnetic Resonance Imaging (MRI) by Paul Lauterbur and Peter Mansfield [54].

Insight into important imaging modalities that are used in medicine on a daily basis is given in Section 2.1 and since the problem of image reconstruction in *CT* falls into the category of inverse problems, we will give an insight into this type of problems in Section 2.2. More details on the inner workings of *CT* scanners are given in Section 2.3, while the process of image acquisition is described in Section 2.4. Next, image reconstruction in *CT* is explained in Section 2.5 and lastly, Section 2.6 presents the limitations of classical reconstruction approaches.

2.1 Imaging Modalities in Medicine

While two dimensional (2D) imaging techniques like radiography allow a look *through* the body by yielding a *2D* projection image of a three dimensional (3D) object, *3D* imaging modalities like *CT* and *MRI* yield a volumetric *3D* image of the body allowing a look *inside* it. Figure 2.1a shows an exemplary radiography image and Figure 2.1b represents a volumetric *3D* image consisting of a number of stacked *2D* image slices incorporating detailed structural information of the *3D* object's interior. Radiography is sufficient to observe diseases and injuries that yield a high contrast like bone fractures, while *CT* and *MRI* are utilized to evaluate soft tissues like the abdominal organs or the brain and are able to visualize more subtle abnormalities like nodules or tumors and additionally allow to precisely locate them in *3D*.



(a) Radiography **(b)** Axial *CT* slice

Figure 2.1: Image (a) shows a radiography image acquired by Wilhelm Röntgen showing the hand of Albert von Kölliker.[1] Image (b) shows an exemplary axial *2D* slice of a *CT* image of a normal thorax.[2]

CT and *MRI* are the most important *3D* imaging techniques and while *CT* is based on X-rays and *MRI* on a strong magnetic field and radio waves, they both result in distinctive properties complementing each other. An important advantage of *CT* imaging is represented by the high contrast it achieves between bones and soft tissues, i.e. different organs of the patient, while *MRI* imaging allows a better contrast between individual soft tissues. The main disadvantage of either method can be summarized into the exposure of the patient to ionizing radiation in *CT* and the long acquistion time in *MRI*. It is well known that the exposure to ionizing radiation increases the risk to develop cancer which leads to the consensus that *CT* imaging has a minor chance to harm the patient in the long term [11, 86]. In contrast to *CT*, where images are acquired within seconds, image acquisition in *MRI* takes from at least several minutes up to one hour depending

---

[1] Radiography image taken from Wilhelm Röntgen on 23 January 1896, this work is in the public domain.

[2] High-resolution *CT* slice of an axial plane of a normal thorax, from Mikael Häggström, 20 May 2017, CC0 1.0 Universal Public Domain Dedication.

on the size of the observed region and requires the patient to not move during the whole time to prevent image disruption. This extensive acquisition time leads to non-neglectible difficulties if a patient is unable to lie still during the whole time and also limits the total number of patients that can be examined in a given amount of time compared to *CT*. Therefore, *CT* is preferred in time critical cases like emergencies or when examining body regions like the thorax or abdomen that are influenced by the breathing movement.

2.1.1 Motivation

While the information gained from a *CT* image to accurately treat a patient is considered to outweigh the risks that are caused by ionizing radiation, a further reduction of the amount of ionizing radiation is still desired and represents an active field of research. A simple method to reduce the amount of ionizing radiation the patient is exposed to is represented by reducing the amount of data that is sampled from the patient during image acquisition which is called *undersampling*. However, classical reconstruction methods rely on a certain minimum of data to correctly reconstruct a *CT* image which leads to a heavily degraded image quality of the reconstructed image if this criterion is not met. The work in [9, 24] observed the relation between the radiation dose and the resulting image quality in *CT* imaging demanding better image reconstruction techniques which also represents the main motivation of this thesis. State-of-the-art reconstruction techniques from undersampled data rely on deep learning and can oftenly be interchanged easily between *CT* and *MRI*, which is the reason why we also consider some recent contributions that have been made to the *MRI* community which are motivated by reducing the acquisition time the patient has to endure.

2.2 Inverse Problems

A classical forward problem is a problem where causal factors like physical laws of a system are known and used to predict the outcome for a set of observations. In contrast to that, an inverse problem represents the inverse operation of a forward problem where the outcome for a set of observations is known and the goal is to find the causal factors that lead to that outcome [91]. While forward problems are oftenly easily solvable and deterministic, inverse problems are hard problems and infeasible to solve if too little data is available to find a good approximation of the system.

2.2.1 Image Processing and Signal Processing

As the name suggests, image processing describes a field that is dedicated to processing images using a set of algorithms to solve problems of which many are inverse problems like image denoising, artifact reduction and image reconstruction to name a few [37]. Image processing represents a subfield of signal processing where signals are commonly defined as functions that convey *information about the behavior or attributes of some* *phenomenon* [68]. Analog signals from the *real world* like audio, images or video are continuous and can be captured by taking measurements and transforming them into digital signals to make them readable by computers. However, this digitization of analog signals inevitably leads to a discretization in space or time with the resolution representing the spatial discrete step size in images and videos and the framerate defining the additional temporal discrete step size in videos. Accordingly, a digital *2D* image can be expressed as a number of signals that have been measured from a *2D* grid-like structure in spatial domain at a given time and it follows, that image processing represents a multidimensional case of signal processing which brings the individually measured signals into a spatial correlation.

2.2.2 Undersampling

To digitize an analog system, measurements are acquired from that system at certain intervals which is called *sampling*, where the *sampling rate* defines the discrete intervals of acquisition [90]. Following this definition, digitizing an analog signal is equivalent to discretizing a continuous function and both represent an approximation of the original signal or function and inevitably introduce an error, which is called *aliasing* in signal processing. The amount of aliasing introduced when digitizing an analog signal is directly influenced by the sampling rate used to measure it and as such, it is possible to formulate the minimal sampling rate required to sufficiently digitize an analog signal so that the analog signal can correctly be reconstructed from the digital signal without being heavily aliased, which is defined by the Nyquist-Shannon sampling theorem, see Section 2.2.3. The term *undersampling* now refers to sampling rates that do not fulfill the Nyquist-Shannon sampling theorem [90].

2.2.3 Nyquist-Shannon Sampling Theorem

The Nyquist-Shannon sampling theorem [64, 82] independently also discovered in [44, 94] defines the minimal sampling rate when digitizing an analog signal that still allows to correctly recover the analog signal from it's digital representation. A violation of the Nyqusit-Shannon sampling theorem leads to a loss of information which can lead to a misrepresentation and consequently to a flawed reconstruction of the original signal. If we consider an analog signal with a frequency of 3 Hertz (Hz) that is digitized with a sampling rate of 1.5 *Hz* as shown in Figure 2.2, recovery of the analog signal from these measurements wrongly results in a 1 *Hz* signal due to aliasing caused by undersampling. The Nyquist-Shannon sampling theorem states that the sampling frequency f_s should be greater than twice the maximum frequency f_{max} of the original signal [90], which yields

$$f_s \geq 2f_{max}, \tag{2.1}$$

representing the minimal sampling rate to allow the correct recovery of any analog signal. In image processing, the sampling rate f_s corresponds to the distance between the positions at which the pixel values have been measured.

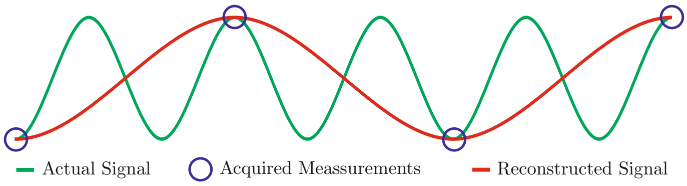


Figure 2.2: Visualization of a signal that was undersampled and reconstructed incorrectly.[3]

2.2.4 Compressed Sensing

Compressed sensing as introduced in [20] allows to recover signals which are undersampled according to the Nyquist-Shannon sampling theorem under certain conditions. In contrast to the Nyquist-Shannon sampling theorem that describes the minimal amount of information required for a linear system to be determined, compressed sensing tackles the problem of finding the correct solution to an underdetermined linear system. The problem of underdetermined systems is that they generally have an infinite number of solutions since they have more unknowns than equations, however, additional constraints can be introduced to find the correct solution to such a system. In the case of compressed sensing, these additional constraints are represented by the sparsity of the observed signal in some domain and incoherence, allowing compressed sensing to determine the optimal solution of the underdetermined system [21].

2.2.5 Computed Tomography as an Inverse Problem

In Section 2.2.1 we explained that image processing represents multidimensional signal processing which follows, that images can be expressed as spatially correlated multidimensional signals. Therefore, if we consider generating a *CT* image of a patient, we first acquire a number of digital signals form the patient that are later used to reconstruct a *CT* image that represents the digital counterpart of the patient. Following the definition given in Section 2.2, signal acquisition in *CT* imaging represents the forward problem, while *CT* image reconstruction poses the inverse problem that requires a certain minimum amount of data to be solved correctly, which is defined by the Nyquist-Shannon sampling theorem as given in Equation (2.1). However, before going into further details, *CT* in general (Section 2.3) as well as the procedures of *CT* image acquisition (Section 2.4) and image reconstruction in *CT* (Section 2.5) need to be explained.

---

[3] Visualization of undersampling, adapted from `http://www.ni.com/newsletter/50078/en/`, last accessed on 22 March 2019.

2.3 Computed Tomography

A *CT* scanner is a device that enables visualization of the interior body structure of a patient yielding a *3D CT* image within seconds. Same as radiography, *CT* is based on ionizing radiation that is sent through the patients body which partially attenuates it, the remaining radiation is then measured. However, in radiography the measurements directly yield the radiography image, whereas in *CT* they just represent one projection view. A *CT* scanner acquires multiple projection views from different angles of the patient, which are then combined using image reconstruction techniques to result in a *3D* reconstructed image of the patient. While a description of the most important parts of a *CT* scanner is given in Section 2.3.1, different types of *CT* devices are explained in Section 2.3.2. The process of acquiring the necessary projection views is described in Section 2.4 and the reconstruction of a *3D CT* image from the acquired projection data is explained in Section 2.5.

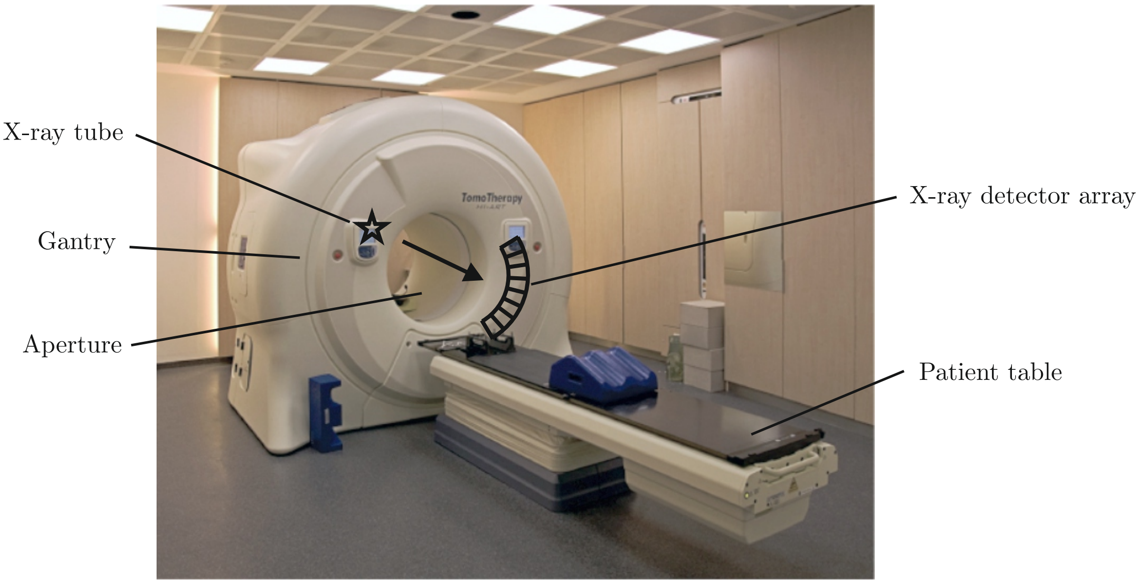


Figure 2.3: A *CT* scanner with additional visualizations and labels.[4]

2.3.1 Schematics of a Computed Tomography Scanner

A *CT* scanner, more precisely a Fan Beam Computed Tomography scanner, as shown in Figure 2.3 consists of the patient table and the gantry. While the patient is lying on the patient table which is a motorized platform that moves slowly through an aperture in the gantry, the gantry acquires the data required for image reconstruction slice by slice. The

---

[4] *CT* scanner with additional visualizations and labels, from OpenStax College, June 2013 via Anatomy and Physiology `http://cnx.org/content/col11496/latest/`, CC BY 4.0 License. © 2013 Rice University

two main components of every *CT* scanner are represented by the X-ray tubes and the X-ray detector arrays which are positioned oppositely to one another in the gantry and rotate around the gantry's aperture during image acquisition as explained in Section 2.4. The number and the placement of the X-ray tubes and the X-ray detector arrays defines the *geometry* and as such, the type of the *CT* scanner.

2.3.2 Types of Computed Tomography Scanners

The technical advance in constructing *CT* scanners lead to a number of various devices that use different geometries for *CT* image acquisition. These individual geometries result in different properties and as such, in different advantages and disadvantages discussed in more detail in [35, 76]. Fan Beam Computed Tomography (FBCT) scanners represent the most widely used devices and will be considered as default unless explicitly stated otherwise throughout this thesis.

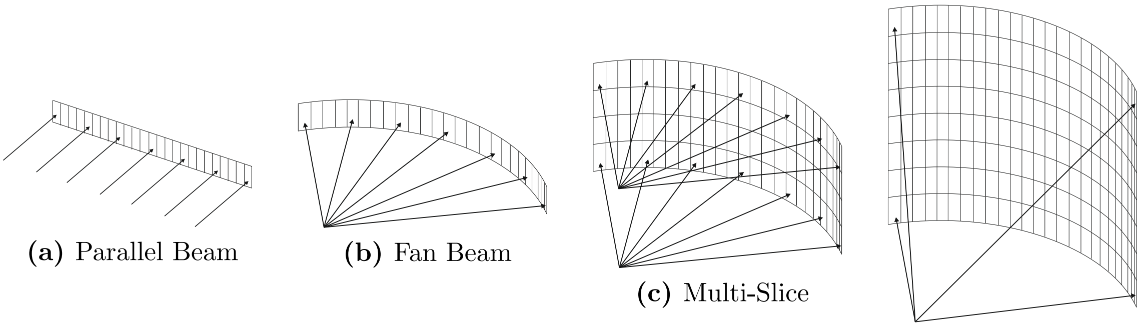

(a) Parallel Beam **(b)** Fan Beam **(c)** Multi-Slice **(d)** Cone Beam

Figure 2.4: Visualization of different geometries of *CT* scanners. Image (a) shows a Parallel Beam, (b) a Fan Beam, (c) a Multi-Slice Fan Beam and (d) a Cone Beam geometry, adapted from [35].

Parallel Beam Computed Tomography Parallel Beam Computed Tomography (PBCT) shown in Figure 2.4a relies as the name suggests on a parallel beam geometry and was used in *CT* scanners of the first generation [35]. In *PBCT* X-rays are emitted as parallel beams and measured accordingly leading to an orthographic projection which is very similar to radiography. This can be achieved either by using multiple X-ray tubes that are positioned next to one another or by a moving X-ray tube that sequentially acquires the necessary signals.

Fan Beam Computed Tomography In *FBCT* visualized in Figure 2.4b the position of the X-ray tube is fixed and X-rays are emitted in a range of angles representing a fan [35]. The detector array is curved as such that the distance to the X-ray tube is equidistant and nowadays, *FBCT* scanners rely on a static detector array that forms a ring in the gantry, however, for simplification we did not visualize that ring in our figures.

While this geometry originally acquires only one slice after another, it can be extended as such, that multiple slices are acquired at once. This results in Multi-Slice Computed Tomography (MSCT) shown in Figure 2.4c, which consists of multiple detector arrays that are stacked on top of one another and allow to simultaneously measure the projection view of a given angle for multiple consecutive slices at once. Most *FBCT* scanners nowadays are implemented as *MSCT* scanners and are widely used in hospitals on a daily basis.

Cone Beam Computed Tomography In Cone Beam Computed Tomography (CBCT) introduced in [71], X-rays are emitted in a cone-shape as shown in Figure 2.4d and the *CT* scanner acquires all necessary projection views in just one rotation [35]. Important benefits of *CBCT* over *FBCT* are a reduced dose [53, 79], a faster image acquisition and a smaller size of the device which makes it also cheaper. However, the cone-beam geometry in *CBCT* leads to problems like beam scatter and beam hardening reducing the contrast of the reconstructed image and degrading it's quality especially when a larger field of view, i.e. a larger part of the body is used [80]. As such, *CBCT* is mostly used and well established in dental and orthodontic imaging where it benefits from it's aforementioned strengths [77].

2.4 Image Acquisition in Computed Tomography

The X-ray tubes and the X-ray detector arrays are the most essential components of a *CT* scanner and described in Section 2.3.1. This section is dedicated to how these components are utilized to yield projection images that can later be used to reconstruct a *CT* image. A patient from which a *CT* scan is conducted is placed between the X-ray tubes and the X-ray detector arrays, in the case of a standard *FBCT* scanner shown in Figure 2.3, the patient lies on the motorized patient table as described in Section 2.3.1. While the X-ray tubes and the X-ray detector arrays rotate around the patient table as visualized in Figure 2.5 to acquire a set of one dimensional (1D) projections from different angles of the currently observed *2D* slice of the patient, the patient table positions the patient accordingly to acquire all views that are necessary for image reconstruction. These acquired sets of *1D* projections are then used to reconstruct the individual *2D* slices of the patient which are finally stacked yielding the *3D* *CT* image. Section 2.4.1 gives insight into ionizing radiation, while the *fundamental photon attenuation law* derived in Section 2.4.2 describes how the measurement of a single *1D* projection view can be used to infer knowledge about the material that was measured. The generation of the individual *1D* projection views using line integrals is given in Section 2.4.3, which is followed by Section 2.4.4 defining the Hounsfield Unit (HU) numbers. Lastly, the creation of the sinogram, i.e. the sequentially stacked *1D* projection views, is described in Section 2.4.5, which is further used for image reconstruction.

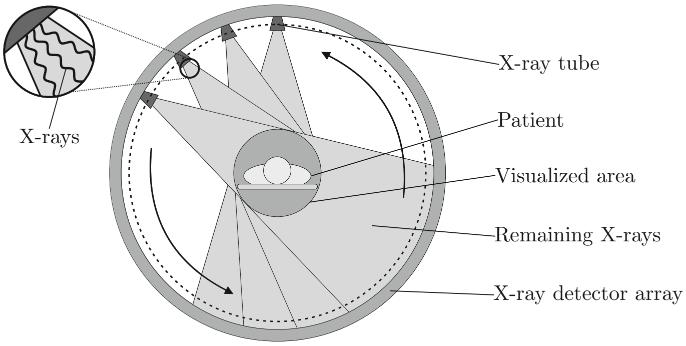


Figure 2.5: A schematic visualization of the image acquisition process in *CT*.[5]

2.4.1 Ionizing Radiation

CT imaging as well as radiography expose the patient to ionizing radiation as mentioned in Section 2.1, which is necessary to acquire a *1D* projection view of the patient. The term *ionizing* refers to radiation with a high enough energy that it is capable of ejecting electrons from atoms, which separates an atom into a ion and a free electron [69]. The two forms of ionizing radiation are represented by electromagnetic and particulate radiation, where the former refers to electromagnetic waves, i.e. photons, while the latter refers to particles like electrons or protons. In the case of *CT* imaging and radiography, high-energetic photons in the wavelength range of X-rays, i.e. with a lower wavelength than visible light, are emitted by the X-ray tube and directed towards the patient.

2.4.2 Fundamental Photon Attenuation Law

To acquire single *1D* projection views, the X-ray tube emits a specific amount of X-rays N_0, i.e. photons with a certain energy E, that are directed towards the patient, which we will replace for now by a slab of some homogeneous material as shown in Figure 2.6. Some X-rays that traverse the material are absorbed or scattered which is dependent on the thickness $\Delta s = s_{\text{out}} - s_{\text{in}}$ of the traversed material as well as on the density of it, which is expressed by the *linear attenuation coefficient* $\mu(E, s)$ [69]. The position s_{in} represents the point where the X-rays enter the material, while the position s_{out} marks the point of exit. The linear attenuation coefficient $\mu(E, s)$ is defined as a function of the energy E and the material at position s and is well defined for different kinds of material [87], see Figure 2.7. The remaining part of the X-rays not attenuated by the body constitute the remaining number of photons N_d, which is measured by the detector array.

Assuming that the material μ, which we want to solve, is homogeneous and the emitted

---

[5] Schematic visualization of *CT* imaging, adapted from `http://www.radtechonduty.com/2017/03/single-detector-row-ct-scan-systems.html`, last accessed on 22 March 2019.

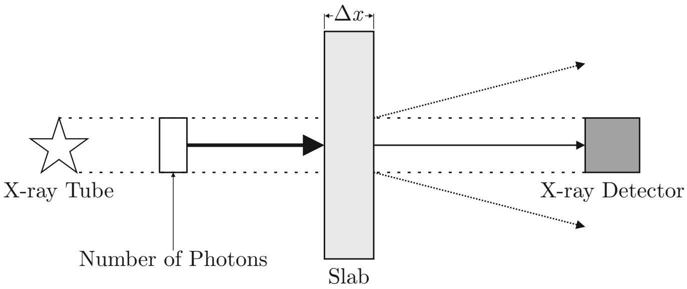


Figure 2.6: A schematic visualization of the attenuation process of photons, adapted from [69].

photons N_0 are monoenergetic allows to define the number of attenuated photons [69] as

$$\begin{aligned} N_a &= \mu \Delta s N_0 \\ &= N_0 - N_d \\ &= -\Delta N, \end{aligned} \tag{2.2}$$

which can be rewritten as

$$\frac{\Delta N}{N_0} = -\mu \Delta s. \tag{2.3}$$

As a next step, we can now integrate on both sides of that equation yielding

$$\int_{N_0}^{N_d} \frac{\mathrm{d}N}{N} = -\int_{s_\text{in}}^{s_\text{out}} \mu \, \mathrm{d}s, \tag{2.4}$$

where the bounds of ΔN are defined by N_0 and N_d as in Equation (2.2), while the bounds of s are given by s_in and s_out representing the positions at which the photons enter respectively exit the material μ. Solving the integral on the left hand side brings us to

$$\ln \frac{N_d}{N_0} = -\int_{s_\text{in}}^{s_\text{out}} \mu \, \mathrm{d}s, \tag{2.5}$$

which can then be reformulated into the *fundamental photon attenuation law* [69] as

$$\begin{aligned} N_d &= N_0 \cdot \mathrm{e}^{-\int_{s_\text{in}}^{s_\text{out}} \mu \, \mathrm{d}s} \\ &= N_0 \cdot \mathrm{e}^{-\mu \Delta s}. \end{aligned} \tag{2.6}$$

2.4.3 Projection View Generation using Line Integrals

The fundamental photon attenuation law defined in Equation (2.6) can also be expressed as an *intensity profile* I instead of using the number of photons N. An intensity profile is defined as

$$I = \hbar \cdot \nu \cdot \frac{N}{A \Delta t}, \tag{2.7}$$

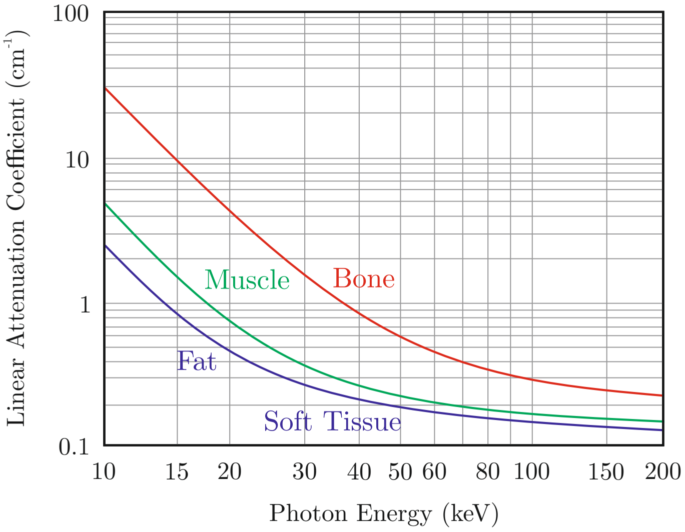


Figure 2.7: A plot of the correlation between the photon energy and the linear attenuation coefficient for different materials related to medical imaging, adapted from [69].

where $\hbar \cdot \nu \cdot N$ describes the combined energy of all photons with the Planck constant $\hbar$ and the photon frequency ν. The area A is normal to the ray and t represents the time [69]. Following this definition allows to reformulate Equation (2.6) as

$$\begin{aligned} I_d &= I_0 \cdot \mathrm{e}^{-\int_{s_{\text{in}}}^{s_{\text{out}}} \mu \, \mathrm{d}s} \\ &= I_0 \cdot \mathrm{e}^{-\mu \Delta s}, \end{aligned} \tag{2.8}$$

however, this formulation still follows the assumptions that the material μ is homogeneous and the photons are monoenergetic defined by energy E. As such, we transform the homogeneous material into a parametric function $\mu(E, s)$, which yields a line integral defined as

$$I_d = \int_0^\infty I_0(E) \cdot \mathrm{e}^{-\int_{s_{\text{in}}}^{s_{\text{out}}} \mu(E,s) \, \mathrm{d}s} \, \mathrm{d}E, \tag{2.9}$$

that also needs to be solved for the energy E. Solving for E is intractable in image reconstruction, however, in *CT* it is sufficient to assume that X-rays are monoenergetic, the term E can be replaced by the *effective energy* $\bar{E}$ which represents the mean of E [69]. Treating $\bar{E}$ as a constant allows it to be omitted [87], leading to the following approximated and simplified formulation

$$I_d = I_0 \cdot \mathrm{e}^{-\int_{s_{\text{in}}}^{s_{\text{out}}} \mu(s) \, \mathrm{d}s}, \tag{2.10}$$

to which we will stick throughout the rest of this thesis, see Figure 2.8.

Bringing the remaining X-ray intensity profile I_d that has been measured by the X-ray detector in relation to the incoming X-ray intensity profile I_0 that has been emitted by the X-ray tube yields the *attenuation profile* representing one acquired *1D* projection g

defined as

$$\begin{aligned} g &= -\ln \frac{I_d}{I_0} \\ &= \int_{s_{\text{in}}}^{s_{\text{out}}} \mu(s)\,\mathrm{d}s. \end{aligned} \tag{2.11}$$

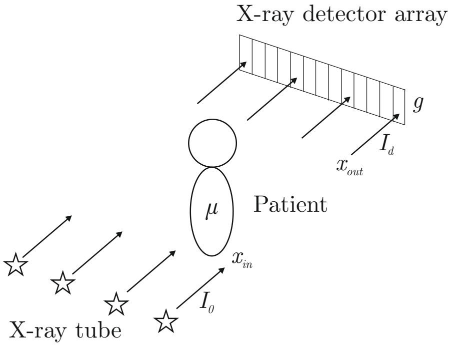


Figure 2.8: A schematic visualization of the projection view generation using line integrals.

2.4.4 Hounsfield Unit Scale

Up to now we also assumed, that the effective energy represented by the intensity profile I_0 is a well-known constant, however, the effective energy depends on the X-ray tube used to acquire the *1D* projections g and is not equivalent among different *CT* scanners, which consequently influences the value of μ when reconstructing the *CT* image. To circumvent this problem and make *CT* images generated from different devices and manufacturers comparable to one another, *CT* images are normalized to the *HU* scale h defined as

$$h = 1000 \times \frac{\mu - \mu_{\text{water}}}{\mu_{\text{water}}}, \tag{2.12}$$

where μ represents the attenuation coefficient of a given material and μ_{water} is defined as the attenuation coefficients of water. The attenuation coefficients μ correspond to the intensity values in *CT* images and by normalizing them to the *HU* scale, the intensity values of different tissues become well-defined. Some value ranges relevant to medical imaging are visualized in Figure 2.9, however, the useful range of the *HU* continues to approximately a value of 3.000, which falls into the the *HU* number range of metal.

2.4.5 Sinogram Generation

In Section 2.4.3 we assumed a *1D* space which is only integrated along it's x-axis. To achieve a more general formulation, the line integral needs to be defined as such, that it can be dynamically used on a *2D* plane. First of all, extending the line integral of

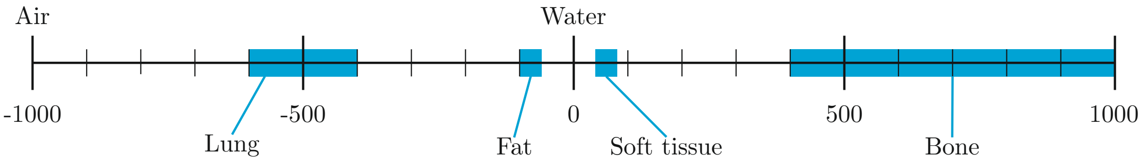


Figure 2.9: A visualization of some *HU* numbers related to medical imaging, adapted from [22].

Equation (2.10) to an xy-plane requires to define a coordinate system (x, y) that represents the axial view of a patient as shown in Figure 2.10. Now, the *2D* slice of the patient that is observed is defined as a *2D* distribution of linear attenuation coefficients $\mu(x, y)$ with it's center point located at the origin of the coordinate system (x, y). We also define a polar coordinate system (ℓ, θ) with the same origin as (x, y), where θ is an angle and ℓ is the distance from the origin as visualized in Figure 2.11. Lastly, we define a circular field of view for $\mu(x, y)$ with diameter FOV that is zero outside similar to [87], where zero corresponds to the linear attenuation coefficient of air defined by the *HU* scale, see Section 2.4.4.

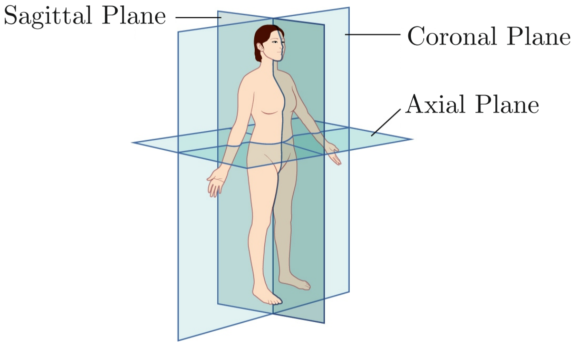


Figure 2.10: A visualization of the different imaging planes of a body in medicine.[6]

Consequently, a *1D* projection as in Equation (2.11) can then be expressed in a *2D* plane in parametric form [69] as

$$\begin{aligned} g_\theta(\ell) &= -\ln \frac{I_\theta(\ell)}{I_0} \\ &= \int_{-\infty}^{\infty} f(x(s), y(s))\, \mathrm{d}s, \end{aligned} \tag{2.13}$$

where the intensity profile I_d is replaced by $I_\theta(\ell)$ expressing the remaining X-rays given a

---

[6] Imaging planes of a body with modified labels, from OpenStax College, June 2013 via Anatomy and Physiology `http://cnx.org/content/col11496/latest/`, CC BY 4.0 License. © 2013 Rice University

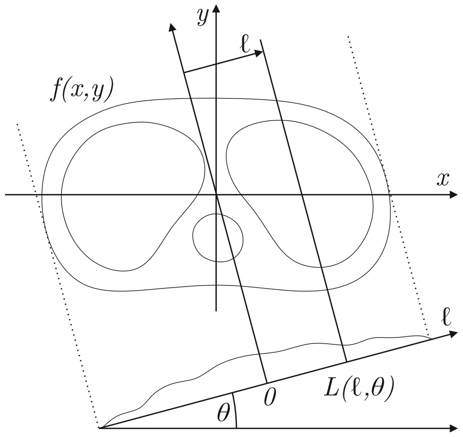


Figure 2.11: A visualization of the forward projection procedure that is relevant to *CT*, adapted from [69].

fixed angle θ, $x(s)$ and $y(s)$ are defined as

$$\begin{aligned} x(s) &= \ell \cdot \cos\theta - s \cdot \sin\theta \\ y(s) &= \ell \cdot \sin\theta + s \cdot \cos\theta. \end{aligned} \tag{2.14}$$

Instead of using the parametric form as in Equation (2.14), a line in a *2D* plane can also be expressed as

$$L(\ell,\theta) = \{(x,y) | x \cdot \cos\theta + y \cdot \sin\theta = \ell\}, \tag{2.15}$$

where θ is an angle and ℓ is the lateral position of the line $L(\ell,\theta)$ which represents a line integral on the *2D* plane (x, y). This allows to reformulate the parametric form of $g_\theta(\ell)$ given in Equation (2.13) as

$$g_\theta(\ell) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x,y) \cdot \delta(x \cdot \cos\theta + y \cdot \sin\theta - \ell) \, \mathrm{d}x \, \mathrm{d}y, \tag{2.16}$$

where the *1D* impulse function $\delta(\cdot)$ leads to zero everywhere except on the line $L(\ell,\theta)$ as shown in Figure 2.12 yielding the line integral [69]. The *1D* projection data $g_\theta(\ell)$ that is acquired from the distribution $\mu(x,y)$ can be stacked for all θ according to the Radon transform [70] resulting in a *2D* dataset $g(\ell,\theta)$ called the sinogram, which is formally expressed for any function $f(x,y)$ as

$$\begin{aligned} g(\ell,\theta) &= \mathcal{R}\{f(x,y)\} \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x,y) \cdot \delta(x \cdot \cos\theta + y \cdot \sin\theta - \ell) \, \mathrm{d}x \, \mathrm{d}y. \end{aligned} \tag{2.17}$$

We used the Shepp-Logan phantom [83] as an exemplary *2D CT* slice that represents the function $\mu(x,y)$ in Figure 2.13 to visualize the generation of inidividual *1D* projections

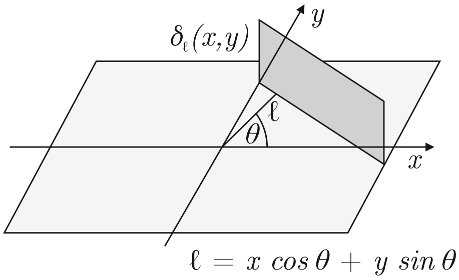


Figure 2.12: A visualization of a line integral on a *2D* plane using a δ function, adapted from [69].

$g_\theta(\ell)$ and leads to the sinogram $g(\ell, \theta)$. The individual *1D* projections $g_\theta(\ell)$ are stacked along the y-axis where the angle θ_y that corresponds to y is defined as

$$\theta_y = \frac{\pi}{N} \cdot (y - 1), \tag{2.18}$$

where N represents the total number of projections $g_\theta(\ell)$. The x-axis of the sinogram $g(\ell, \theta)$ contains the individual line integrals measured and directly corresponds to $g_\theta(\ell)$ as given in Equation (2.16). The projection $g_\theta(\ell)$ is zero for $|r| \geq \frac{\text{FOV}}{2}$ and can be measured from the angle $\theta = [0, 2\pi)$, however, the information is redundant for opposite angles and as such, it is sufficient to measure from $[0, \pi)$ in *PBCT* and from $[0, \pi + \phi)$ in *FBCT*, where ϕ constitutes the fan angle of the *FBCT* scanner [87]. Both, the *1D* projections $g_\theta(\ell)$ as well as the angular information θ from which each projection was acquired from the patient, i.e. the material distribution function $\mu(x, y)$, is then used to reconstruct the original image as described in Section 2.5.

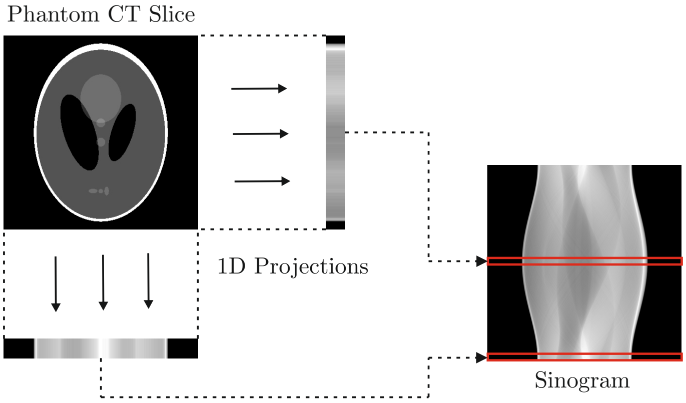


Figure 2.13: A visualization of the sinogram generation procedure.

2.5 Image Reconstruction in Computed Tomography

The *1D* projection data $g_\theta(\ell)$ which is stored in the *2D* sinogram $g(\ell, \theta)$ as explained in Section 2.4 is used to reconstruct the interior of the *3D* patient slice by slice, where each *2D* slice is represented as a *2D* material distribution $\mu(x, y)$. The reconstruction of any given function $f(x, y)$ from it's singram $g(\ell, \theta)$ can mathematically be expressed as the Inverse Radon transform defined as

$$f(x, y) = \mathcal{R}^{-1}\{g(\ell, \theta)\}, \tag{2.19}$$

which demands to find the inverse function to the Radon transform $\mathcal{R}$ given in Equation (2.17). An intuitive method to reconstruct any function $f(x, y)$ and solve the Inverse Radon transform is represented by the backprojection algorithm explained in Section 2.5.1, which however suffers from the discrete nature of the data. The projection-slice theorem described in Section 2.5.2 deduces an analytical solution to the Inverse Radon transform, which is used to derive an optimal reconstruction technique represented by the Filtered Backprojection (FBP) method given in Section 2.5.3.

2.5.1 Backprojection

The backprojection algorithm is an intuitive approach to approximate the Inverse Radon transform given in Equation (2.19) and is conducted by accumulating the backprojections of each *1D* projection $g_\theta(\ell)$ of the sinogram $g(\ell, \theta)$ on a *2D* plane as visualized in Figure 2.15. Backprojection is accomplished by repeating each *1D* projection $g_\theta(\ell)$ that has been acquired from the *2D* material distribution $\mu(x, y)$ representing one slice of the patient in the direction of θ at position ℓ, see Figure 2.14. Mathematically, the backprojection is expressed as

$$\begin{aligned} b_\theta(x, y) &= \mathcal{B}\{g(\ell, \theta)\} \\ &= \int_0^\pi g(x \cdot \cos\theta + y \cdot \sin\theta, \theta)\,\mathrm{d}\theta, \end{aligned} \tag{2.20}$$

and can be transformed into a function in the discrete space as

$$\begin{aligned} b_\theta(x_i, y_j) &= \mathcal{B}\{g(\ell_m, \theta_n)\} \\ &= \sum_{n=1}^{N} g(x_i \cdot \cos\theta_n + y_j \cdot \sin\theta_n, \theta_n)\Delta\theta, \end{aligned} \tag{2.21}$$

where N represents the number of projections and $\Delta\theta$ the rotation interval between subsequent views. Furthermore and due to the rotation by θ, the positions $(x_i \cdot \cos\theta_n + y_j \cdot \sin\theta_n)$ typically need to be interpolated since they generally do not coincide with the discrete positions ℓ_m [87].

The main issue when using the simple backprojection algorithm for image reconstruction becomes clear when looking at the quality of the reconstructed image shown in Figure 2.15, which yields a blurry looking result even when the Nyquist-Shannon sampling theorem in Equation (2.1) is fulfilled, which is caused by the discrete nature of the measurements [87].

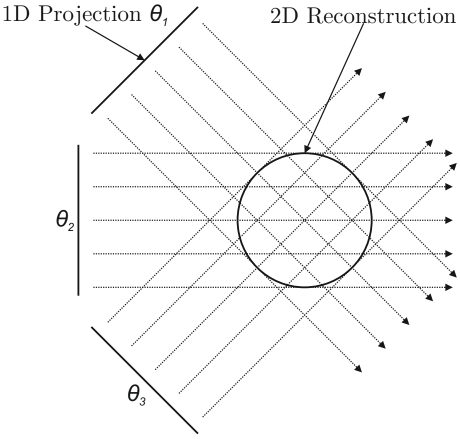


Figure 2.14: A schematic visualization of the backprojection procedure, adapted from [23].

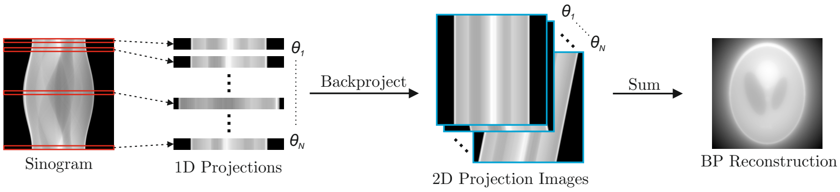


Figure 2.15: A sequential visual demonstration of the backprojection procedure.

2.5.2 Projection-Slice Theorem

The projection-slice theorem introduced in [10] represents a mathematical approach to find a solution to the Inverse Radon transform as given in Equation (2.19). Considering any function $f(x, y)$, we define it's *2D* Fourier transform (FT) $\mathcal{F}_{\text{2D}}$ as

$$\begin{aligned} F(u, v) &= \mathcal{F}_{\text{2D}}\{f(x, y)\} \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x, y) \cdot \mathrm{e}^{-2\pi \mathrm{j}(ux+vy)}\, \mathrm{d}x\, \mathrm{d}y, \end{aligned} \tag{2.22}$$

where the parameters u and v are given as

$$\begin{aligned} u &= \rho \cdot \cos\theta \\ v &= \rho \cdot \sin\theta, \end{aligned} \tag{2.23}$$

with ρ denoting the spatial frequency in arbitrary directions [69]. Furthermore, the *1D FT* $\mathcal{F}_{\mathrm{1D}}$ of the projection $g(\ell,\theta)$ of function $f(x,y)$ is defined as

$$\begin{aligned} G(\rho,\theta) &= \mathcal{F}_{\mathrm{1D}}\{g(\ell,\theta)\} \\ &= \int_{-\infty}^{\infty} g(\ell,\theta)\cdot \mathrm{e}^{-2\pi \mathrm{j}\rho\ell}\,\mathrm{d}\ell. \end{aligned} \tag{2.24}$$

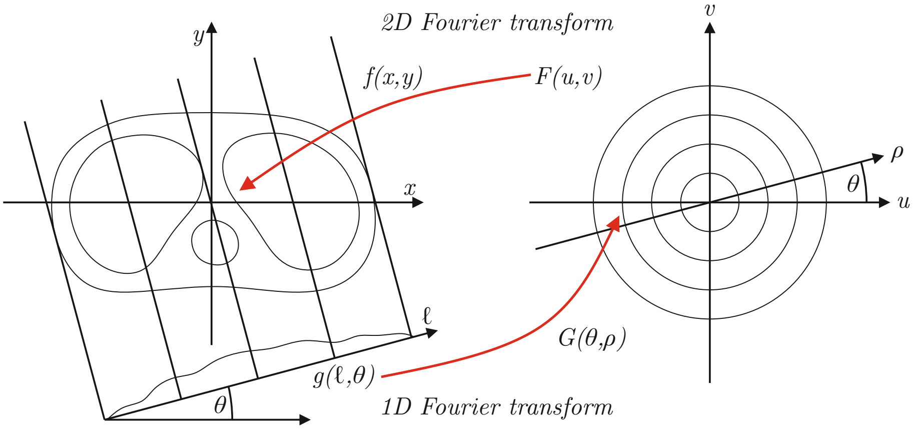


Figure 2.16: A visualization of the projection-slice theorem, adapted from [69].

According to the Radon transformation in Equation (2.17), we can substitute $g(\ell,\theta)$ as such that it yields

$$G(\rho,\theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\cdot\delta(x\cdot\cos\theta + y\cdot\sin\theta - \ell)\mathrm{e}^{-2\pi\mathrm{j}\rho\ell}\,\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}\ell, \tag{2.25}$$

which can then be further manipulated into

$$G(\rho,\theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\int_{-\infty}^{\infty} \delta(x\cdot\cos\theta + y\cdot\sin\theta - \ell)\mathrm{e}^{-2\pi\mathrm{j}\rho\ell}\,\mathrm{d}\ell\,\mathrm{d}x\,\mathrm{d}y. \tag{2.26}$$

As a next step, this expression can then be integrated in respect to ℓ finally resulting in

$$G(\rho,\theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\delta(x\cdot\cos\theta + y\cdot\sin\theta - \ell)\mathrm{e}^{-2\pi\mathrm{j}\rho(x\cdot\cos\theta+y\cdot\sin\theta)}\,\mathrm{d}x\,\mathrm{d}y, \tag{2.27}$$

which is reminiscent of the *2D FT* $\mathcal{F}_{\mathrm{2D}}$ given in Equation (2.22) with the parameters defined in Equation (2.23) [69]. This brings us to the important relationship between the *1D FT* of a projection $g(\ell,\theta)$ and the *2D FT* of any function $f(x,y)$ [87] which defines the projection-slice theorem as

$$G(\rho,\theta) = F(u,v). \tag{2.28}$$

In other words, the projection-slice theorem states that the *1D FT* of the projection

$g(\ell, \theta)$ equals the line that passes through the origin of the *2D FT* of $f(x, y)$ at angle θ, see Figure 2.16. This allows to calculate all points (x, y) of the function $f(x, y)$ from it's projections $g(\ell, \theta)$.

2.5.3 Filtered Backprojection

The *FBP* is a well established analytical method that improves the backprojection algorithm explained in Section 2.5.1 and results in sharp and high quality reconstuction images by introducing an additional filtering step that can be derived from the projection-slice theorem in Section 2.5.2. This filtering step is used to linearly decrease the contribution of low frequency content which is overrepresented in the projection data compared to the high frequency content and leads to blurry looking reconstructions when using the backprojection algorithm [87] as visualized in Figure 2.17.

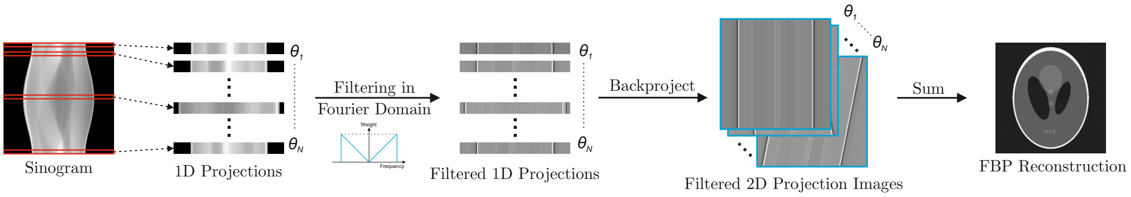


Figure 2.17: A sequential visual demonstration of the *FBP* method.

The straightforward approach called Direct Fourier Reconstruction (DFR) first transforms the projections $g(\ell, \theta)$ into $G(\rho, \theta)$ using the *1D FT* function $\mathcal{F}_{1\mathrm{D}}$ as in Equation (2.24). Then, the analogy between $G(\rho, \theta)$ and $F(u, v)$ as stated by the projection-slice theorem given in Equation (2.28) can be used, which finally allows to utilize the Inverse Fourier transform (IFT) $\mathcal{F}_{2\mathrm{D}}^{-1}$ defined as

$$f(x, y) = \mathcal{F}_{2\mathrm{D}}^{-1}\{G(\rho, \theta)\} \tag{2.29}$$

yielding the reconstruction of $f(x, y)$. However, the *DFR* method leads to an interpolation that would downgrade the quality of the reconstructed image [87]. This interpolation step can be avoided by using the polar variant of the *2D IFT* defined as

$$f(x, y) = \int_0^{2\pi} \int_0^{\infty} F(\rho \cdot \cos\theta, \rho \cdot \sin\theta) \cdot \mathrm{e}^{\mathrm{j}2\pi\rho(x \cdot \cos\theta + y \cdot \sin\theta)} \cdot \rho \, \mathrm{d}\rho \, \mathrm{d}\theta, \tag{2.30}$$

where the additional factor ρ is introduced by the inner derivative when transforming the function $f(x, y)$ into polar coordinates representing the determinant of the Jacobian matrix [69]. Following the projection-slice theorem in Equation (2.28) allows to express $f(x, y)$ as

$$f(x, y) = \int_0^{2\pi} \int_0^{\infty} G(\rho, \theta) \cdot \mathrm{e}^{\mathrm{j}2\pi\rho(x \cdot \cos\theta + y \cdot \sin\theta)} \cdot \rho \, \mathrm{d}\rho \, \mathrm{d}\theta. \tag{2.31}$$

The symmetry of a projection given as $g(\ell, \theta) = g(-r, \theta + \pi)$ makes it possible to change the bounds of the integrals to replace ρ by it's norm $|\rho|$ that represents a ramp filter in Fourier domain [69] yielding

$$f(x, y) = \int_0^\pi \int_{-\infty}^\infty |\rho| \cdot G(\rho, \theta) \cdot \mathrm{e}^{\mathrm{j}2\pi\rho(x\cdot\cos\theta + y\cdot\sin\theta)} \,\mathrm{d}\rho \,\mathrm{d}\theta. \tag{2.32}$$

Since a multiplication in Fourier domain corresponds to a convolution in spatial domain, the term $|\rho|$ can be transformed by the *1D IFT* $\mathcal{F}_{\mathrm{1D}}^{-1}$ into a function $q(\ell)$ given as

$$\begin{aligned} q(\ell) &= \mathcal{F}_{\mathrm{1D}}^{-1}\{|\rho|\} \\ &= \int_{-\infty}^\infty |\rho| \cdot \mathrm{e}^{\mathrm{j}2\pi\rho\ell} \,\mathrm{d}\rho, \end{aligned} \tag{2.33}$$

which allows to express $g(\ell, \theta)$ as

$$g(\ell, \theta) = \int_{-\infty}^\infty g(\ell', \theta) \cdot q(\ell - \ell') \,\mathrm{d}\ell', \tag{2.34}$$

where the convolution $g(\ell', \theta) \cdot q(\ell - \ell')$ can also be written as $g(\ell, \theta) * q(\ell)$ [87].

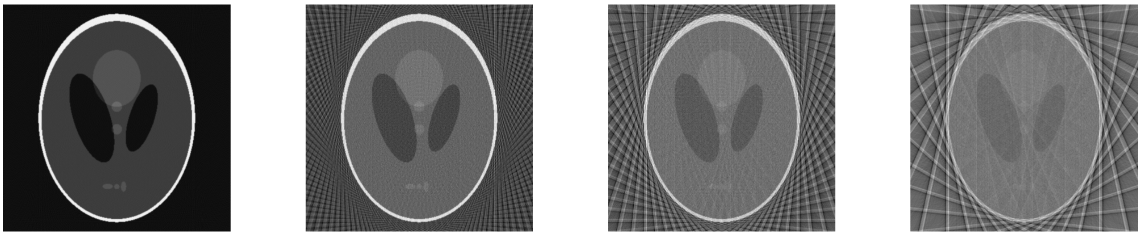
This results in a procedure consisting of two steps, first, each *1D* projection of $g(\ell, \theta)$ is filtered by a ramp filter ρ in Fourier domain or by convolving it with the convolution kernel $q(\ell)$ in spatial domain. Second, the filtered *1D* projections are then backprojected just as in the backprojection algorithm described in Section 2.5.1, which finally yields the definition of the *FBP* method as

$$\begin{aligned} f(x, y) &= \int_0^\pi \int_{-\infty}^\infty |\rho| \cdot G(\rho, \theta) \cdot \mathrm{e}^{\mathrm{j}2\pi\rho(x\cdot\cos\theta + y\cdot\sin\theta)} \,\mathrm{d}\rho \,\mathrm{d}\theta \\ &= \int_0^\pi \int_{-\infty}^\infty g(\ell, \theta) * q(\ell) \,\mathrm{d}\ell \,\mathrm{d}\theta, \end{aligned} \tag{2.35}$$

with $\ell = x\cdot\cos\theta + y\cdot\sin\theta$ resulting in a sharp reconstruction result as shown in Figure 2.17.

2.6 Limitations of Classical Reconstruction Approaches

Classical reconstruction methods like the *FBP* are widely used and reliable methods to reconstruct *CT* images if a sufficient amount of projection views as defined by the Nyquist-Shannon sampling theorem described in Section 2.2.3 is available. However, if only undersampled data is available, classical methods are heavily burdenend by artifacts degrading the quality of the reconstructed image [8] as exemplary shown in Figure 3.7. The number of projection views directly correlates to the amount of ionizing radiation the patient is exposed to, which stands in stark contrast to greatly reducing the risk for the patient to develop cancer [11, 86]. Some work was conducted to observe the relation between the amount of ionizing radiation used and the quality of the reconstructed *CT* image resulting from that dose and demanded better reconstruction techniques [9, 24]. Most state-of-the-art methods that aim to reduce the amount of ionizing radiation used for *CT* image reconstruction rely on machine and especially deep learning, which will be explained in Chapter 3, before we go into more details about different approaches that have been recently proposed in Chapter 4.



(a) 360 views **(b)** 60 views **(c)** 30 views **(d)** 15 views

Figure 2.18: Different reconstruction results using a reduced number of projection views generated by the *FBP* method.

3

Machine Learning

Contents

3.1 Supervised Learning **28**
3.2 Deep Learning **29**
3.3 Optimization **32**
3.4 Convolutional Neural Networks **35**
3.5 Architectures of Neural Networks **37**
3.6 Generative Adversarial Networks **40**
3.7 Training Neural Networks **46**

Machine learning is a thriving field of research in computer science that has a wide range of applications. A definition that is attributed to Arther Samuel going back to 1959 describes machine learning as the ability of computers to learn how to solve problems *without being explicitly programmed* [45]. Following this definition, machine learning systems are systems that are able to learn good features from available data and optimize themselves from that past experience to solve a given task without the need of human intervention or assistance. Instead, learning is accomplished by utilizing data and implicitly learning a general representation of that data, which can then be applied to unseen data samples from which new information is deduced.

While different subfields of machine learning exist, we are focusing on supervised learning which is explained in Section 3.1 in this thesis. Deep learning will be explained in Section 3.2, which is followed by Section 3.3 that is dedicated to the optimization of deep learning algorithms. In Section 3.4 we describe Convolutional Neural Networks (CNNs) that resemble an important kind of deep learning widely used in the field of computer vision, while typical architectures of neural networks are given in Section 3.5. Generative Adversarial Networks (GANs) will be discussed in Section 3.6 and lastly, Section 3.7 is dedicated to topics related to the training of neural networks.

3.1 Supervised Learning

Supervised learning is the most commonly used type of machine learning and requires not only a large amount of data samples $x \in X$ but also a corresponding label for each sample serving as the ground truth $y \in Y$. When considering the classical classification problem of distinguishing images of cats and dogs from one another, then the images x represent the data samples that serve as an input for the classifier algorithm A, whereas the binary label information for the class *cat* or *dog* is the correct answer for a given data sample and called the ground truth y. The classifier algorithm A consists of model parameters $\boldsymbol{\theta}$ and for a given input x, algorithm A yields an output prediction $\hat{y}$ that is influenced by $\boldsymbol{\theta}$. The output prediction $\hat{y}$ is then compared to the corresponding ground truth y as shown in Figure 3.1, where the optimal solution given as $y = \hat{y}$ defines the optimal model parameters $\boldsymbol{\theta}^*$.

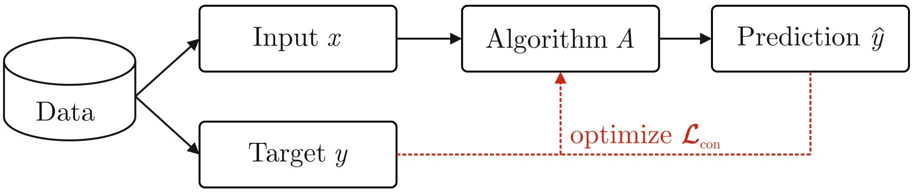


Figure 3.1: A schematic visualization of supervised learning.

To improve the performance of A during training, an objective or loss function $\mathcal{L}$ is used to meassure the error which is typically accomplished by calculating the distance between y and $\hat{y}$. The model parameters $\boldsymbol{\theta}$ are then updated according to the error as such, that the performance of A is improved in the next iteration when calculating the loss function $\mathcal{L}$ as explained in more detail in Section 3.3. The overall performance of a classifier algorithm A given a data set with labels can be estimated by generating the prediction of the classifier for each data sample in that data set and calculating the correct prediction rate.

Depending on the specific problem formulation and how the algorithms or models are trained, they can be further separated into discriminative and generative models [62]. Discriminative models like the mentioned cat and dog classifier are explained in Section 3.1.1, while generative models are given in Section 3.1.2.

3.1.1 Discriminative Models

A discriminative model predicts a label y from a given data sample x and models the decision boundary between different classes. As such, a discriminative model makes it's prediction by checking on which side of the learned decision boundary a given data sample falls focusing on distinguishing between the predefined set of classes without learning specific properties of them. Discriminative models learn a conditional probability distribution and can be described as functions that optimize

$$\arg\max_{y} p(y|x). \tag{3.1}$$

3.1.2 Generative Models

A generative model is able to learn the actual distribution of each class and thus, is able to generate a sample of x given a label y. In contrast to discriminative models, a generative model learns the specific properties of each class and solves the task of classification by calculating the similarity of the features of a given sample to the typical features of each class. By utilizing the Bayes' theorem

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}, \tag{3.2}$$

we can reformulate a discriminative model as a generative model as

$$\arg\max_{y} p(x|y)p(y) = \arg\max_{y} p(x, y), \tag{3.3}$$

where the term $p(x)$ can be omitted due to optimizing $\arg\max$ of y resulting in a joint probability distribution.

3.2 Deep Learning

Deep learning is one method to implement machine learning and is typically based on Artificial Neural Networks (ANNs) that consist of millions of interconnected artificial neurons which learn to transform given inputs into an output with some meaning. During the optimization of the model parameters $\boldsymbol{\theta}$ of an *ANN* called training as explained in Section 3.3, the *ANN* learns features from the training data and in the ideal case, the *ANN* is able to identify a general representation of the data and also achieves a good performance on unseen data samples. The breakthrough of deep learning in 2012 with the AlexNet [46] significantly outperforming state-of-the-art methods for image classification in that time lead to a highly increased interest in deep learning related research. Amongst them is the work proposed in [33] which was able to surpass human performance in image classification as well as contributions to other fields like speech recognition [4, 88] and text processing [103]. An overview on recent accomplishments in the field of medical imaging is given in [52, 56].

3.2.1 Artificial Neurons

Artificial neurons are inspired by neurons in biological brains shown in Figure 3.2 and similar to their biological counterpart, artificial neurons are interconnected signal processing units that modify and transmit incoming signals originally introduced in [57].

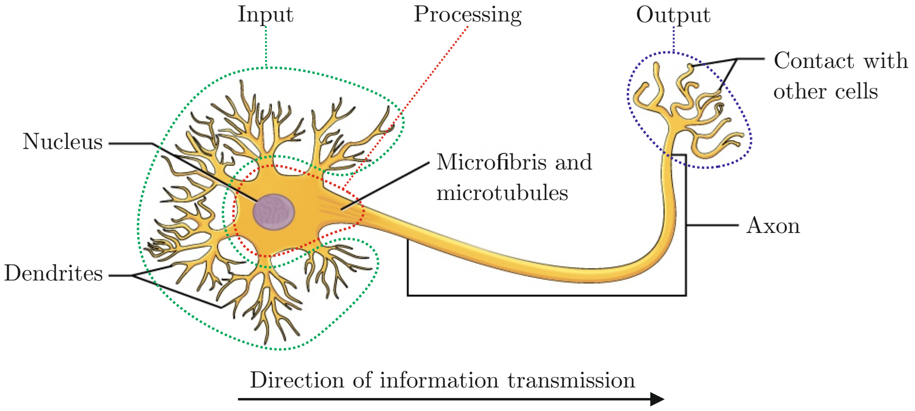

Input
Processing
Output
Contact with other cells
Nucleus
Microfibris and microtubules
Axon
Dendrites
Direction of information transmission

Figure 3.2: A schematic visualization of a biological neuron.[1]

Mathematically speaking, each artificial neuron receives a specific number of K inputs $\boldsymbol{x}$ and contains a set of $K+1$ parameters $\boldsymbol{\theta} = \{b, \boldsymbol{w}\}$ consisting of one bias b and K weights $\boldsymbol{w} = \{w_1, \ldots, w_K\}$, see Figure 3.3. To arrive at a more convenient formulation, we concatenate a constant factor $x_0 = 1$ to the input vector which is then defined as $\boldsymbol{x} = \{x_0, x_1, \ldots, x_K\}$ also having a size of $K+1$ equally to $\boldsymbol{\theta}$.

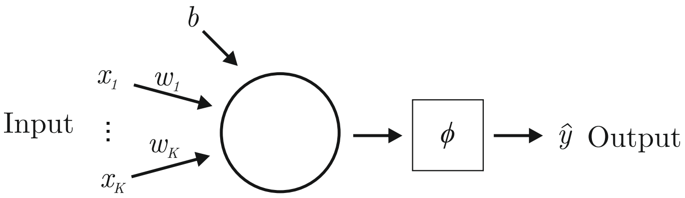


Figure 3.3: A schematic visualization of an artificial neuron.[2]

Following this formulation, each parameter θ_k corresponds to one input x_k defining the relative importance of x_k in that neuron, while the bias b represents a weight that is independent of any input and allows a constant translation along the x-axis of the resulting function. Combining the set of inputs with the parameters yields the output prediction $\hat{y}$ of that neuron defined as

$$\hat{y} = \sum_{k=0}^{K} x_k \cdot \theta_k. \tag{3.4}$$

This formulation, however, allows only a linear transformation of the input to calculate the output prediction $\hat{y}$, which represents a limitation if the distribution of the data is non-

---

[1] Visualization of a biological neuron, mirrored with modified labels and additional visualizations, from OpenStax College, June 2013 via Anatomy and Physiology `http://cnx.org/content/col11496/latest/`, CC BY 4.0 License. © 2013 Rice University

[2] Visualization of an artificial neuron, adapted from `https://medium.com/@jayeshbahire/the-artificial-neural-networks-handbook-part-4-d2087d1f583e`, last accessed on 22 March 2019.

linear. By introducing a non-linear activation function ϕ visualized in Figure 3.3, neurons are able to learn non-linear representations of the data. Since real world data is typically non-linear, an activation function improves the performance of a neuron significantly. Incorporating ϕ leads to the following formulation:

$$\hat{y} = \phi(\sum_{k=0}^{K} x_k \cdot \theta_k), \tag{3.5}$$

important activation functions are explained in Section 3.4.3. This formulation can be manipulated by directly expressing the inputs and the weights as vectors yielding the simplified notation given as

$$\hat{y} = \phi(\boldsymbol{x} \cdot \boldsymbol{\theta}) = \phi(\boldsymbol{x}; \boldsymbol{\theta}). \tag{3.6}$$

The first learning algorithm for single artificial neurons was proposed in [74] and showed that they are capable of finding a solution for linearly separable data. While it was shown that single artificial neurons can solve the logical operators AND, OR and NOT, they are unable to solve the XOR problem which requires a non-linear solution as demonstrated in [58], which, however, can be solved by stacking multiple artificial neurons and forming a simple *ANN* [25, 27].

3.2.2 Feedforward Neural Networks

Concatenating multiple artificial neurons as such that the output of a preceding neuron is used as an input of it's successive neuron leads to an *ANN*. Restricting the network by prohibiting circular connections and only allowing the input to be strictly forwarded through the artificial neurons until the output is reached forms a Feedforward Neural Network (FNN), see Figure 3.4.

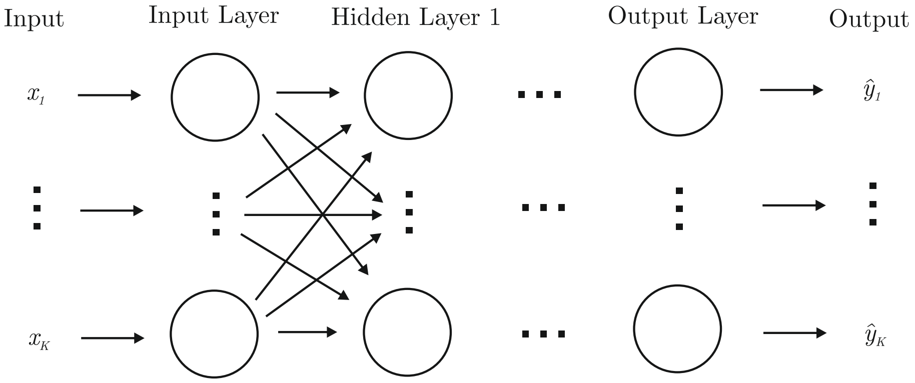


Figure 3.4: A schematic visualization of a *FNN*.

All artificial neurons of a given *depth* in the *FNN* are referred to as one *layer* and we define all layers in this simple *FNN* to be *fully-connected layers*, which means that all artificial neurons of a preceding layer are connected to all artifical neurons of a successive layer. While the first and last layer are called the input and output layer accordingly, intermediate layers are defined as hidden layers. The number of layers present in a *FNN* is referred to as the *depth* of the network and explains the nomenclature of deep learning oftenly quantified with at least 8 layers going back to the AlexNet [46]. Considering a *FNN* of depth D and reformulating the input of the network as $a_0 = x$ and the output as $a_D = \hat{y}$ allows to extend the definition of Equation (3.6) into

$$a_d = \phi_{d-1}(\boldsymbol{a}_{d-1}; \boldsymbol{\theta}_{d-1}) \tag{3.7}$$

with $1 \leq d \leq D$. Following this formulation shows that a *FNN* can be expressed as a parametric function f with parameters $\boldsymbol{\theta}$ that learns to map an input x to some output $\hat{y}$ yielding

$$\hat{y} = f(x; \boldsymbol{\theta}). \tag{3.8}$$

The function f can be disassembled into a set of sequential functions [25] defined as

$$f(x; \boldsymbol{\theta}) = f_d(f_{d-1}(x; \boldsymbol{\theta}_{d-1}); \boldsymbol{\theta}_d), \tag{3.9}$$

again with $1 \leq d \leq D$. The parameters $\boldsymbol{\theta}$ of the function f are iteratively optimized using data which is called training. The goal of training the function f is to approximate the optimal parameters $\boldsymbol{\theta}^*$ representing the ideal solution defined as $y = f(x; \boldsymbol{\theta}^*)$, see Section 3.3.

3.3 Optimization

The optimization of a deep learning based algorithm A is done by minimizing an objective function, i.e. a loss function $\mathcal{L}$ that measures the error between the ground truth y and the prediction $\hat{y}$ generated from A given a data sample x. Calculating the global minimum of a typically non-convex loss function $\mathcal{L}$ used by deep learning networks that consist of millions of parameters is analytically infeasible and demands a solution achieved through approximation. Over the course of training the algorithm A, this objective function $\mathcal{L}$ is used to iteratively measure the error by slightly modifying the internal parameters $\boldsymbol{\theta}$ based on the data samples seen during the current training iteration. Calculating the gradients of the loss function $\mathcal{L}$ in respect to the parameters $\boldsymbol{\theta}$ allows to modify $\boldsymbol{\theta}$ as such, that $\mathcal{L}$ is minimized until it saturates in a local minimum, where $\mathcal{L}$ is considered to have converged and the training procedure stops.

3.3.1 Loss Function

Conventional loss functions like the $\mathcal{L}_1$ or the $\mathcal{L}_2$ loss function are still widely used to optimize the parameters $\boldsymbol{\theta}$ due to their universal applicability and good performance on many tasks. While the $\mathcal{L}_1$ loss represents the Mean Absolute Error (MAE) calculated as the mean absolute difference of the prediction $\hat{y}$ to the ground truth y of a given data set $m \in M$ and is defined as

$$\mathcal{L}_1 = \frac{1}{m} \sum_{m \in M} |\hat{y}_m - y_m|, \tag{3.10}$$

the $\mathcal{L}_2$ loss or Mean Squared Error (MSE) is the mean squared difference and defined as

$$\mathcal{L}_2 = \frac{1}{m} \sum_{m \in M} (\hat{y}_m - y_m)^2. \tag{3.11}$$

These rather basic conventional loss functions, however, can be replaced by more task specific loss functions or combinations of multiple loss functions to improve the performance of the algorithm on a specific task. A more sophisticated training scheme is represented by a *GAN* introduced in [26] and explained in Section 3.6.

3.3.2 Optimizer

An optimizer O defines how the parameters $\boldsymbol{\theta}$ need to be adapted from iteration to iteration to minimize $\mathcal{L}$ as far as possible and ultimately arrive at a good local minimum. Gradient Descent (GD) represents a traditional optimizer that requires to calculate the gradient with respect to the parameters $\boldsymbol{\theta}$ for each sample x_m in the dataset X_M, where M represents the number of samples. The gradient of any sample x_m with the corresponding target y_m is defined as

$$g_{\boldsymbol{\theta}_m} = \nabla_{\boldsymbol{\theta}} \mathcal{L}(f(x_m; \boldsymbol{\theta}), y_m), \tag{3.12}$$

where $\mathcal{L}$ is a loss function as explained in Section 5.3.3 and the function f is defined in Equation (3.8) [25]. The final gradient over all samples in X_M is given as

$$g_{\boldsymbol{\theta}} = \frac{1}{M} \sum_{m=1}^{M} g_{\boldsymbol{\theta}_m}, \tag{3.13}$$

which is then used to update the parameters $\boldsymbol{\theta}$ yielding

$$\boldsymbol{\theta} := \boldsymbol{\theta} + \eta \cdot -g_{\boldsymbol{\theta}}. \tag{3.14}$$

The term η represents a hyperparameter called the *learning rate* that controls the magnitude of the update. As such, each update using *GD* requires to calculate the gradient $g_{\boldsymbol{\theta}_m}$ for each sample x_m in the dataset X_M, which becomes time intensive if the number of data samples M is very large. Stochastic Gradient Descent (SGD) presents a solution, where the gradients are only calculated for a small random subset $X_K \subset X_M$, instead of calculating them for the full dataset X_M. The reduced dataset X_K is called a *mini-batch* and is typically defined as such, that repeating samples x_m are only allowed in the subset X_K after each sample of X_M has been chosen exactly once [25]. This brings us to the formal expression of *SGD* which represents an approximation of the *GD* given in Equation (3.13) and is defined as

$$g_{\boldsymbol{\theta}} \approx \frac{1}{K} \sum_{k=1}^{K} g_{\boldsymbol{\theta}_k}. \tag{3.15}$$

However, both *GD* and *SGD* require to find a good value for the learning rate η that defines how much the parameters $\boldsymbol{\theta}$ are modified per iteration. Finding the optimal value for the learning rate η is task specific and not trivial, while a small learning rate significantly increases the time the algorithm requires to converge and might result in the algorithm getting stuck in a bad local minimum, a large η can lead to an oscillation of the loss function $\mathcal{L}$ where the local minimum is overshot back and forth oftenly resulting in non-convergence.

Adaptive learning rate optimizers do not rely on an individual global learning rate η but utilize an individual learning rate for every trainable parameter in $\boldsymbol{\theta}$, which showed to work better for a wide range of setups than *SGD*. A popular adaptive learning rate optimizer is represented by Adaptive Moment Estimation (ADAM) introduced in [41], which incorporates momentum that accelerates learning by accumulating an exponentially decaying moving average of the past gradients defined as

$$\boldsymbol{\mu} := \beta_1 \cdot \boldsymbol{\mu} + (1 - \beta_1) \cdot g_{\boldsymbol{\theta}} \tag{3.16}$$

$$\boldsymbol{v} := \beta_1 \cdot \boldsymbol{v} + (1 - \beta_1) \cdot {g_{\boldsymbol{\theta}}}^2. \tag{3.17}$$

The estimates of the first and second moment $\boldsymbol{\mu}$ and $\boldsymbol{v}$ of the gradients allow a movement in the optimal direction without suffering from small or noisy gradients [25]. Following the definition in [41] we define

$$\hat{\boldsymbol{\mu}} = \frac{\boldsymbol{\mu}}{1 - \beta_1} \tag{3.18}$$

$$\hat{\boldsymbol{v}} = \frac{\boldsymbol{v}}{1 - \beta_2}, \tag{3.19}$$

which represent bias-corrected estimates that finally allow to redefine the parameter update given in Equation (3.14) as

$$\boldsymbol{\theta} := \boldsymbol{\theta} - \frac{\eta}{\sqrt{\hat{\boldsymbol{v}}} + \epsilon} \cdot \hat{\boldsymbol{\mu}}, \tag{3.20}$$

where ϵ is used to prevent a division by zero. The recommended hyperparameters achieving a good performance among various tasks according to [41] are given by $\eta = 10^{-3}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$.

3.4 Convolutional Neural Networks

In fully-connected *FNNs* every artifical neuron of a preceding layer is connected to every neuron of a successive layer with the exception of the input. The vast increase of parameters per layer that need to be trained in fully-connected *FNNs* heavily influences the performance and memory requirements of these networks limiting their potential. One possibility to circumvent this problem is exploiting additional information that is implicitly encoded in some data, like spatial correlation in images or time information from sensor data or in speech. This is implemented in *CNNs* that are defined in [25] as

> *Convolutional networks are neural networks that use convolution in place of general matrix multiplication in at least one of their layers.*

3.4.1 Convolution

A discrete convolution in two dimensional (2D) space is defined as

$$S(i,j) = (K * I)(i,j) = \sum_m \sum_n I(i-m, j-n)K(m,n), \tag{3.21}$$

where each pixel i, j of a *2D* image I is convolved with each pixel m, n of a *2D* kernel K yielding the convolved image S called feature map [25]. While an example of a *2D* convolution of an image with a kernel of size 3 × 3 is shown in Figure 3.5, this formulation can be extended to three dimensional (3D) volumetric data by adding an additional dimension to the formula leading to

$$S(i,j,k) = (K * I)(i,j,k) = \sum_m \sum_n \sum_o I(i-m, j-n, k-o)K(m,n,o). \tag{3.22}$$

Sparse Connectivity In contrast to a fully-connected *FNN*, a *CNN* is a specialized neural network based on convolutions, that restricts the number of connections between a preceding layer and a successive layer. This important property of *CNNs* is called sparse connectivity and leads only to connections between neighboring pixels in spatial domain, where the convolution kernel acts as a receptive field of a given pixel.

Parameter Sharing Another property of *CNNs* is parameter sharing which allows to reuse the learned weights of a convolution kernel at any pixel in an image rather than learning individual weights for each pixel position as it would be necessary for fully-connected layers. As such, each convolution kernel of a *CNN* that was trained to detect one specific feature is able to detect that feature independently of the position in the image. Parameter sharing in *CNNs* heavily reduces the amount of parameters that need to be trained per layer and allows to use more layers yielding better results in many tasks.

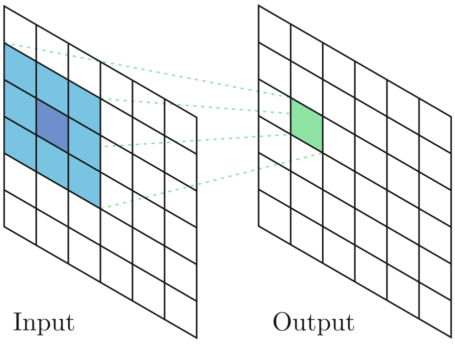


Figure 3.5: A visualization of a convolution kernel.

3.4.2 Pooling

Pooling layers are widely used in *CNNs* after a convolution layer or a set of convolution layers and transform it's input into a summary statistics consequently decreasing the size of the input, see Figure 3.6. As such, pooling layers fulfill different tasks, they enforce invariance to small translations, reduce the number of parameters of the network as well as increase the receptive field of the following layers by downsampling the image [25]. An established method is represented by max pooling, where a square neighborhood typically of a size 2×2 is observed and only the maximum value of that neighborhood is preserved in the output resulting in a reduced image size [107].

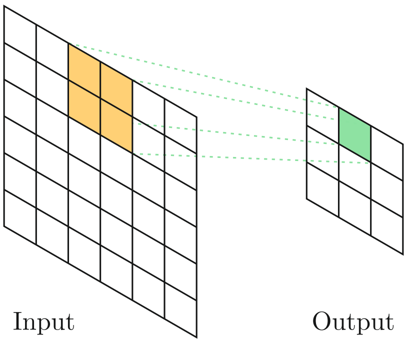


Figure 3.6: A visualization of a pooling layer.

3.4.3 Activation Functions

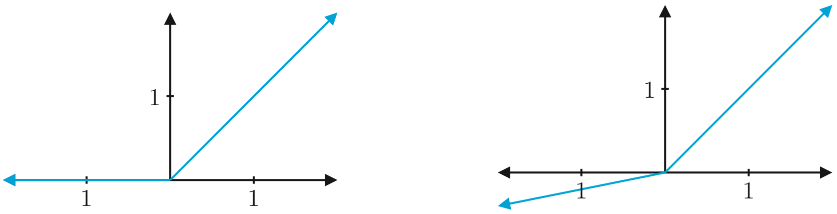
As explained in Section 3.2.1, activation functions can be used to introduce non-linear functions to a neural network, which enables the network to learn non-linearities in the data. The Rectified Linear Unit (ReLU), see Figure 3.7a, represents an important activation function which was introduced in [60] as

$$\phi(x) = \max(0, x), \tag{3.23}$$

where the input x corresponds to the output of an artificial neuron. *ReLUs* are widely used in the convolution layers of *CNNs*, since they show a good performance on many different tasks leading to a fast convergence towards a local minimum [25]. Due to the definition of *ReLUs* yielding zero for $x \leq 0$, it is possible that some neurons *die*, i.e. that they are always inactive and never contribute to the prediction of the neural network. A solution to this problem was introduced in [33] called Leaky Rectified Linear Unit (Leaky ReLU) which is defined as

$$\phi(x) = \begin{cases} \alpha x & x < 0 \\ x & x \geq 0, \end{cases} \tag{3.24}$$

where the additional parameter typically defined as $\alpha = 0.2$ ensures the existence of a non-zero gradient for negative inputs x as visualized in Figure 3.7b. As such, *Leaky ReLUs* solve the problem of (dying) artificial neurons [25].



(a) *ReLU* **(b)** *Leaky ReLU*

Figure 3.7: Visualization of important activation functions. Image (a) shows a *ReLU*, image (b) shows a *Leaky ReLU*.

3.5 Architectures of Neural Networks

The sequence of layers implemented into a neural network is called the architecture of the network and implicitly defines how a network is operating. Each layer performs a transformation from the given input to an output and forwards the output to the next layer. Two prominent and widely used architectures that serve different purposes and solve a wide range of problems are represented by encoder and decoder networks. While encoders are explained in Section 3.5.1, a description of decoders is given in Section 3.5.2. More specific architectures that are relevant to this thesis are given in Section 3.5.1.1 and Section 3.5.2.1.

3.5.1 Encoder

An encoder network is a neural network that learns to reduce the complexity of a given input by transforming it into a feature representation ideally consisting only of the most significant features for the task. *CNNs* typically accomplish this by concatenating multiple sets consisting of individual or multiple convolution layers with a non-linear activation function as in Equation (3.5) which is followed by a pooling layer. During the training of such a *CNN*, the first convolution layer learns to extract meaningful features directly from the input, while follow-up convolution layers combine the previously learned features into more complex ones. The pooling layers serve the purpose of making the *CNN* translational invariant and also reduce the number of features for all successive layers. An exemplary encoder *CNN* is given in Figure 3.8.

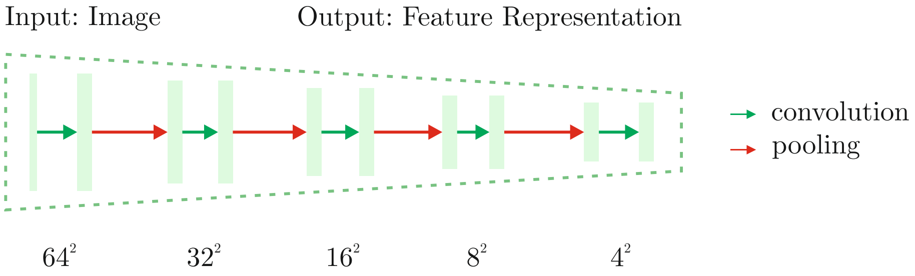


Figure 3.8: A visualization of an encoder neural network.

3.5.1.1 Classification Neural Network

Binary classification *CNNs* that predict one class out of two for a given input image are implemented as encoder networks with an additional fully-connected *FNN* at the end that transforms the feature representation into a single scalar value, see Figure 3.9. This single scalar value can also be replaced by a number of C scalar outputs, where C corresponds to the number of classes which is defined as $C = 2$ in the case of binary classification. These outputs can then be transformed using a softmax function [25] into values that sum up to one, where the maximum value represents the predicted class. The advantage of this formulation is that it can also solve a multiclass classification task, since the number of classes C can be chosen arbitrarily. Additionally, in the case of binary classification the C outputs of the *CNN* directly represent the probabilities for either class.

3.5.2 Decoder

A decoder network is a neural network that inversely complements an encoder network by learning a transformation from a feature representation to an output, i.e. an image with some meaning in the case of *CNNs*. A typical decoder consists of a consecutive series of an upsampling and one or multiple convolution layers with a non-linear activation

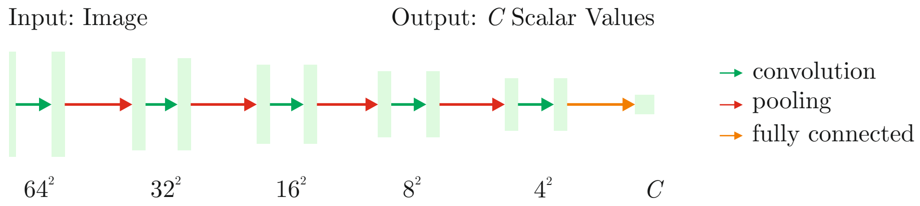


Figure 3.9: A visualization of a classification neural network.

function as in Equation (3.5), where a possible implementation of an upsampling layer is presented by a resampling of the given input to the desired size which is followed by an interpolation resulting in the desired output. The consecutive convolution layers learn new representations of the data by transforming the received features into new ones. Figure 3.10 shows a typical decoder *CNN*, however, a decoder that is defined as such still yields an output representing features. To finally transform the feature representation into a more meaningful output like an image in the case of a *CNN*, a fully-connected layer similarly to a *FNN* is used at the end of the decoder to learn the optimal combination of all features yielding that image.

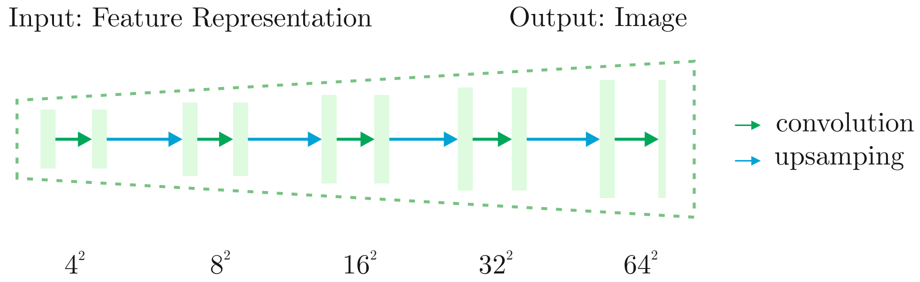


Figure 3.10: A visualization of a decoder neural network.

3.5.2.1 Image-to-Image Neural Network

Another type of network architectures is represented by image-to-image *CNNs* that receive an image as an input and learn to transform the image of some input space into an image of some output space, where the input and output space are defined by the task. Image-to-image *CNN* combine an encoder and a decoder with the encoder being trained to transform the input image into a feature representation and the decoder being optimized the construct an image in the target space from that feature representation. A well established and widely used architecture is represented by the U-Net introduced in [72] and visualized in Figure 3.11. The U-Net uses additional skip connections that directly connect intermediate convolution layer outputs from the contracting path representing the encoder to the expanding path, i.e. the decoder. These skip connections allow the contribution of high frequency contents to the final output of the *CNN*.

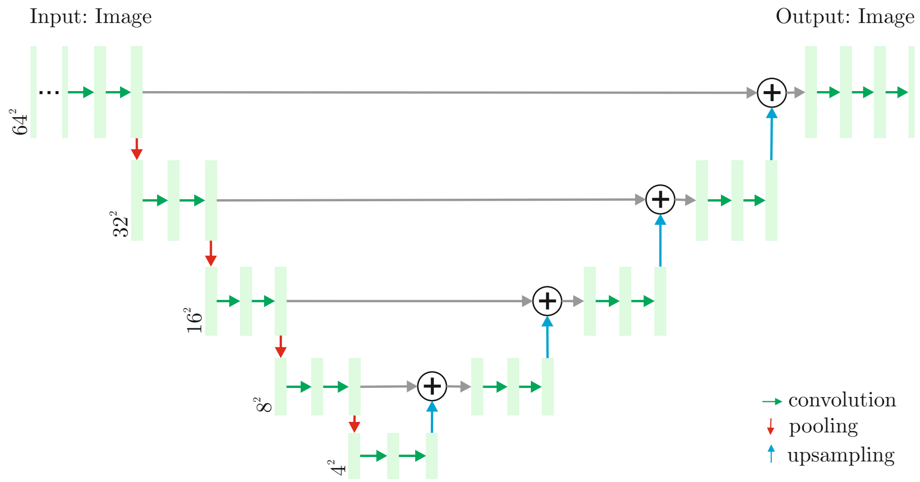


Figure 3.11: A visualization of the U-Net [72] which represents an image-to-image neural network with skip connections.

3.6 Generative Adversarial Networks

GANs represent generative modelling based approaches as exlained in Section 3.1.2 implemented as neural networks that have received a lot of attention due to their good performance and wide range of applications since their introduction in [26]. While the original formulation of *GANs* showed to be hard to train due to instability issues, many of these problems have been overcome by introducing Wasserstein Generative Adversarial Network (WGAN) in [6]. The work in [28] further improved *WGANs* by proposing to use a gradient penalty term yielding Wasserstein Generative Adversarial Network with Gradient Penalty (WGAN-GP).

A *GAN* consists of two differentiable neural networks, namely the generator G and the discriminator D, that play a game theory based minimax game against one another. While the discriminator D is defined as a encoder network similarly to the network described in Section 3.5.1.1, the generator G is represented by a decoder network as in Section 3.5.2. Considering a real data distribution $\mathbb{P}_r$ and following the original formulation as introduced in [26], the generator G now receives random noise z drawn from a distribution $p(z)$ and learns to generate new data samples $\hat{y}$ from z, see Figure 3.12. Defining the generated *synthetic* data distribution $\mathbb{P}_g$ of the new data samples $\hat{y}$ leads to the following formulation

$$\hat{y} \sim \mathbb{P}_g, \hat{y} = G(z), z \sim p(z), \tag{3.25}$$

where the distribution $p(z)$ represents an arbitrary simple distribution like a uniform or Gaussian distribution. To accomplish this, the generator G optimizes its parameters $\boldsymbol{\theta}_G$

to learn a transformation from $p(z)$ to a synthetic data distribution $\mathbb{P}_g$ with the goal that $\mathbb{P}_g$ is as similar as possible and ideally equal to $\mathbb{P}_r$.

The discriminator D is trained to distinguish between real data samples y drawn from $\mathbb{P}_r$ and synthetic data samples $\hat{y}$ drawn from $\mathbb{P}_g$ optimizing the parameters $\boldsymbol{\theta}_D$ of D, see Figure 3.12. Since D yields the probability of either y or $\hat{y}$ belonging to the real data distribution $\mathbb{P}_r$, it ideally manages to result in $D(\hat{y}) = 0$ and $D(y) = 1$. This formulation leads to the following minimax game

$$\min_G \max_D v(D, G) = \mathbb{E}_{y \sim \mathbb{P}_r}[\log D(y)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))], \tag{3.26}$$

where the payoff function $v(D, G)$ is evaluated to alternately update $\boldsymbol{\theta}_D$ and $\boldsymbol{\theta}_G$. While the discriminator is incentivized to optimize the correct classification of the given data into real and synthetic samples, the generator tries to confuse the discriminator by making him believe that the generated samples are real. The optimal generator of a *GAN* is defined as such, that the discriminator's output yields $\frac{1}{2}$ for both, y and $\hat{y}$.

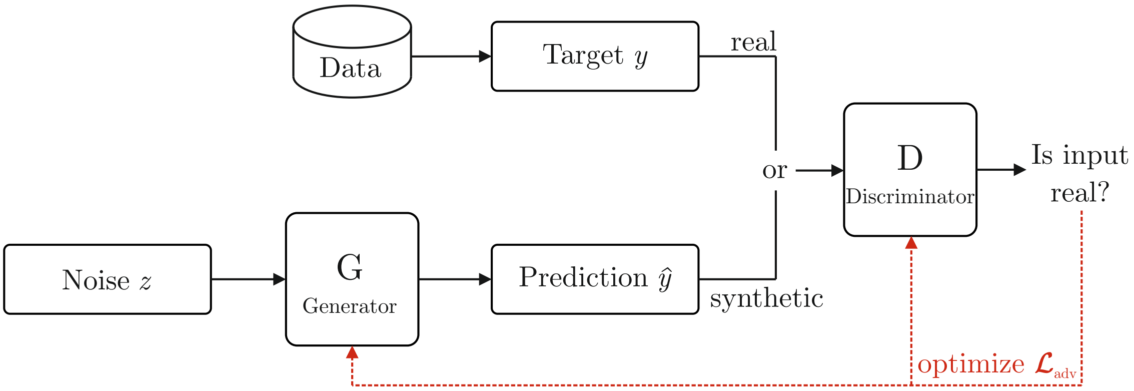


Figure 3.12: A schematic visualization of a *GAN* according to [26].

However, while *GANs* are guaranteed to converge when $v(D, G)$ maximized in respect to D is convex [25], in practice this condition is typically not met resulting in severe difficulties when training *GANs* as initially introduced. More precisely, the optimization of a *GAN* requires to find a Nash equilibrium, where complementing optimal strategies need to be found as such, that neither the discriminator nor the generator has anything to gain by only changing it's own strategy [61]. Finding a Nash equilibrium is difficult and typically results in the situation that one network heavily outperforms the other leading to an unstable training process that causes suboptimal results. While this problem can be circumvented by carefully tuning the hyperparameters of the *GAN* and finding a set of parameters that yields good results for a given task and data, this is a cumbersome process, since it is still required to find a Nash equilibrium. Another prominent problem identified in [26] is represented by mode collapse, where the generator G only learns to map multiple different inputs to the same output and produces very similar samples of $\hat{y}$

instead of a broad variety of samples, which is mainly caused by independent evaluation of $\hat{y}$ in a minibatch by the discriminator [75].

To improve the stability issues and problems like mode collapse present in traditional *GANs*, the work in [6] provided a comprehensive analysis of popular probability distances and convergences when learning distributions. Learning a probability distribution $\mathbb{P}_g$ that is as similar as possible to the real data distribution $\mathbb{P}_r$ is classically achieved by maximizing the likelihood estimation of a probability density $p_g(y)$ on the real data y. This problem statement can then be reformulated by transforming the maximum likelihood estimation into minimizing the Kullback-Leibler (KL) divergence defined as

$$KL(\mathbb{P}_r \parallel \mathbb{P}_g) = \int \log \left(\frac{p_r(y)}{p_g(y)} \right) \cdot p_r(y) \, \mathrm{d}\mu(y), \tag{3.27}$$

where it is assumed that both $\mathbb{P}_r$ and $\mathbb{P}_g$ admit a density with respect to the same measure μ [6]. As such, the *KL* divergence requires that the density $p_r(y)$ exists, however, this is not guaranteed especially in the common case when the distributions are supported by low dimensional manifolds. This leads to no intersection or just a negligible intersection between the model distribution and the support of the true distribution resulting in the *KL* divergence to be undefined (or infinite).

A typical solution that circumvents the problem of the *KL* divergence being undefined is implemented by many generative model based approaches by adding noise sampled e.g. from a Gaussian distribution to the model distribution. However, while this method is sufficient to prevent the undefined states when minimizing the *KL* divergence, the additive noise consequently also leads to a blurred and thus, degraded quality of the generated samples. An advantage of a generator neural network G of a *GANs* is, that G represents a parametric function that directly generates samples following a distribution $\mathbb{P}_g$ from random noise z. Since the generator G is a parametric function, updating the parameters $\boldsymbol{\theta}_G$ of G directly influences the generator's distribution $\mathbb{P}_g$, which is sufficient to bring $\mathbb{P}_g$ closer to $\mathbb{P}_r$. This eliminates the assumption of the *KL* divergence that both $\mathbb{P}_r$ and $\mathbb{P}_g$ admit a density and as such, it is not mandatory that the density p_r exists. Furthermore, since the densities p_r and p_g are not required to optimize the generator, it is irrelevant whether the densities are only supported by low dimensional manifolds or not. Thus, adding noise to the model distribution on which many generative model based approaches rely on is not necessary [6].

GANs as originially introduced in [26] used the Jensen-Shannon (JS) divergence to measure the distance between the real data distribution $\mathbb{P}_r$ and the generated data distribution $\mathbb{P}_g$ defined as

$$JS(\mathbb{P}_r, \mathbb{P}_g) = KL(\mathbb{P}_r \parallel \mathbb{P}_m) + KL(\mathbb{P}_g \parallel \mathbb{P}_m), \tag{3.28}$$

where $\mathbb{P}_m = \frac{\mathbb{P}_r + \mathbb{P}_g}{2}$. Similarly to the *KL* divergence, the *JS* divergence also suffers when learning distributions that are supported by low dimensional manifolds as shown in [6].

3.6.1 Wasserstein Generative Adversarial Networks

Motivated by analysing the properties of different popular probability distance metrics, the work proposed in [6] found that the Earth Mover (EM) distance or Wasserstein-1 distance is able to solve problems like the instability issues and mode collapse that are present in traditional *GANs* and introduced *WGANs*. The *EM* distance is defined as

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{y, \hat{y} \sim \gamma}[\| \, y - \hat{y} \, \|], \tag{3.29}$$

where $\Pi(\mathbb{P}_r, \mathbb{P}_g)$ defines the set of all joint distributions $\gamma(y, \hat{y})$ whose marginals are $\mathbb{P}_r$ and $\mathbb{P}_g$ respectively, resembling the cost of the optimal transport plan to transform the distribution $\mathbb{P}_r$ into $\mathbb{P}_g$. While the infimum in Equation (3.29) is highly intractable, the Kantorovich-Rubinstein duality [93] allows to reformulate the optimization scheme as

$$W(\mathbb{P}_r, \mathbb{P}_g) = \sup_{||f||_L \leq 1} \mathbb{E}_{y \sim \mathbb{P}_r}[f(y)] - \mathbb{E}_{\hat{y} \sim \mathbb{P}_g}[f(\hat{y})], \tag{3.30}$$

where the supremum is over all 1-Lipschitz functions. Next, the work in [6] suggests to replace the 1-Lipschitz functions by K-Lipschitz functions given some constant K yielding $K \cdot W(\mathbb{P}_r, \mathbb{P}_g)$, where K is swallowed by the learning rate and the function space f transforms into a parameterized family of functions f_w with $w \in \mathcal{W}$ that are all K-Lipschitz. This parameterized family of functions f_w can then be replaced by our discriminator neural network D with the parameters θ_D which finally allows us to derive the following objective function

$$\max_D \mathbb{E}_{y \sim \mathbb{P}_r}[D(y)] - \mathbb{E}_{\hat{y} \sim \mathbb{P}_g}[D(\hat{y})]. \tag{3.31}$$

A reformulation is possible by following Equation (3.25) and defining $\hat{y} = G(z)$ sampled from the distribution $\mathbb{P}_g$, where $\hat{y}$ is generated by the generator network G given an input noise z sampled from a noise distribution $p(z)$ such that $\mathbb{E}_{\hat{y} \sim \mathbb{P}_g}[\hat{y}] = \mathbb{E}_{z \sim p(z)}[G(z)]$. This allows to define the objective value loss function $v(D, G)$ of the *WGAN* as

$$\min_G \max_D v(D, G) = \mathbb{E}_{y \sim \mathbb{P}_r}[D(y)] - \mathbb{E}_{z \sim p(z)}[D(G(z))]. \tag{3.32}$$

Since the parameters θ_D are defined to lie in a compact space $\mathcal{W}$, backpropagation leads to estimating $\mathbb{E}_{z \sim p(z)}[\nabla D(G(z))]$ resulting in a training scheme similar to the one proposed in the original *GAN* in [26]. As the last step, it is now necessary to define the compact space $\mathcal{W}$ which then implies that all functions D will ultimatly be K-Lipschitz and the stated assumption holding true. It is suggested in [6] to clip the weights in θ_D to a fixed box defined as $\mathcal{W} = [-0.01, 0.01]$ after each gradient update to yield a compact space.

While for original *GANs* it is important to balance the progressive improvement of the discriminator D and the generator G as such, that neither is able to outperform the other, this is not necessary when training *WGANs*, since the *EM* distance is still able to provide gradients in contrast to the *JS* divergence. Furthermore, the *EM* distance does not rely on the existence of the density p_r, which represents a limitation of the *KL* and *JS* divergence as explained in Section 3.6. As such, the *EM* distance allows to train the discriminator D to optimality before updating the generator G as recommended in [6], which yields more reliable gradients and makes the training more stable. An additional advantage of an optimal discriminator D is, that the loss function becomes interpretable and correlates to the quality of the generated images in comparison to the real data.

3.6.2 Wasserstein Generative Adversarial Networks with Gradient Penalty

WGAN solved important issues present in the orginial *GAN* formulation [26] by using the *EM* or Wasserstein-1 distance as a value function. However, their formulation requires the discriminator's parameters θ_D to lie in a compact space $\mathcal{W}$ of 1-Lipschitz functions, which they define by clipping the weights of θ_D to a fixed box defined as $\mathcal{W} = [-0.01, 0.01]$. The work in [28] argues that weight clipping is a suboptimal solution potentially leading to non-convergence or unstable gradients of *WGANs* and prove that the optimal discriminator has a unit gradient norm almost everywhere under $\mathbb{P}_r$ and $\mathbb{P}_g$, since the discriminator D is defined to be a 1-Lipschitz function. To enforce the 1-Lipschitz constraint and the unit gradient norm on θ_D, [28] suggests to constrain the gradient norm of the discriminator's output by it's input using a soft two-sided gradient penalty which encourages the norm of the gradient to go towards 1. Following this formulation and modifying the value function of Equation (3.32) accordingly leads to *WGAN-GP* defined as

$$\min_G \max_D v(D,G) = \mathbb{E}_{y\sim\mathbb{P}_r}[D(y)] - \mathbb{E}_{z\sim p(z)}[D(G(z))] - \gamma \cdot \mathbb{E}_{\tilde{y}\sim\mathbb{P}_{\tilde{y}}}[(||\nabla_{\tilde{y}} D(\tilde{y})||_2 - 1)^2], \quad (3.33)$$

where $\mathbb{P}_{\tilde{y}}$ is implicitly defined by uniformly sampling along straight lines between pairs of points sampled from the real data distribution $\mathbb{P}_r$ and the generated data distribution $\mathbb{P}_g$, which they motivate via the optimal discriminator that connects coupled points from these two distributions. The term γ represents the penalty coefficient which was found to work well when set to 10 across a variety of experiments [28]. *WGAN-GP* does not work if batch normalization is used, since it would lead to an invalid gradient penalty, however, [28] argues that their method works with normalization schemes that do not introduce correlations between samples and recommend to use layer normalization [7] instead of batch normalization.

The introduction of the gradient penalty term in [28] allowed to make the training of *WGANs* even more stable. *WGAN-GP* oftenly shows a good performance by using the default parameters as proposed in [28] without the need of additional hyperparameter tuning simplifying the application of *WGAN-GP* independently of the task and the data.

3.6.3 Extending Generative Adversarial Networks with a Prior

While *GANs* and especially the extension *WGAN-GP* as explained in Section 3.6.2 show great potential for data generation from a random noise distribution, they can also be modified to solve other tasks. More precisely, if there exists a prior x with some meaning that is drawn from a distribution $\mathbb{P}_x$ as such that $x \sim \mathbb{P}_x$, this prior x can then be used to replace the random noise $z \sim p(z)$ as it is defined in the original *GAN* [26]. Instead of learning a transformation from z to $\hat{y}$, the generator G can now be optimized to learn a transformation from the prior x to $\hat{y}$ with $\hat{y}$ forming the data distribution $\mathbb{P}_g$, see Figure 3.13. The generator G still aims to bring the generated data distribution $\mathbb{P}_g$ as close as possible to the real data distribution $\mathbb{P}_r$ as explained in the previous sections. However, in contrast to the random noise distribution, this prior constrains the generator's input distribution since it contains some meaning which ultimately simplifies the problem for the generator G to learn a transformation to $\mathbb{P}_g$ aiming for $\mathbb{P}_g$ to be as similar as possible to $\mathbb{P}_r$. By incorporating that prior, the original formulation of *GAN* [26] as given in Equation (3.26) can now be modified to

$$\min_G \max_D v(D, G) = \mathbb{E}_{y\sim\mathbb{P}_r}[\log D(y)] + \mathbb{E}_{x\sim\mathbb{P}_x}[\log(1 - D(G(x)))], \tag{3.34}$$

where the random noise distribution $p(z)$ is replaced by the prior $\mathbb{P}_x$. Following this reformulation, the definition of *WGAN-GP* as introduced in [28] given in Equation (3.33) can also be modified accordingly, yielding

$$\min_G \max_D v(D, G) = \mathbb{E}_{y\sim\mathbb{P}_r}[D(y)] - \mathbb{E}_{x\sim\mathbb{P}_x}[D(G(x))] - \gamma \cdot \mathbb{E}_{\tilde{y}\sim\mathbb{P}_{\tilde{y}}}[(||\nabla_{\tilde{y}} D(\tilde{y})||_2 - 1)^2], \tag{3.35}$$

where $\mathbb{P}_{\tilde{y}}$ is now defined as uniformly sampled along straight lines connecting the real data distribution $\mathbb{P}_r$ and the generated data distribution $\mathbb{P}_g$.

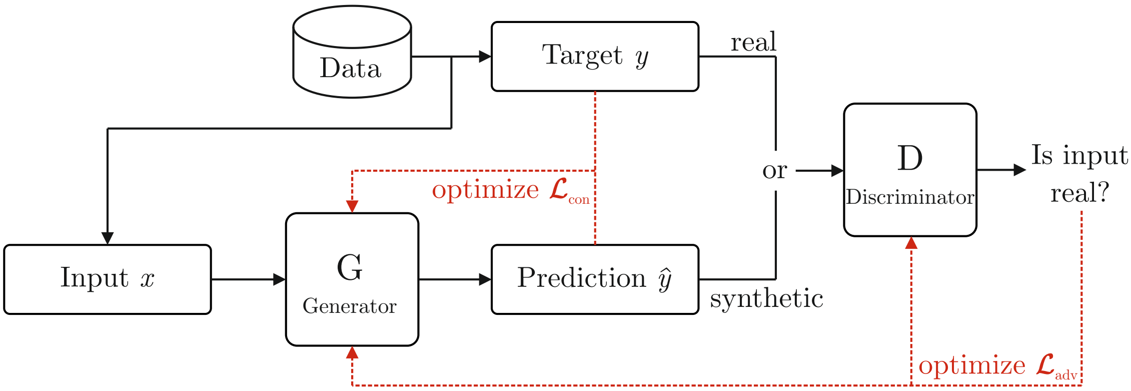


Figure 3.13: A schematic visualization of an extension to *GANs* that allows to use an additional loss function.

While this prior-incorporating formulation of *GANs* can be used to solve tasks like super resolution, domain transfer, image denoising and image reconstruction, they still need to be further constrained to improve their performance as shown e.g. in [47, 100]. This additional constraint can be accomplished by combining the adversarial loss of the *GAN* with a conventional content loss function like $\mathcal{L}_1$ loss or $\mathcal{L}_2$ loss, see Equation (3.10) and Equation (3.11) respectively, see Figure 3.13. This was done e.g. in the work in [47], which applied the adapted *GAN* formulation given in Equation (3.34) to a 4× super resolution task and achieved state-of-the-art results by combining the adversarial loss with a content loss evaluating different content loss functions. Another approach in [100] evaluated the performance of *WGAN-GP* using a prior similarly to Equation (3.35) for Computed Tomography (CT) image denoising and experimented with different combined loss function aiming to improve the quality of simulated quarter-dose *CT* images to be similar to normal-dose *CT* images explained in more detail in Section 4.1.2.

3.7 Training Neural Networks

While we have discussed deep learning in this chapter and explained how the performance of neural networks is iteratively improved in Section 3.3, we have only optimized the neural network towards the data used during training called the training set. Considering the training set X_{train} consisting of a usually small subset of all possible valid samples X defined as $X_{train} \subsetneq X$, we aimed to train an algorithm A to learn a data distribution $\mathbb{P}_A$ from X_{train} that shall be as similar as possible to the real data distribution $\mathbb{P}_r$ representative for all samples in X. So, in the ideal case of $\mathbb{P}_A$ being equal to $\mathbb{P}_r$, our learning algorithm A is able to perform as good on samples seen during training X_{train} as on samples not seen during training $X \setminus X_{train}$, which is practically very unlikely. The validation of how good an algorithm A is generalizing to data unseen during training is typically accomplished by introducing a test set X_{test}, which represents another subset of all possible valid samples X disjoint of X_{train} defined as $X_{test} \subsetneq X \setminus X_{train}$ [25]. Generalization of A to unseen data is then expressed as the performance of A on X_{test}.

3.7.1 Underfitting and Overfitting

Two problems can be observed when training an algorithm A on a training set X_{train}, namely underfitting and overfitting of A to X_{train}. Underfitting means that A is unable to learn a meaningful representation of the training data X_{train}, which, can sufficiently be solved by increasing the number of parameters $\boldsymbol{\theta}$ learned by A directly enabling it to learn more complex data distributions [25].

Overfitting on the other hand means that A is too powerful allowing it to learn too specific features of individual data samples in X_{train} and leads to A starting to memorize the distribution of these individual samples by heart rather than learning a general distribution of the data similar to $\mathbb{P}_r$ [25]. As such, overfitting results in A performing significantly better on the samples X_{train} used during training than on unseen samples

$X \setminus X_{train}$ and makes it unable to generalize.

3.7.2 Early Stopping

The observation of the performance of A with a high enough number of parameters $\boldsymbol{\theta}$ on X_{train} compared to X_{test} during training typically shows, that while the training loss on X_{train} is continuously decreasing, the loss on X_{test} initially decreases but starts to increase at some point again. The parameters $\boldsymbol{\theta}$ at the point where the loss of A on X_{test} is minimal, is considered as the optimal parameter setup of A that best generalizes to unseen data. As such, early stopping defines a regularization scheme aiming to find the optimal $\boldsymbol{\theta}$ to minimize the loss on the test data [25].

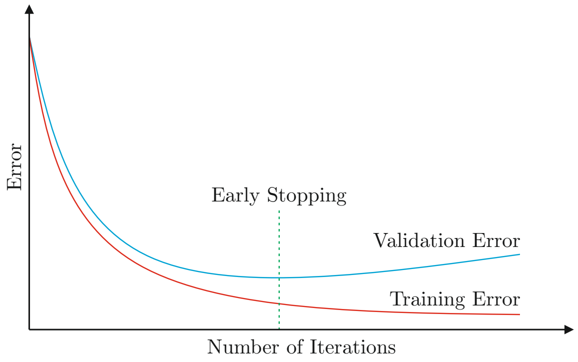


Figure 3.14: A visualization of early stopping.

3.7.3 Data Augmentation

For some problems in computer vision there are huge public datasets[3] available, e.g. the ImageNet[4] used for recognition tasks containing over 14 million annotated images. However, in other domains like medical imaging, one problem that is typically encountered in deep learning is the sparse amount of data publicly available due to ethical reasons and acquisition cost. A very small training set X_{train} simplifies memorization of the individual data samples of an algorithm A and can lead to A overfitting to X_{train} more easily. While this results in serious problems of A to generalize, it is rather simple to counter this problem by augmenting the training data before passing it to A.

Data augmentation defines the process of generating artificial data samples by introducing some variance into the available data samples and increase the number of unique samples available during training. As shown e.g. in [19, 46, 84], data augmentation is an effective method to improve the performance of a learning based method significantly.

---

[3] A list of publicly available datasets in computer vision: `https://github.com/jbhuang0604/awesome-computer-vision/#datasets`

[4] The well-known ImageNet: `http://image-net.org/`

Furthermore, most applications augment data on the fly, which in theory allows for an *infinite* amount of unique data samples without a memory overhead.

While data augmentation is usually done by simple global operations like translation or intensity shift, also more sophisticated methods as e.g. elastic deformation exist. Data augmentation is useful and beneficial for a *CNN*, when the variance introduced to the data is natural and can potentially occur as such. Considering *CT* imgaging, translations up to some degree can occur in the data, while a variation of intensity values is mostly prevented by the normalization to the Hounsfield Unit (HU) scale which is explained in Section 2.4.4. Large rotations are typically also not the case, since orientation and coordinates are defined by the procedure, i.e. the setup of the device and the position and orientation of the patient during image acquisition. As such, only a slight rotation in each dimension will be required to be utilized for augmentation.

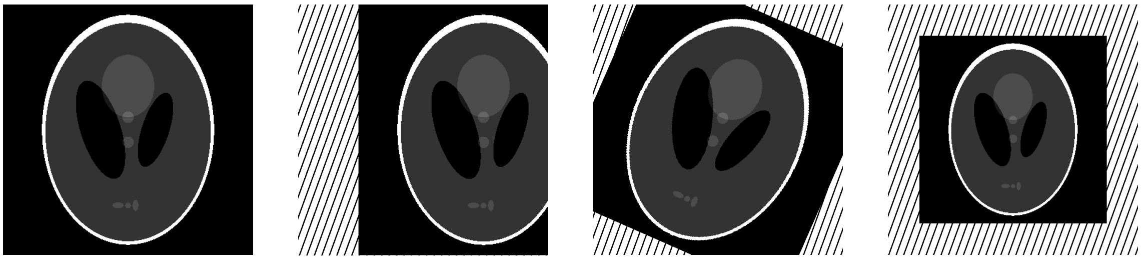

(a) Original **(b)** Translation **(c)** Rotation **(d)** Scaling

Figure 3.15: A visualization of different data augmentation operations. Image (a) shows the original image, (b) shows a translation, (c) shows a rotation, (d) shows scaling augmentation.

However, while other data augmentation methods exist, we will only focus on the ones that were actually used in this work, which are translation, rotation and scale. In comparison to the not augmented target image shown in Figure 3.15a, by utilizing a random translation the image is shifted e.g. as shown in Figure 3.15b. Random rotation is done by rotating the image around it's center, an example is given in Figure 3.15c. A random scale is performed to expand or contract the image in the respective dimension, see e.g. Figure 3.15d.

Furthermore, these image augmentation modalities potentially allow image areas previously located outside of the image's boundaries to be visible after augmentation. For that purpose, different boundary handling methods are in existence to define these newly introduced image areas, however, for our application we defined these newly introduced image areas as air according to the *HU* scale, which defines them as *empty*.

4

Related Work

Contents

4.1 Different Strategies to Apply Deep Learning 50
4.2 Tube Current Reduction based Approaches 53
4.3 Beam Blocking based Approaches 55
4.4 Sparse-View based Approaches 57
4.5 Accelerated Magnetic Resonance Imaging 60

This chapter is dedicated to recently published work that contributes to the field of medical imaging and especially Computed Tomography (CT) image reconstruction of which many are motivated by reducing the amount of ionizing radiation exposed to the patient. Due to the breakthroughs of machine learning in general and deep learning for imaging applications which is explained in Chapter 3, many of the approaches addressed in this chapter are based on Convolutional Neural Networks (CNNs). And since there are different strategies to apply deep learning and solve the task of *CT* image reconstruction, we will first explain these different strategies in Section 4.1. The subsequent sections are separated into groups of similar approaches that reduce the ionizing radiation dose used during image acquisition and learn to reconstruct from that reduced set of available data. One group of approaches to reduce the radiation dose is the reduction of the tube current explained in Section 4.2. A second group of approaches relies on physical beam blockers which are described in Section 4.3 and another group reduces the number of views acquired and utilized for image reconstruction given in Section 4.4. Lastly, Section 4.5 summarizes some contributions to the Magnetic Resonance Imaging (MRI) community where the goal is to improve the quality of accelerated *MRI* images, which represents a similar problem to low-dose *CT* image reconstruction.

4.1 Different Strategies to Apply Deep Learning

Image reconstruction in *CT* is done by first acquiring a set of projection views as described in Section 2.4 yielding the sinogram, which contains the projection data used to reconstruct a *CT* image that resembles the patient, see Section 2.5. Reducing the radiation dose used to reconstruct a *CT* image consequently also reduces the information contained in the sinogram, which leads to a degraded reconstruction quality when classical methods are used for *CT* image reconstruction as explained in Section 2.6. The problem of improving the reconstruction quality from a low-dose sinogram was in the recent years typically tackled by deep learning based methods and especially *CNNs*, see Section 3.2. The advent of deep learning to the field of medical imaging also introduced a number of different strategies to apply deep learning and solve the task of *CT* image reconstruction from a low-dose sinogram which will be explained in this section.

We differentiate between these strategies depending on the stage at which a *CNN* is applied to solve the reconstruction problem of which a subset of strategies have been observed in [51] for sparse-view *CT* reconstruction, however, following a broader definition, these strategies can also be applied to other low-dose reconstruction approaches. More precisely, the observed strategies found in literature include a strategy that learns sinogram-to-sinogram reconstruction which is explained in Section 4.1.1 and another strategy that is optimized in image-to-image reconstruction, see Section 4.1.2. A third strategy represents a sequential combination of both independently training two *CNNs* as described in Section 4.1.3. All of the forementioned strategies rely on classical reconstruction methods like the Filtered Backprojection (FBP) method, which we see as a limitation that needs to be addressed. As such, the method proposed in this thesis follows a forth strategy, namely the strategy of learning sinogram-to-image reconstruction, where a *CNN* directly learns the *CT* image reconstruction from the low-dose sinogram without the need of utilizing a classical reconstruction method as explained in Section 4.1.4.

4.1.1 Sinogram-to-Sinogram Reconstruction

One reconstruction strategy that can be found in literature is represented by sinogram-to-sinogram reconstruction, where a *CNN* f_S is applied to learn the mapping from a low-dose sinogram x_S to a improved low-dose sinogram y_S that is similar to a normal-dose sinogram. The improved low-dose sinogram y_S is then utilized as input for the *FBP* method – or any other classical reconstruction technique – which finally yields the reconstructed *CT* image y_I, see Figure 4.1. This strategy can formally be expressed as

$$\min_{\boldsymbol{\theta}}(f_S(x_S;\boldsymbol{\theta}) - y_S), \tag{4.1}$$

where the parameters $\boldsymbol{\theta}$ of the *CNN* f_S are optimized to approximate the optimal parameters. The strategy of learning sinogram-to-sinogram reconstruction represents a learned preprocessing step of a classical *CT* image reconstruction method and was found to yield a bad result compared to other strategies [51].

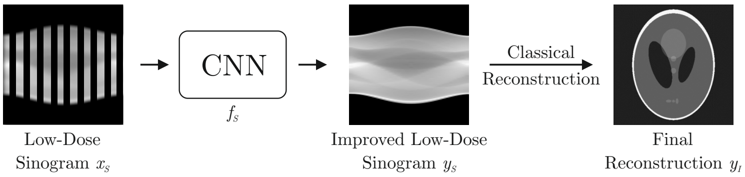


Figure 4.1: A visualization of a sinogram-to-sinogram reconstruction procedure.

4.1.2 Image-to-Image Reconstruction

Learning image-to-image reconstruction is similar to the problem of image denoising or image enhancement, where the low-dose sinogram x_S is directly used as an input to a classical reconstruction method yielding an intermediate low-dose *CT* image x_I. As a next step, this strategy now applies a *CNN* f_I to improve the quality of the low-dose *CT* image x_I to be similar to a normal-dose *CT* image yielding the final reconstruction y_I as shown in Figure 4.2. Optimization of the parameters $\boldsymbol{\theta}$ towards the optimal parameters is formally expressed as

$$\min_{\boldsymbol{\theta}}(f_I(x_I;\boldsymbol{\theta}) - y_I). \tag{4.2}$$

The strategy of image-to-image reconstruction can be seen as a learned postprocessing step and was found to yield decent reconstruction results in [51]. This strategy is widely used in research nowadays due to it's decent results and simplicity, since access to the projection data – which is usually not available – is not required by the *CNN*. However, a limitation of this method is represented by the dependency on a classical reconstruction method, since the *CNN* learns to remove the artifacts that have been introduced by this classical reconstruction method.

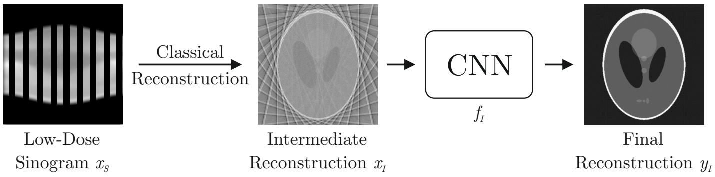


Figure 4.2: A visualization of an image-to-image reconstruction procedure.

4.1.3 Sinogram-to-Sinogram Reconstruction and then Image-to-Image Reconstruction

A combination of both, sinogram-to-sinogram reconstruction and then image-to-image reconstruction as visualized in Figure 4.3 represents another approach, where two independent *CNNs* are trained and sequentially executed to obtain the final reconstruction result. This approach as it is observed in [51] first utilizes a learned preprocessing step, where a *CNN* f_S is optimized according to Equation (4.1) as explained in Section 4.1.1, which results in an improved low-dose sinogram y_S. This improved low-dose sinogram y_S is then used to reconstruct a *CT* image x_I using a classical reconstruction method like the *FBP*. Finally, the reconstructed *CT* image x_I is then used as input for a different *CNN* f_I which is optimized as in Equation (4.2) and results in the final reconstruction image y_I representing a learned postprocessing step as explained in Section 4.1.2. As such, two consecutive problems are independently optimized and concatenated by a classical reconstruction method, which we see as a limitation of this strategy.

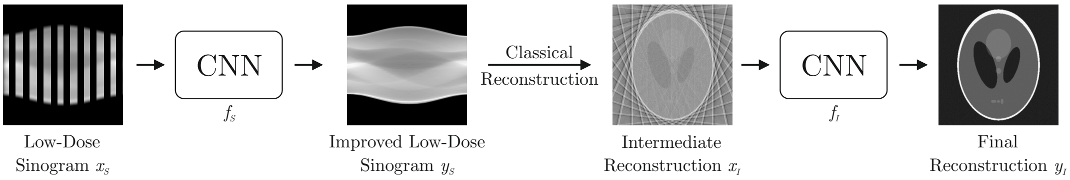

CNN
f_S
Classical Reconstruction
CNN
f_I
Low-Dose Sinogram x_S
Improved Low-Dose Sinogram y_S
Intermediate Reconstruction x_I
Final Reconstruction y_I

Figure 4.3: A visualization of a sinogram-to-sinogram reconstruction which is followed by an image-to-image reconstruction procedure.

4.1.4 Sinogram-to-Image Reconstruction

We assume that the forementioned strategies suffer from the utilization of a classical reconstruction method which is known to heavily suffer from artifacts when undersampled data is used for image reconstruction leading to suboptimal results as shown in Section 2.6. The strategy of learning sinogram-to-image reconstruction uses the low-dose sinogram x_S as an input to a *CNN* f_R which directly learns to reconstruct a final *CT* image y_I without relying on a classical reconstruction method, see Figure 4.4. This strategy can formally be expressed as

$$\min_{\boldsymbol{\theta}}(f_R(x_S;\boldsymbol{\theta}) - y_I), \tag{4.3}$$

where the parameters $\boldsymbol{\theta}$ are optimized as such, that they approximate the optimal parameters. By directly providing the projection data to the *CNN* it can access all the available information to reconstruct the *CT* image which bypasses the need of using a classical reconstruction method, while the *CNN* is still incentiviced to learn a function that is similar to classical reconstruction methods.

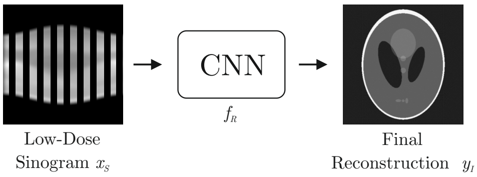


Figure 4.4: A visualization of a sinogram-to-image reconstruction procedure.

4.2 Tube Current Reduction based Approaches

In comparison to normal-dose *CT* imaging, this type of approaches uses a reduced tube current to acquire the projection images resulting in a low-dose *CT* image. By reducing the tube current, the amount of electrons flowing through the X-ray tube is reduced which consequently also reduces the amount of X-rays emitted by the X-ray tube. As such, reducing the tube current ultimately reduces the amount of ionizing radiation that is exposed to the patient, see Figure 4.5. However, this also means that less photons can be measured by the detector array, which leads to degraded signals and consequently also to a lower quality *CT* image. Approaches utilizing reduced tube current *CT* images try to improve the quality of these lower quality *CT* images as such, that they have a similar quality to normal-dose *CT* images.

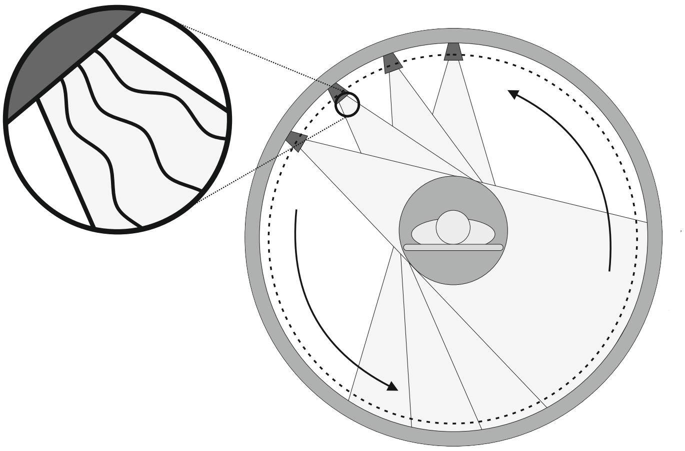


Figure 4.5: A schematic visualization of tube current based approaches.[1]

---

[1] Schematic visualization of *CT* imaging, adapted from `http://www.radtechonduty.com/2017/03/single-detector-row-ct-scan-systems.html`, last accessed on 22 March 2019.

4.2.1 Tube Current Reduced Approaches

The deep learning based approach in [101] implemented a sinogram enhancement method, utilizing pairs of low-dose and normal-dose *CT* projections to learn the mapping from a low-dose to a normal-dose *CT* projection. In contrast to that, the *CNN* utilized in [15] learns the mapping from a low-dose to a normal-dose *CT* image and thus, learns to remove artifacts introduced by the image reconstruction of the acquired low-dose projections, representing the method explained in Section 4.1.2. The follow-up work by the same authors in [14] replaced the *CNN* of their method by a residual *CNN* utilized as an encoder-decoder network which further improved their results. Differently to that, the work in [38] uses a *CNN* to suppress noise specific to low-dose *CT* images directly on the wavelet transform coefficients, as such, the low-dose reconstructed image is transformed into wavelet domain and the *CNN* learns a wavelet-to-wavelet optimization. Lastly, a K-sparse autoencoder was utilized in the work in [96] to train priors and iteratively solve *CT* image reconstruction and improve the quality of low-dose *CT* images.

As mentioned in Section 2.3.2, Cone Beam Computed Tomography (CBCT) imaging is able to reduce the amount of radiaton exposure in comparison to Fan Beam Computed Tomography (FBCT), however, the *CBCT* acquired images typically suffer from a reduced contrast leading to a degraded image quality. As such, the work in [39] reduced the amount of radiation by utilizing *CBCT* imaging and trained a *CNN* to transform the *CBCT* images into images similar to *FBCT* images. To accomplish this, the U-Net [72] based *CNN* was provided with pairs of *CBCT* and *FBCT* images of the same patient which are registered to one another. In contrast to the aforementioned method, the work in [102] aims to improve the visual quality of *CT* images acquired from lower resolution devices motivated by reduced costs as well as a possibly lower radiation exposure compared to higher resolution devices. As such, their approach represents a super-resolution technique based on a *CNN* to enhance low-dose *CT* images to achieve a higher resolution and a better image quality. The approach in [102] is evaluated as a single-slice as well as a multi-slice method, where the multi-slice method also takes the information of neighboring slices into account.

4.2.2 Approaches Using Generative Adversarial Networks

The work in [100] argues that *CNN* based methods that minimize $\mathcal{L}_2$ can compromise the visibility of important structural details which can be improved by utilizing a perceptual i.e. adversarial loss coming from a Generative Adversarial Network (GAN). More precisely, they utilized a Wasserstein Generative Adversarial Network (WGAN) to solve a tube current reduced problem by learning image-to-image reconstruction as explained in Section 4.1.2, where the generator's architecture is a simple sequential *CNN*. The evaluation in [100] includes some combinations of various loss functions amongst which are a content $\mathcal{L}_2$ loss, an adversarial $\mathcal{L}_{wGAN}$ loss and a pre-trained $\mathcal{L}_{VGG}$ loss, where the latter is coming from the Visual Geometry Group (VGG) network [85]. A similar approach was proposed in [95] which is also optimized for image-to-image reconstruction and solves the problem of tube current reduced *CT* image reconstruction by utilizing a combination of a content $\mathcal{L}_2$ loss and an adversarial $\mathcal{L}_{GAN}$ loss. The generator *CNN* is designed as a sequential network learning the noise pattern from a low-dose *CT* image which is in a final layer subtracted from the low-dose *CT* image yielding a denoised *CT* image similar to a normal-dose *CT* image. The evaluation compares the performance of the content loss, the adversarial loss and a combination of both.

4.3 Beam Blocking based Approaches

In beam blocking or Many-View Undersampling (MVUS) based approaches the *CT* itself is operated normally without a direct reduction of the radiation dose, where the utilized beam blockers act as physical barriers that block some of the X-rays from reaching the patient, see Figure 4.6. While beam blockers reduce the radiation dose received by the patient, they are most oftenly utilized in *CBCT* based approaches to reduce the amount of beam scatter and consequently increase the contrast of the reconstructed image. As explained in more detail in Section 2.3.2, normal dose *CBCT* already leads to a reduced ionizing radiation exposure in comparison to *FBCT* but suffers from a reduced reconstruction quality due to beam scatter. As such, beam blockers allow for an even further reduction of ionizing radiation exposure while also increasing the contrast of the reconstructed image in the case of *CBCT*. However, using beam blockers leads to missing information in every single projection posing a problem that is similar to inpainting and results in the introduction of artifacts when directly used for image reconstruction.

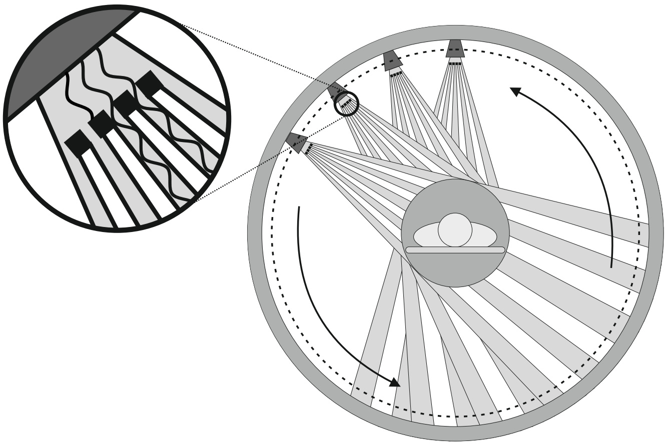


Figure 4.6: A schematic visualization of beam blocking based approaches.[2]

---

[2] Schematic visualization of *CT* imaging, adapted from `http://www.radtechonduty.com/2017/03/single-detector-row-ct-scan-systems.html`, last accessed on 22 March 2019.

4.3.1 Observations on Physical Properties of Beam Blockers

One of the most prominent questions related to beam blocking based approaches is about the physical properties of the utilized beam blockers, which is a subject to research on it's own. While variance in size, position and distance between single blockers as well as the field of view play a role, further distinction can be done by whether the beam blockers utilized are stationary or moving, where moving beam blockers usually represent more recent approaches. A study observing some variation in beam blockers in *CBCT* especially in respect to the field of view was conducted in [66] and found that a small field of view decreases the *CBCT*-typical beam scatter and reduces the radiation dose. The work in [49] observed moving beam blockers in *CBCT*, more specifically they tested the performance according to the number of slits each with a different reciprocation frequency, which was motivated by finding the configuration with the highest contrast to noise ratio. Another study was conducted in [16] observing the speed and design, i.e. the strip width and interspace between strips of a moving beam blocker in *CBCT*.

4.3.2 Beam Blocking in Cone-Beam Computed Tomography

The approaches in this category utilize beam blocking in *CBCT* imaging, where each approach exploits some additional information that is available due to more specific use cases. Furthermore, all of the following approaches rely on classical reconstruction methods without utilizing deep learning. The moving beam blocker based approach proposed in [104] utilizes a multi-view scatter correction method which uses adjacent views to estimate and correct scattering. As such, they exploit the information of the neighboring views which leads to better reconstruction results compared to not using this additional information. Differently to the previous method, the work in [105] sequentially gathers multiple partially beam blocked *CBCT* images introducing a temporal dimension that contains some redundant information. By defining the movement of the beam blocker and the respiratory motion asynchronous, the partially beam blocked and sequentially gathered information is not identical to the next captured sequence, which maximizes the benefit of exploiting the temporal dimension. Another moving beam blocker based approach was proposed in [65], where a *CBCT* and a Volumetric Modulated Arc Therapy (VMAT) image are acquired concurrently and for each, the information of the other is exploited to improve the quality of both, the *CBCT* and the *VMAT* image.

4.3.3 Beam Blocking in Fan-Beam Computed Tomography

Differently to the approaches described in Section 4.3.2, where beam blockers have been utilized to improve the quality of *CBCT* images, this section is dedicated to the underrepresented field of utilizing beam blockers in *FBCT* imaging. The approach in [59] utilizes small beam blockers they call high-resolution coded apertures in *FBCT* and investigate their use in combination with low-resolution detectors. By utilizing Compressed

Sensing (CS) they exploit the sparsity of the acquired data which allows to utilize the high-resolution coded apertures to achieve super-resolution from low-resolution detectors resulting in an improved image quality.

4.4 Sparse-View based Approaches

Instead of reducing the tube current or physically blocking some X-ray beams, sparse-view based approaches achieve a reduction of ionizing radiation exposure by limiting the number of X-ray projections acquired around the patient as shown in Figure 4.7. As such, sparse-view based approaches can easily be utilized and simulated, the former by simply reducing the number of acquired projection views and the latter by omitting some of the projections contained in a normal-dose sinogram. To maximize the entropy between the utilized X-ray projection views, they are typically acquired from equidistant positions around the patient. However, sparse-view *CT* also allows for some special cases like limited angle *CT*, where projections are only acquired within a limited range of angles, which can be restricted due to physical constraints of the scanned object or the *CT* scanner in use. Greatly reducing the number of projection views leads to streaking artifacts in the direction of single views, which is typically tackled in state-of-the-art approaches by a *CNN* that learns to reduce the amount of introduced artifacts.

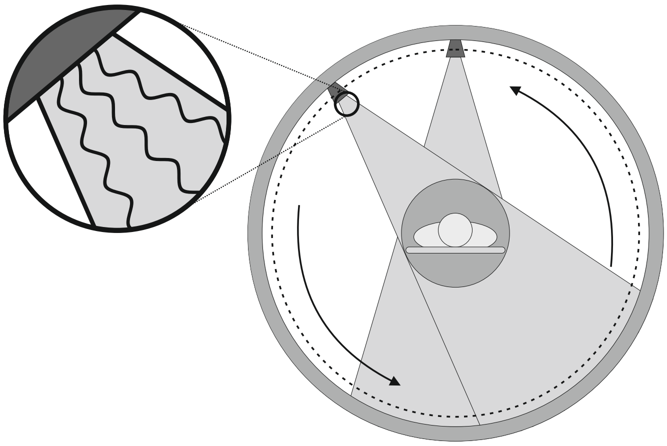


Figure 4.7: A schematic visualization of sparse-view based approaches.[3]

4.4.1 Sparse Sinogram-to-Sinogram Reconstruction based Approaches

The method proposed in [50] utilizes a sinogram-to-sinogram reconstruction based approach to fill in missing sinogram information due to sparse sampling. Their approach is

---

[3] Schematic visualization of *CT* imaging, adapted from `http://www.radtechonduty.com/2017/03/single-detector-row-ct-scan-systems.html`, last accessed on 22 March 2019.

based on sinusoid-like curve decomposition and eigenvector-guided interpolation with the goal of ensuring texture continuity of the inpainted sinogram. Differently to that, the work in [40] proposes a special use case sparse-view reconstruction method where multiple *CT* images with different kilovoltage peak (kVp) settings are reconstructed from just one *CT* scan. They propose a switching *kVp* system, where the *kVp* switches rapidly during gantry rotation resulting in individual sparse-view sinograms per *kVp* system each consisting of another set of views. Image reconstruction is improved by exploiting the additional information of the other sinograms by utilizing the in between angle sparse-view projections acquired using another *kVp* setting.

4.4.2 Image-to-Image Reconstruction based Approaches

The approaches in this section use already reconstructed low-dose *CT* images that have been generated using classical reconstruction methods and learn image-to-image reconstruction to improve the quality of these images following the definition given in Section 4.1.2. The approach in [37] uses a more general formulation and proposes a solution for inverse problems where the forward operator is a convolution, which includes problems like denoising and deconvolution as well as image enhancement in *CT* and *MRI*. For their *CT* evaluation they use the *FBP* method to reconstruct *CT* images from a sparse number of views, a U-Net [72] based *CNN* is then trained to map the sparse-view reconstructed *CT* image to a full-view reconstructed image. The work in [106] derived their problem formulation from dictionary learning, however, similar to the aforementioned work, the *CNN* utilized in [106] learned the mapping from a sparse-view to a full-view *CT* image. Instead of utilizing the *FBP* method for *CT* image reconstruction, they used the Algebraic Reconstruction Technique (ART) method and evaluated their method solely on synthetic images. Another approach was proposed in [98] which is based on an improved residual GoogLeNet [89] utilizing sparse-view *FBP* method reconstructed *CT* images. Differently to the other methods, the work in [98] learns the mapping from the reconstructed sparse-view *CT* image directly to an artifact image which is then subtracted from the sparse-view *CT* image to retrieve a denoised *CT* image. The method proposed in [29] represents an extension of [37] utilizing the same network architecture to train the *CNN* but where the *CNN* replaces the projector in a Projected Gradient Descent (PGD) method. Their method enforces the consistency of the reconstructed image to the available measurements and achieves state-of-the-art results.

Differently to the previous methods, the following ones utilize other learning based methods than *CNNs*. The approach in [36] utilizes a dictionary learning method applied in gradient domain instead of image domain, since it leads to sparser representations which reduces the required complexity of the learned dictionary. Their method utilizes a horizontal and a vertical gradient image generated from a sparse-view *ART* method reconstructed *CT* image, image recovery from gradient domain is done by solving the least-square method. A different approach proposed in [92] utilizes an iterative sparse-view and/or tube current reduced *CT* reconstruction method based on an adaptive edge-preserving Total Variation (TV) regularization. The introduced edge-preserving regularization term serves as a penalty for the *TV* regularization that reduces the amount of smoothing close to edges. Differently to that, the work in [63] combines the concept of a generalized *TV* regularization and a penalized weighted least-squares scheme to improve the quality of sparse-view reconstructed *CT* images. An important advantage of generalized *TV* over *TV* is the relieved piecewise constant assumption leading to a better reduction of artifacts and an improved preservation of structural details.

4.4.3 Sparse Sinogram-to-Image Learning based Approaches

The approaches described in the following used a sparse sinogram and directly learned to reconstruct a *CT* image without relying on any classical reconstruction method as explained in Section 4.1.4. The work in [67] represents an early approach to replace the *FBP* method by an Artificial Neural Network (ANN) to improve the reconstruction quality from a sparse sinogram, where they viewed the *ANN* as a combination of multiple *FBP* operations. While they provided early insights into using *ANNs* instead of the *FBP* method, they only worked with two dimensional (2D) phantom images and did not investigate *CNNs*. In contrast to that, the work in [17] proposed a learning based reaction diffusion model for image restoration which represents a learned regularizer that was extended in their follow-up work in [13] to solve the task of *CT* image reconstruction from sparse-view projection data. Their *CNN* is optimized using $\mathcal{L}_2$ loss and their architecture consists of a number of iteration-inspired layers. The work in [17] also inspired the work in [42], where they call their *CNN* a variational network. Their follow-up work proposed in [43] utilizes a variational network to reconstruct three dimensional (3D) *CT* images from either tube current reduced or sparse-view *CT* images. Since variational networks only require a small model size, they consequently also reduce the amount of data necessary to train the *CNN* which is advantageous. In contrast to the previous methods, the work in [2] is based on the well known primal-dual algorithm proposed in [12], where the proximal operators are replaced by a learned reconstruction operator trained by a *CNN*. The proposed method extends their previous work in [1] by increasing the complexity and flexibility of the *CNN* which alternately optimizes the primal and dual, i.e. the forward and back projection for a fixed number of iterations.

4.4.4 Limited Angle Computed Tomography

Limited angle *CT* represents a special case of sparse-view *CT* reconstruction. While the projection images are acquired from uniformly distributed angles and 360° around the object in default sparse-view *CT*, in limited angle *CT* the projection images are acquired from less than 360° due to physical restrictions of the object or the scanner. As such, limited angle *CT* also leads to a reduction of ionizing radiation exposure, however, limited angle *CT* leads to the introduction of a different kind of artifacts to the reconstruction result than non-limited angle *CT*.

The work in [97] extends the previous joint work on limited angle *CT* reconstruction in [31] to a *CBCT* geometry. By introducing a novel back projection layer it is possible to formulate the Feldkamp-Davis-Kress algorithm as a *CNN* which allows learning in both, projection and volume domain and consequently expands current post processing methods to benefit from projection data. A non-*CNN* limited angle *CT* reconstruction approach is proposed in [18], where an anisotropic *TV* based method is utilized to optimize the *ART* based reconstruction of the image. They show that the missing projection data leads to an imbalance between the anisotropic data fidelity constraint and the *TV* minimization, which can be improved by minimizing the sparsity of the image according to the scanning range.

4.4.5 Approaches Using Generative Adversarial Networks

The *CT* reconstruction method proposed in [5] optimizes sinogram-to-sinogram reconstruction as described in Section 4.1.1 and learns to complete a sparse sinogram to solve a limited angle problem. Their proposed method is composed of three stages: First, a generator *CNN* is used to reconstruct a *CT* image from the limited angle sinogram data, second, a full-view sinogram is generated from that *CT* image and finally, a well established classical reconstruction technique like the *FBP* method is used to reconstruct the final *CT* image from the generated full-view sinogram. The utilized generator *CNN* is optimized by an adversarial $\mathcal{L}_{GAN}$ loss based on a *GAN* as introduced in [26] and a content $\mathcal{L}_2$ loss.

4.5 Accelerated Magnetic Resonance Imaging

While *CT* faces the problem of ionizing radiation exposure to the patient, *MRI* opposes the problem of long data acquisition time necessary for a good reconstruction result which is explained in more detail in Section 2.1. This similarity between low-dose *CT* and accelerated *MRI* allows to easily interchange methods between these imaging modalities, which is the reason why this section is dedicated to contributions that have been made to the *MRI* community.

4.5.1 Typical Accelerated Magnetic Resonance Imaging Approaches

The work in [78] utilized a deep cascade of *CNNs* to improve the quality of a reconstructed *MRI* image from undersampled data. The utilized cascading network consists of multiple concatenated *CNNs* and intermediate data consistency terms yielding state-of-the-art results for heavily undersampled data. Another approach was proposed in [30], which uses variational networks, a combination of variational models with deep learning, to learn the reconstruction of accelerated *MRI* data and showed that the natural appearance and pathologies not present in the training set are preserved. The approach in [48] utilized deep residual learning composed of a separate magnitude and phase network that solves image reconstruction in framelet representation. This separation allows the utilization of data samples without having access to the full k-space data making it possible to pre-train the magnitude network on magnitude *MRI* images which serves as a postprocessing step due to the formulation. Differently to the aforementioned approaches, the work in [32] learned to improve the accelerated *MRI* reconstruction utilizing a domain adapted *CNN* to increase the amount of accessible data. This *CNN* was pre-trained on *CT* and synthetic radial *MRI* data, the domain adpation was conducted by fine-tuning the trained model using real radial *MRI* data.

4.5.2 Generative Adversarial Networks in Magnetic Resonance Imaging

The *MRI* reconstruction method proposed in [55] utilizes a *GAN* of which the generator receives a highly undersampled *MRI* image and learns to improve the quality of this image by reducing the contained artifacts. The generator's loss function consists of a least-square adversarial loss and a content $\mathcal{L}_1$ loss and the architecture of the generator is represented by a residual *CNN* that contains skip connections. Differently to that, the work in [81] proposes to use a two stage procedure which separates the content loss and the adversarial loss into two sequentially performed optimizations. While it is sensible to combine a content loss and an adversarial loss due to their complementary nature, they argue that the different training objectives of the two loss functions compete with each other which leads to the convergence to a suboptimal solution. The first stage consists of the reconstruction network that receives a reconstructed undersampled *MRI* image and has a sequential network architecture trained to optimize the $\mathcal{L}_2$ loss. The second stage is the visual refinement network, which uses the output of the reconstruction network and refines the quality of the *MRI* image utilizing a U-Net [72] based architecture which is optimized utilizing an adversarial $\mathcal{L}_{GAN}$ and a $\mathcal{L}_{VGG}$ [85] loss. Another *GAN* based approach is represented by the work in [99] which also utilizes a U-Net [72] based network architecture for the generator. To reduce the complexity of the model, the work in [99] proposes to use a refinement connection which adds the generator's input to the generator's output before calculating the loss. Their loss function combines an adversarial $\mathcal{L}_{GAN}$ loss and a content loss assembled from three parts, which are an image domain $\mathcal{L}_2$ loss, a frequency domain $\mathcal{L}_2$ loss and a $\mathcal{L}_{VGG}$ [85] loss retrieved through transfer learning.

5

Method

Contents

5.1 Preprocessing of the Data 64

5.2 On-the-fly Projection Data Generation 66

5.3 Learned Computed Tomography Reconstruction 70

5.4 Experimental Setup 74

This chapter is dedicated to the method we implemented to solve the demanding task of low-dose Computed Tomography (CT) reconstruction from undersampled data, which is motivated by reducing the amount of ionizing radiation exposure to the patient. The term undersampling refers to a sampling rate that violates the Nyquist-Shannon sampling theorem explained in Section 2.2.3 and is insufficient to correctly reconstruct a *CT* image yielding reconstruction results that are burdened by artifacts, especially when classical reconstruction methods are used for *CT* image reconstruction as described in Section 2.6. The success of deep learning and especially Convolutional Neural Networks (CNNs) explained in Section 3.2 lead to an increased interest in the domain of medical imaging and consequently also for medical image reconstruction, where *CNNs* are used to compensate the reduced amount of information by introducing a prior that was learned from data. Nowadays, many recently published approaches tackling the task of medical and *CT* image reconstruction from undersampled data rely on deep learning, Chapter 4 is dedicated to related work in this field.

To reduce the amount of ionizing radiation exposure, our method follows the definition of sparse-view *CT* reconstruction, where the number of projection views that are acquired and used for reconstruction is reduced as explained in Section 4.4. To circumvent the necessity of using a classical reconstruction method on which many recent reconstruction methods rely on, our method utilizes a *CNN* that directly learns to reconstruct a *CT* image from sparse projection data, i.e. the sparse sinogram as described in Section 4.1.4. Related work that is based on sparse-view *CT* reconstruction and also uses a *CNN* to directly learn *CT* image reconstruction from the sinogram is summarized in Section 4.4.3.

More precisely, we experimented with multiple setups of which our three dimensional (3D) pipeline uses *3D* data and network architectures, whereas our two dimensional (2D) pipeline respectively utilizes the data and network architectures in *2D*. While we optimize both pipelines using $\mathcal{L}_1$ loss, we also experimented with an additional adversarial $\mathcal{L}_{wGAN}$ loss for our *2D* pipeline, where $\mathcal{L}_{wGAN}$ comes from a Wasserstein Generative Adversarial Network (WGAN) as explained in Section 3.6. The essential differences between the *3D* and the *2D* pipeline as well as the differences between the *2D* pipeline without and with a *WGAN* are pointed out in the following subsections.

First, the preprocessing of the already reconstructed normal-dose *CT* images is described in Section 5.1. Since our approach requires not only normal-dose *CT* images but also projections of the *CT* images, we implemented a projection image generator explained in Section 5.2, which synthesizes projections and prepares them for further processing. Next, in Section 5.3, we give more insight into the core idea of our method and explain the *CNNs* we utilized. Finally, the experimental setup consisting of the material and the implementation details is given in Section 5.4.

5.1 Preprocessing of the Data

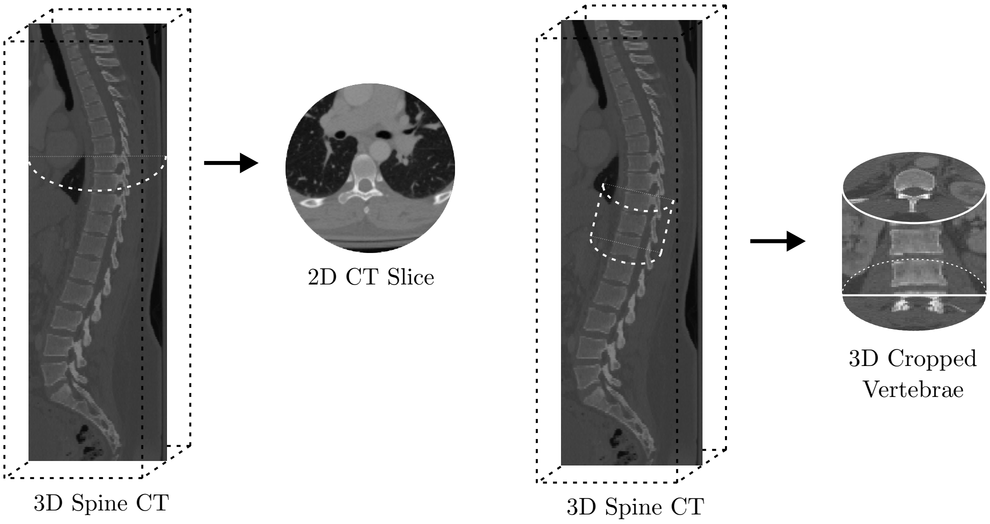
In this section we will give insight into the preprocessing of the available *3D* *CT* images which we employed in our method, detailed information on the used material is given in Section 5.4.1. While we used the same base material for all our experiments, the preprocessing of the data for our *3D* and *2D* pipeline is different. The data preprocessing for our *2D* pipeline is described in Section 5.1.1, while the data preprocessing for our *3D* pipeline is given in Section 5.1.2.

5.1.1 2D Computed Tomography Slice Extraction for our 2D Pipeline

For our *2D* pipeline we utilized *2D* *CT* slices in the axial plane see Figure 2.10, which we extracted from the *3D* *CT* images as shown in Figure 5.1a. Due to the reduced dimensionality it is feasible to train our *CNNs* with an image size of 128×128, which lead to a good resolution preserving small structural details. An additional advantage of using *2D* *CT* slices is that the number of samples increases from the number of *3D* *CT* images to the number of slices contained in these images, which leads to approximately 4.000 training and 1.000 test samples, see Section 5.4.1. The canonical position of our *2D* *CT* slice images is defined by the center position of the slice and the unchanged orientation of the *3D* *CT* image. The *2D* *CT* images are further processed according to this position.

5.1.2 3D Vertebra Extraction for our 3D Pipeline

For our *3D* pipeline we found a *CT* image size of $64 \times 64 \times 64$ to be the largest still feasible to work with. However, simply downsampling the full *CT* image to this rather small size



(a) *2D CT* slice extraction. **(b)** *3D CT* vertebra extraction.

Figure 5.1: Visualizations of the data extraction procedures. Image (a) shows the extraction of a *2D CT* slice, (b) shows the extraction of a *3D CT* vertebra.

leads to a huge amount of details lost in the downsampled *CT* image. Furthermore, the very small size of just eight training and two test *CT* images also represents a huge limitation even for highly self-similar medical images when utilizing a *CNN*. To solve both of these problems, we decided to crop individual *3D* vertebra images including their surroundings from the *3D CT* images and use these *3D CT* vertebra images as the training and test data for our *CNN*. First, cropping the smaller *3D* vertebra images from the full *3D CT* image leads to a higher remaining resolution when downsampling the images to a size of 64 × 64 × 64 with less details lost in these regions. Second, using the *3D CT* vertebra images increases the number of samples from the number of images to the number of vertebrae present in these images and as such, increases the total number of samples from 10 to 176 with a training and test split of 141 and 35 respectively. The individual vertebrae have a high self-similarity due to similar anatomical properties, while showing inter- as well as intra-subject variability which we aim to exploit.

Before cropping a *3D CT* vertebra image from the full *3D CT* image, the vertebra is brought into a canonical position. To calculate the canonical position of a vertebra, we utilize the segmentations that came with the dataset as explained in Section 5.4.1. First of all, the center of the canonical position of a vertebra is calculated as the center of mass of the respective vertebra's segmentation. The orientation of the vertebra is given by the relative position of the center of mass to the tip of the spinous process, which is the point that is farthest from the center of mass, and by the vertebra's relative position to the other vertebrae in the spine. After calculating the canonical position, the vertebra image is further processed relatively to that canonical position.

Lastly, considering the *3D* vertebra images as a stack of *2D CT* slices brings us to the analogy between our preprocessed *3D* and *2D* data as shown in Figure 5.2. This analogy allows us to simplify the explanations in the following sections to which we will refer accordingly.

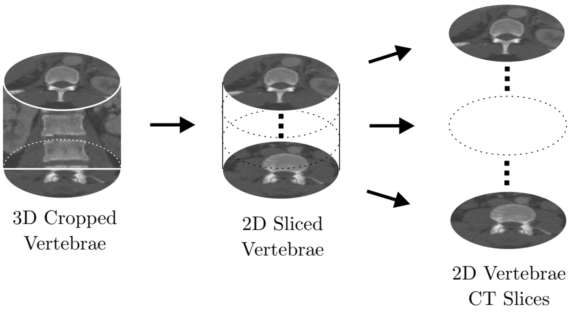


Figure 5.2: A visualization of the analogy between *3D* *CT* images and *2D* *CT* slices.

5.2 On-the-fly Projection Data Generation

As explained in Section 2.3, a *CT* scanner utilizes X-radiation to acquire the sinogram consisting of multiple forward projection views from different angles of a patient, which is then used to reconstruct the interior body structure of that patient. Our method uses the forward projection views contained in the sinogram and directly learns to reconstruct the *CT* image without the need of any classical reconstruction method as described in Section 4.1.4. To accomplish this, our method requires not only normal-dose reconstructed *CT* images but also the corresponding projection data used for *CT* image reconstruction, however, most available datasets do not incorporate the original projection data. Thus, we simulated the projection data from already reconstructed *CT* images by generating the forward projections from different angles on the axial plane of the *CT* images, similarly to the real forward procedure in *CT* imaging which is explained in Section 2.4. Simulating the projection images leads to important advantages: First, generating projection views at will also allows to augment the *CT* images before generating the forward projections at will, which maximizes the variability of the available data. Second, the angle from which the simulated projection views are generated can be chosen arbitrarily, while for real projection data the angles that can be used are limited to the angles that have been acquired during *CT* imaging. Lastly, simulating the projection data allows to generate projection views from cropped *CT* images which would be impossible otherwise.

Up to now we explained how we attain the *3D* vertebra respectively the *2D* *CT* slice images as well as the corresponding canonical position from the available material, see Section 5.1. As a next step, the canonical position is used to augment the training data samples explained in Section 5.2.1, which is followed by masking data samples as described in Section 5.2.2. The procedure to generate the forward projections is explained in Section 5.2.3 and the back projection to finally arrive at the input used for our *CNNs* is given in Section 5.2.4.

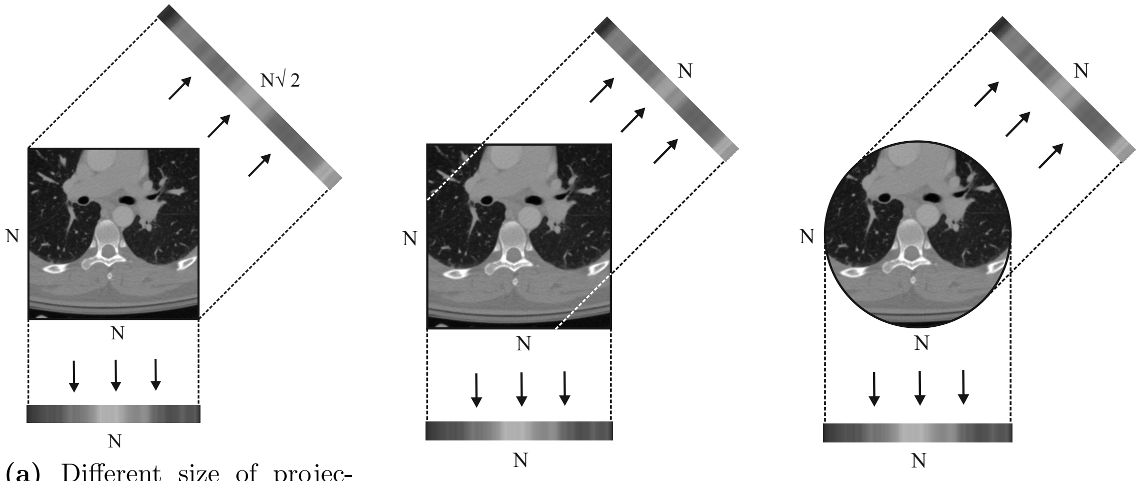
5.2.1 Augmentation

We performed a random data augmentation for any sample drawn from the training set to increase the variability of the available training data, which decreases the risk of the *CNN* to overfit to the training data and helps it to generalize as explained in Section 3.7. Each training sample is augmented on the fly, i.e. without the need of storing the respective augmented sample on the hard drive, by a random translation, rotation and scale. The data augmentation for both, our *3D* and *2D* pipeline is directly applied to the canonical position of the given sample in *3D* space before a *3D* vertebra image respectively a *2D* *CT* slice is extracted. This procedure allows rotational data augmentation on three axes instead of just one also for our *2D* *CT* image slices and further increases the number of possible outcomes. The operations used to augment the training data are explained in Section 3.7.3, more details regarding the parameters for augmentation are given in Section 5.4. In contrast to the training samples, test samples are not augmented.

5.2.2 Masking

The next step after augmenting training samples and extracting training or test samples is to mask the respective image with a circular shape, in this section, we will follow the analogy between the *3D* and *2D* data as shown in Figure 5.2. Masking is required to ensure that all forward projections generated from that *CT* image as explained in Section 5.2.3 have the same properties, which is consequently necessary to correctly reconstruct the *CT* image from the projections. These properties are twofold, first, a one dimensional (1D) projection from any arbitrary angle on the axial plane has to have the same size as every other *1D* projection from that *2D* image and second, every *1D* projection has to contain the same amount of information of the *2D* image. Figure 5.3a shows the problem of varying projection size with the vertically and diagonally acquired projection of a *2D* square image resulting in a different *1D* projection size, which is dependent on the projection angle. Simply enforcing the same size for each *1D* projection, however, leads to different amounts of image information contributing to the projections. As shown in Figure 5.3b, the diagonally acquired *1D* projection would not incorporate the upper left und lower right corners of the image, whereas the vertically acquired projection would contain them. Masking the *2D* *CT* image with a circular shape solves both of these problems as shown in Figure 5.3c, where each generated *1D* projection has the same size and also contains the same image information of the *2D* *CT* slice.

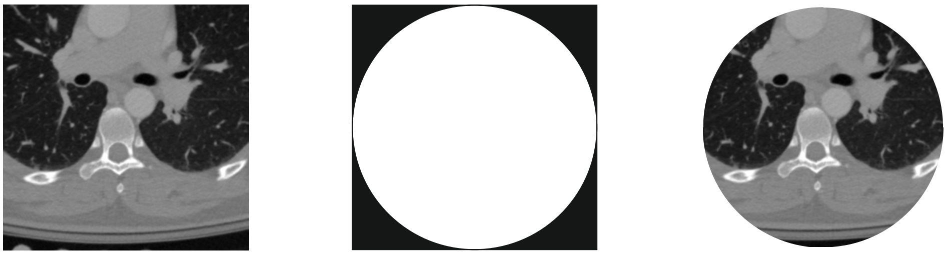
As such, masking of an exemplary *2D* *CT* slice as given in Figure 5.4a is accomplished by utilizing the largest possible circle contained in the given square image as shown in Figure 5.4b and setting all pixels outside of this circle to zero. This procedure results in a circular masked *2D* *CT* slice as given in Figure 5.4c, which prevents the contribution



(a) Different size of projections. **(b)** Caption **(c)** Caption

Figure 5.3: Visualization of difficulties encountered when generating projection data. (a) shows a different size of the individual projections, (b) shows that different image contents contributed to the projections, (c) shows a solution to both problems.

of image areas outside of this circle to only a subset of projections that are generated from that image. Replacing these values by zero corresponds to defining them as air in real *CT* imaging. As such, masking basically serves the purpose of defining the patient to be surrounded by air, which eliminates the problem introduced by using images that are cropped in a rectangular shape. Lastly, while the mask has a circular shape in *2D*, it extends in *3D* to a cylinder, since the individual projections in the *3D* case are only acquired by rotating around the axial plane.



(a) Original. **(b)** Mask. **(c)** Masked image.

Figure 5.4: Image (a) shows the original image, (b) shows a circular image mask, (c) shows the masked image.

5.2.3 Forward Projection

As of now, we augmented samples drawn from the training data to support the *CNN* to generalize as explained in Section 5.2.1 and also talked about masking training and test samples using a circular shaped mask in Section 5.2.2. In the following we will consider a

3D vertebra image as a stack of *2D* *CT* slices as shown in Figure 5.2 and will only focus on the *2D* case that uses masked *2D* *CT* slices to generate a set of *1D* projection views. The *1D* projection views simulate the real projection data that was used to reconstruct the *CT* image in the first place and are required by our *CNN* to learn reconstructing a *CT* image from it's projection data explained in Section 5.3.

Similar to *1D* projection generation during real *CT* imaging, where the local density of matter of a *2D* slice of the patient given a certain direction is basically accumulated as explained in Section 2.4, we utilize a sum projection in which pixel values of a *2D* *CT* slice image representing the matter are summed up given a certain angle. More precisely, we reformulate the parametric version of the line integral in Equation (2.13) that defines how the *1D* projection data $g(\ell, \theta)$ is acquired from a *2D* slice of the patient $\mu(x, y)$ by applying it in discrete space to a *2D* *CT* image $\nu(x, y)$. As such, individual *1D* projections $g_\theta(\ell)$ are defined as

$$\begin{aligned} g_\theta(\ell) &= \sum_s \nu(x(s), y(s)) \\ &= \sum_s \nu(\ell \cdot \cos\theta - s \cdot \sin\theta, \ell \cdot \sin\theta + s \cdot \cos\theta), \end{aligned} \tag{5.1}$$

where θ represents the angle from which a current projection is acquired, $x(s)$ and $y(s)$ are defined in Equation (2.14). The parametric formulation represents a rotation of the coordinate system (x, y) around the center of the image $\nu(x, y)$, which can be viewed as the formation of a new coordinate system (ℓ, s). Following this formulation allows to transform Figure 2.8 representing a visualization in continuous space into a visualization in discrete space as shown in Figure 5.5 as such, that the patient μ is replaced by a discrete image ν. In addition to the coordinate system (x, y), we also added the rotation parameter θ as well as the new coordinate system (ℓ, s) for better visualization. All generated *1D* forward projections $g(\ell, \theta)$ are then used for further processing as described in Section 5.2.4 before they result in the input of our *CNN*.

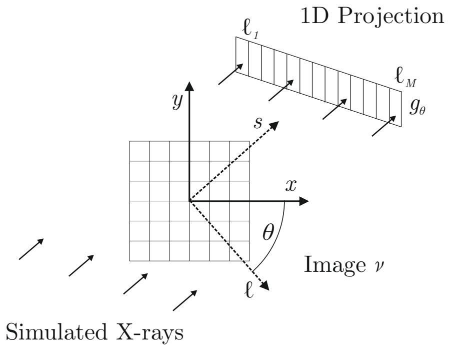


Figure 5.5: A visualization of the discrete forward projection generation.

5.2.4 Back Projection

After the forward projections have been acquired from the *CT* images from different angles around the axial plane, they are used to generate back projection images from them. Again, we will focus on the *2D* case in the following and use the analogy between *2D* and *3D CT* images where we view *3D CT* images as a stack of *2D CT* slices that are processed individually as shown in Figure 5.2.

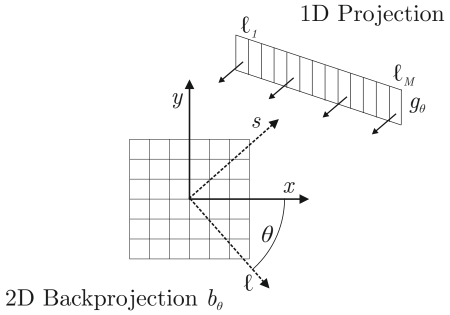


Figure 5.6: A visualization of the discrete back projection procedure.

Each *1D* forward projection we generate as explained in Section 5.2.3 is back projected onto a *2D* plane similar to the back projection algorithm described in Section 2.5.1. Back projection is simply done by repeating a *1D* forward projection $g_\theta(\ell)$ onto an empty *2D* image $b_\theta(x, y)$, where the size of this image is equivalent to the size of the *2D CT* slice image $\nu(x, y)$ from which the set of *1D* forward projections has been generated. The key difference to the simple back projection algorithm – of which the discrete version is given in Equation (2.21) – is, that our method back projects each *1D* forward projection $g_\theta(\ell)$ onto an individual *2D* projection image $b_\theta(x, y)$ to prevent the accumulation of the individual back projections. Following this formulation, we define a *2D* back projection image $b_\theta(x, y)$ that corresponds to the forward projection $g_\theta(\ell)$ of angle θ as

$$\begin{aligned} b_\theta(x, y) &= g_\theta(\ell) \\ &= g_\theta(x \cdot \cos\theta + y \cdot \sin\theta), \end{aligned} \tag{5.2}$$

which is schematically visualized in Figure 5.6. The sequential visualization of the forward and back projection as we used it is given in Figure 5.7 showing an exemplary target *CT* image as well as the corresponding back projected forward projections that are used as an input by our *CNN* as explained in Section 5.3.

5.3 Learned Computed Tomography Reconstruction

In this thesis we propose a learning based *CT* reconstruction method from a reduced set of forward projections, where the reconstruction itself is learned by a *CNN* we call the

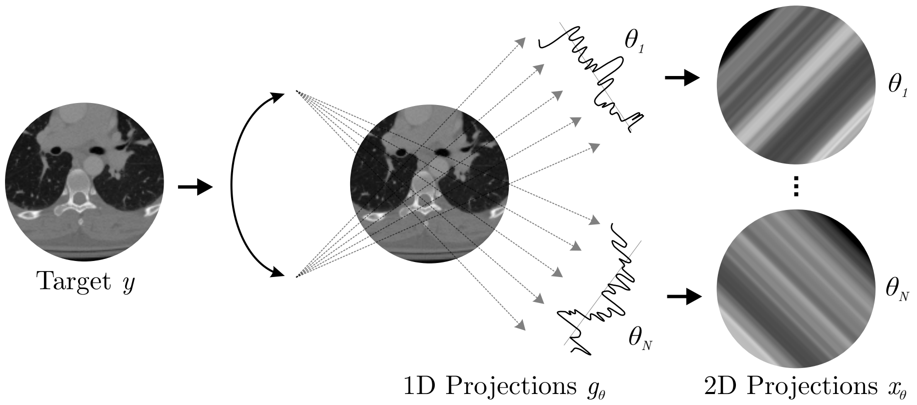


Figure 5.7: A visualization of the projection image generation from a target image.

generator G. This generator *CNN* utilizes a reduced number of back projected forward projections as explained in Section 5.2 as an input and the corresponding full-view reconstructed *CT* image as a target to optimize *CT* image reconstruction. As such, the generator *CNN* falls into the category of approaches that learn a domain transfer from projection to image domain as explained in Section 4.1.4. The forward projections have been simulated to allow data augmentation and increase the variability of the available data as explained in Section 5.2.1 as well as to allow a broader variety of experiments. First, we proposed our *3D* pipeline which we later extended by utilizing a *WGAN* as explained in Section 3.6, however, utilizing a *WGAN* in *3D* turned out to be unfeasible due to the vastly increased complexity. Thus, we decided to reduce the dimensionality of the data from *3D* to *2D* which additionally allowed to increase the resolution for our *2D* experiments as well as the number of training iterations, more details regarding the setup parameters are given in Section 5.4. While Figure 5.8 visualizes our *3D* pipeline, the *2D* pipeline we utilized is shown in Figure 5.9.

The generator *CNN* used by our *3D* and *2D* pipeline is explained in Section 5.3.1, while the discriminator only utilized by our *2D* pipeline is described in Section 5.3.2. Finally, Section 5.3.3 is dedicated to the loss functions used to optimize the *CNNs*.

5.3.1 Generator Network Architecture

Our generator G expects a set of N projections generated from the target image $y \sim \mathbb{P}_r$ as explained in Section 5.2.4 as input. Optimized by the loss function explained in Section 5.3.3, the generator learns to reconstruct an estimation image $\hat{y} \sim \mathbb{P}_g$, which is as similar as possible to y. The network architecture we used for G is based on the U-Net [72] representing an image-to-image network as explained in Section 3.5.2.1, the U-Net is visualized in Figure 3.11.

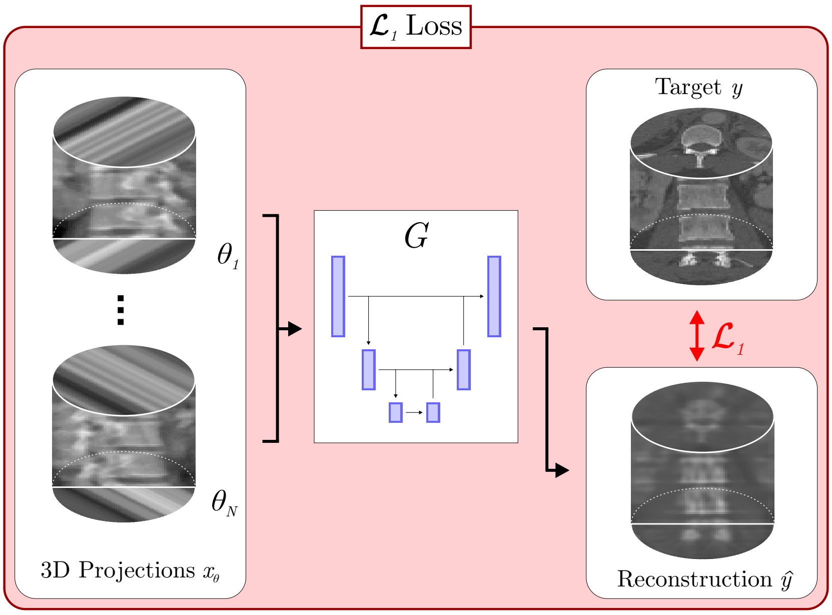


Figure 5.8: A schematic visualization of our *3D* method.

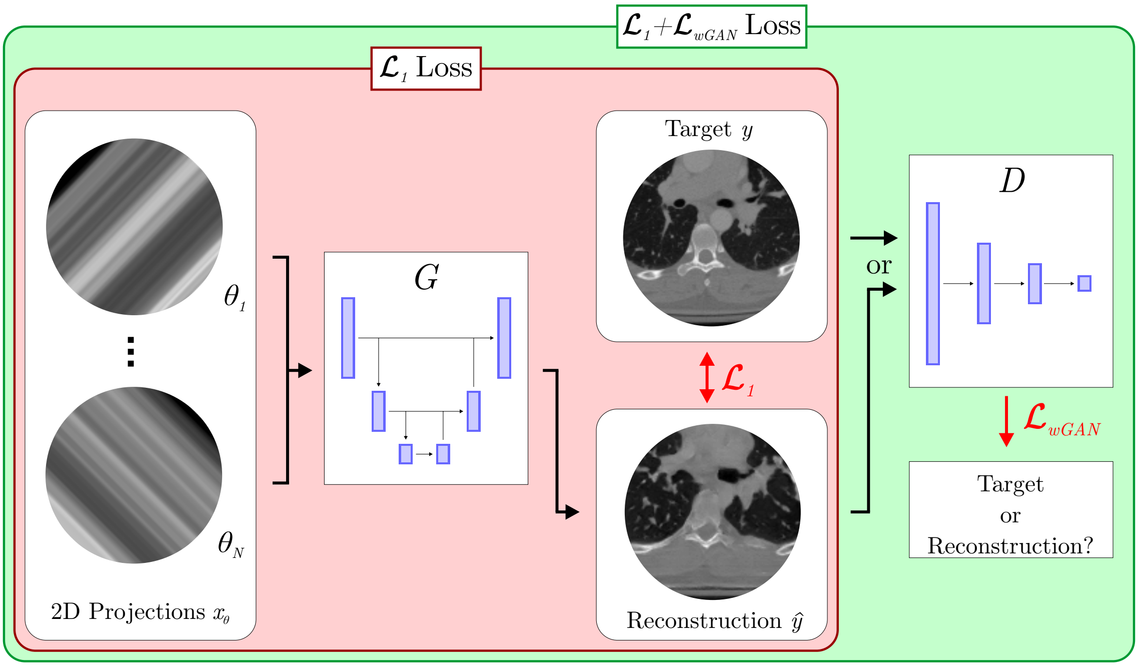


Figure 5.9: A schematic visualization of our two *2D* methods.

5.3.2 Discriminator Network Architecture

The discriminator D is required for the *WGAN* based training scheme as explained in Section 3.6 and is utilized only in our *2D* pipeline to support optimizing the generator by providing an additional adversarial $\mathcal{L}_{wGAN}$ loss explained in Section 5.3.3. The discriminator expects either an image y from the real data distribution or a reconstructed image $\hat{y}$ from the generator's reconstructed data distribution. The task of the discriminator is to decide, whether an observed image is coming from $\mathbb{P}_r$ or $\mathbb{P}_g$ by computing a scalar value, which represents the probability of any observed image $\{y, \hat{y}\}$ belonging to the set of real images $\mathbb{P}_r$. The network architecture of the discriminator represents a binary classification network as described in Section 3.5.1.1, a visualization is given in Figure 3.9.

5.3.3 Loss Function

The generator *CNN* in our *3D* pipeline was optimized utilizing only $\mathcal{L}_1$ loss as defined in Equation (3.10), while for our *2D* pipeline we extended the generator *CNN* to also use an adversarial $\mathcal{L}_{wGAN}$ loss coming from a Wasserstein Generative Adversarial Network with Gradient Penalty (WGAN-GP) similar to [28]. For that purpose a second *CNN*, namely the discriminator D was introduced which helps to optimize the generator G as explained in Section 3.6.2. Since Generative Adversarial Networks (GANs) have been introduced for data generation from random noise, we modified the loss function as such that a prior instead of a random noise vector is used as explained in Section 3.6.3. This brings us to the formulation defined in Equation (3.35), in our simplified notation we express the loss function used to optimize the discriminator D as

$$\mathcal{L}_D = -D(y) + D(\hat{y}) + \rho, \tag{5.3}$$

where $D(y)$ represents the predicted probability by the discriminator of a real sample y coming from the real distribution $\mathbb{P}_r$. $D(\hat{y})$ is defined as the predicted probability for a generated sample $\hat{y}$ coming from the real data distribution $\mathbb{P}_r$ and not from the distribution of generated samples $\mathbb{P}_g$. The gradient penalty is given by ρ, which stabilizes the training of the *WGAN-GP* similar to [28]. The simplified notation of the loss function of the generator G according to Equation (3.35) and also to [28] is defined as

$$\mathcal{L}_{wGAN} = -D(\hat{y}), \tag{5.4}$$

however, we additionally combined the adversarial loss of the generator G with a content loss as described in Section 3.6.3. We define this combined loss function as

$$\mathcal{L}_G = \mathcal{L}_1 + \lambda \cdot \mathcal{L}_{wGAN}, \tag{5.5}$$

where $\mathcal{L}_1$ represents the content loss as in Equation (3.10) and $\mathcal{L}_{wGAN}$ is the adversarial loss as in Equation (5.4). The weight between the two loss functions is given by λ and represents a hyperparameter for which we conducted a variety of experiments explained in detail in Chapter 6.

5.4 Experimental Setup

The experimental setup gives insight into the material that we used in this work as well as into implementation details regarding our method. While the material is explained in Section 5.4.1, we separated the implementation details into three parts, where Section 5.4.2 represents details that are identical for all experiments. Section 5.4.3 focuses on implementation details that are only valid for our *3D* pipeline, while the Section 5.4.4 introduces details specific to the *2D* pipeline.

5.4.1 Material

The material we used consists of 10 already reconstructed normal-dose *3D* *CT* images each from a different patient. The data was published for a challenge at the Computational Methods and Clinical Applications for Spine Imaging (CSI) 2014 Workshop[1] at Medical Image Computing and Computer Assisted Intervention (MICCAI). Each of these *3D* *CT* images contains information from neck to pelvis of the patient and is cropped around the spine. Furthermore, a segmentation of every vertebra contained in each image is also available, which we utilized for our *3D* pipeline as explained in Section 5.1.2.

The *CT* images have a size of $512 \times 512 \times \{507, \ldots, 625\}$, however, since utilizing the *CT* images at full size is not feasible due to hardware limitations and immense execution time for training, we downsampled the data to a more reasonable size, i.e. $64 \times 64 \times 64$ for our *3D* pipeline and 128×128 for our *2D* pipeline. The huge downsampling factor that is required by our *3D* pipeline leads to the loss of a significant amount of information, which is not neglectable. However, since it is sufficient to proof the concept of the *3D* implementation of our approach, we decided to solve this problem by cropping individual *3D* vertebra images from the full *3D* *CT* images before downsampling them to a resolution of $64 \times 64 \times 64$, see Section 5.1.2. This solution allowed us to decrease the downsampling factor which consequently reduces the amount of lost information. For our *2D* pipeline we directly extracted the axial *2D* *CT* slices from the full size *CT* images without any cropping. The downsampling was conducted after extracting the *2D* *CT* slices to increase the number of different slices.

The 10 *3D* *CT* images are separated by patient into a training and a test set, where the training set consists of eight and the test set of two *3D* *CT* images. Since our *3D* pipeline uses *3D* vertebra images that are cropped from the full *3D* images, the number of samples increases to the number of vertebrae contained in the images yielding 141 training samples and 35 test samples. Similarly, for our *2D* pipeline the number of available samples

---

[1] The dataset we used in this work: `csi-workshop.weebly.com/challenges.html`

increases to the number of axial slices in the full size *CT* images, i.e. to approximately 4.000 training and 1.000 test samples.

For the experiment conducted in Section 6.2.2.6 where we evaluated the performance of our method on a site unseen during training, we required an additional dataset. We decided to use *2D* head *CT* slices which we extracted from an in-house dataset that consists of a number of full-body *CT* images of corpses. After cropping the head from these *CT* images, we downsampled them to a resolution of 128×128.

5.4.2 Setup Parameters for All Experiments

Both, our *3D* and our *2D* pipeline utilized a U-Net [72] based architecture for the generator which is described in Section 5.3.1. The generator is trained with a depth of four levels and He normal [33] is used as a kernel initializer. All convolution layers except for the discriminator use a zero padding. Intermediate convolutions utilize a kernel size of 3×3 respectively $3 \times 3 \times 3$, 64 filters and Rectified Linear Unit (ReLU) [60] as an activation function. For the final convolution a kernel size of 1×1 respectively $1 \times 1 \times 1$, only one filter and no activation function is used. Upsampling was conducted using nearest neighbor. The kernel size of the upsampling layer and the pooling layer was set to 2×2 and $2 \times 2 \times 2$ respectively. For all experiments we utilized data augmentation as explained in Section 3.7.3, which consists of a random translation, rotation and scale sampled from a uniform distribution, where newly introduced image areas were defined as zero representing air. Adaptive Moment Estimation (ADAM) [41] was used as an optimizer for all networks, however, the *ADAM* parameters were chosen differently.

5.4.3 Setup Parameters Specific to 3D Experiments

Our *3D* experiments are conducted using images with a size of $64 \times 64 \times 64$. The *3D* target images used as network input are cropped as such that they incorporate 120 mm in each dimension. The networks are trained for 40.000 iterations with a mini-batch size of one. *ADAM* was used as an optimizer with a learning rate of 0.0002, the first and second momentum estimates are set to $\beta_1 = 0.9$ and $\beta_2 = 0.999$. As a loss function $\mathcal{L}_1$ was used, see Equation (3.10), and as weight regularization $\mathcal{L}_2$ was utilized with a factor of 0.0005. Average pooling was utilized to downsample the images. Agumentation was conducted with a translation defined in physical coordinates by 15 Millimeter, a rotation defined by $30°$ and a scale defined by 15 percent in either direction.

5.4.4 Setup Parameters Specific to 2D Experiments

For our *2D* experiments we used an image size of 128×128. The generator was trained for 80.000 iterations, while the discriminator was trained five times per generator iteration. The mini-batch size was set to 16 for all *2D* experiments. All networks are optimized using *ADAM*, the learning rate was set to $\eta = 10^{-3}$, while $\beta_1 = 0.5$, $\beta_2 = 0.9$ and

$\epsilon = 10^{-8}$. Different loss functions were used for different experiments. The generator was either trained solely on $\mathcal{L}_1$ loss, see Equation (3.10), or on a combination of $\mathcal{L}_1$ and an adversarial loss $\mathcal{L}_{wGAN}$ loss, see Equation (5.5). We also conducted some experiments where we only utilized an adversarial loss $\mathcal{L}_{wGAN}$ loss. The discriminator was trained using Equation (5.3), similar to [28]. Image downsampling was done using max pooling layers. The discriminator utilized Leaky Rectified Linear Unit (Leaky ReLU) [33] as an activation function and was trained using five levels. Data augmentation was done utilizing a translation defined in pixel space with a value of 20, rotation was defined in radiant with a value of 0.1 and scale in percent with a value of 20. Scale, however, was not applied along the homogenous axis of the axial plane since that would interfer with a translation along this axis and also would be obsolete due to the slices extraction anyways. Furthermore, rotational augmentation was done directly on the *3D* volume before *2D* slices have been extracted to vastly increase the augmentation possibilites for our *2D* slices.

6

Results

Contents

6.1 Evaluation of 3D Experiments 77
6.2 Evaluation of 2D Experiments 83

The proposed Convolutional Neural Networks (CNNs) include a three dimensional (3D) or a two dimensional (2D) *CNN* trained only on $\mathcal{L}_1$ loss. We also trained a *2D CNN* on a combined loss function consisting of $\mathcal{L}_1$ loss and $\mathcal{L}_{wGAN}$ loss utilizing a Generative Adversarial Network (GAN) as proposed in [28]. Section 6.1 represents the evaluation of our *3D* method *3D*-$\mathcal{L}_1$-only, while in Section 6.2 we give insight into our *2D* methods, i.e. *2D*-$\mathcal{L}_1$-only and *2D*-$\mathcal{L}_1$-$\mathcal{L}_{wGAN}$.

6.1 Evaluation of 3D Experiments

The results generated using our *3D* pipeline where *CNNs* are optimized utilizing $\mathcal{L}_1$ loss as explained in Section 5.3 are called *3D*-$\mathcal{L}_1$-only. We compare the results of *3D*-$\mathcal{L}_1$-only quantitatively and qualitatively to the results of the Filtered Backprojection (FBP) method, which is a non-learning based analytical method and explained in Section 2.5.3. Evaluation is conducted using a different number $N \in \{1, 2, 4, 6, 8, 15, 30, 60, 120, 180\}$ of projection views utilized for reconstruction. The quantitative evaluation of our *3D* method is given in Section 6.1.1, while the qualitative evaluation is shown in Section 6.1.2.

6.1.1 Quantitative Evaluation

We evaluated the quantitative results of our *3D*-$\mathcal{L}_1$-only and the *FBP* method by calculating the Mean Absolute Error (MAE) between the target volume and the reconstructed prediction generated using the respective method for a different number of projection views. As such, Figure 6.1 compares the results generated by our *3D*-$\mathcal{L}_1$-only method to the results of the non-learning based *FBP* method. Additionally, Table 6.1 shows the

N	$\mathcal{L}_1$	FBP
1	**6.06** ± 2.24	41.5 ± 7.39
2	**3.59** ± 0.8	31.77 ± 5.43
3	**3.59** ± 0.72	19.07 ± 3.32
4	**3.17** ± 0.58	17.11 ± 2.87
5	**3.12** ± 0.58	11.9 ± 1.88
6	**2.99** ± 0.63	12.04 ± 2.07
7	**2.76** ± 0.51	8.9 ± 1.43
8	**2.68** ± 0.53	8.95 ± 1.46
15	**2.02** ± 0.38	4.63 ± 0.73
30	**1.69** ± 0.33	2.62 ± 0.37
60	**1.4** ± 0.26	1.76 ± 0.23
120	**1.27** ± 0.21	1.74 ± 0.23
180	**1.43** ± 0.3	1.74 ± 0.23

Table 6.1: Quantitative results evaluating the *MAE* of our *3D* method compared to the *FBP* method. The results are multiplied by 10^2.

exact results of the *MAE* and the standard deviation that have been generated from both methods.

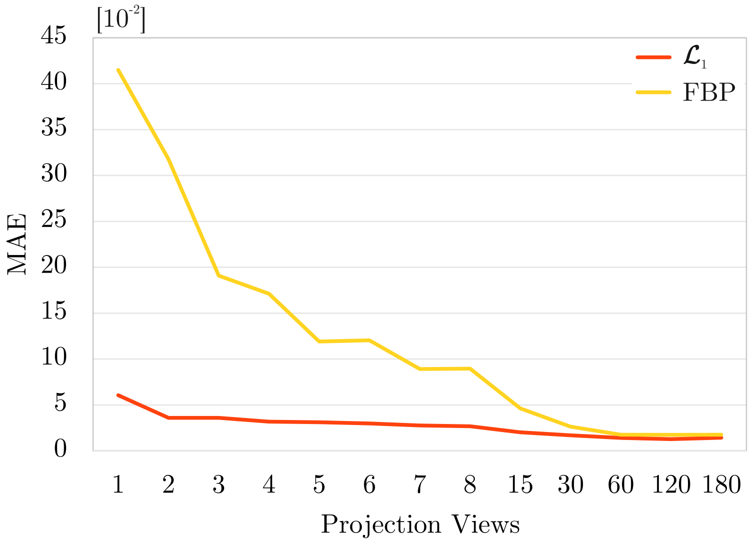


Figure 6.1: Quantitative results evaluating the *MAE* of our *3D* method compared to the *FBP* method.

6.1.2 Qualitative Evaluation

In this section we evaluate the results of our *3D*-$\mathcal{L}_1$-only method qualitatively. Since the resolution of the cropped vertebra volumes we utilized in our *3D* method as explained in Section 5.1.2 is rather small and each volume contains a vertebra and it's surroundings, we will focus on a qualitative comparison of the reconstructed vertebra to the target volume's vertebra. All images correspond to one another by sharing the same brightness setting, however, some values are truncated to yield a better contrast, especially true for the visual representation of the *FBP* results generated from a very small number of views. Furthermore, all results correspond to one another by having the same center pixel, which is also true for the different views. In Section 6.1.2.1 we give some insight into how we conducted the qualitative evaluation in this work. Different qualitative results using a different number of views for a selected axial and sagittal slice are given in Section 6.1.2.2. Lastly, reconstructed images of some additional slices are shown in Section 6.1.2.3.

6.1.2.1 Overview of a Selected Volume Used for Evaluation

To demonstrate the qualitative results, we selected a representative vertebra including it's surrounding structures of which we extracted the central axial and sagittal slice. An overview of the selected axial and sagittal view with an additional visualization of different substructures of the vertebra as well as the corresponding plane in the other view is shown in Figure 6.2. To compare the different results, each reconstructed volume is visualized showing the center slice of the axial and sagittal view. Furthermore, we focus on the reconstruction quality of the vertebra and the largest substructures of it, which are the vertebra's body, the left and right transverse process as well as the spinous process.

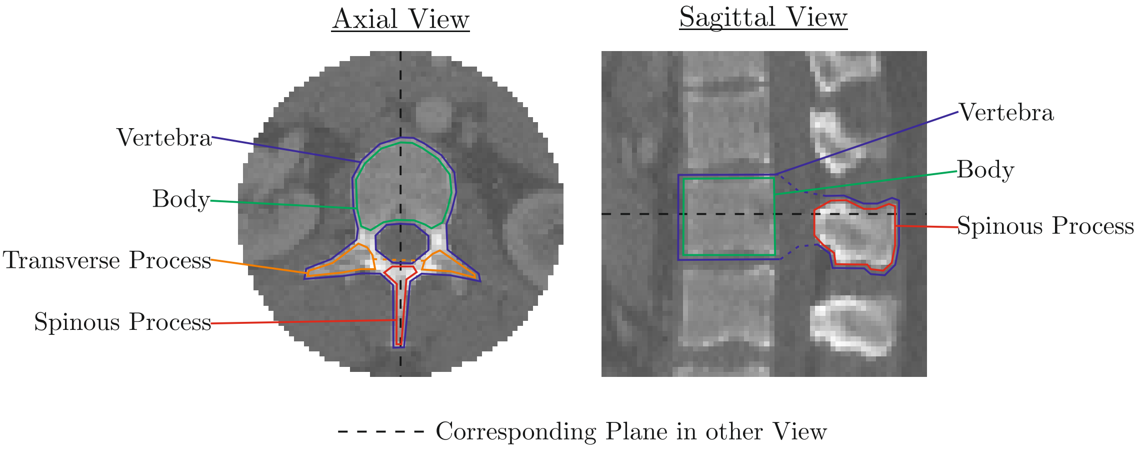


Figure 6.2: A visualization of important structures considered in the qualitative evaluation of our *3D* method.

6.1.2.2 Varying the Number of Projection Views for a Selected Volume

For the evaluation of our *3D* results we generated images using a different number of projection views $N \in \{1, 2, 4, 6, 8, 15, 30, 60\}$, where we trained an individual *CNN* for each N. We simulated the real projection data by generating projection views from already reconstructed normal-dose Computed Tomography (CT) images as explained in Section 5.2. The number of projection views used by our *CNN* to reconstruct a *CT* image correlates to the amount of information that is available and heavily influences the quality of the reconstructed *CT* image. In the following, we observe the influence of the different number of views to the resulting reconstruction quality.

Up to Eight Views Using only one projection view for reconstruction does not lead to a meaningful result for either *3D*-$\mathcal{L}_1$-only or the *FBP* method, see Figure 6.3. However, by increasing the number of views utilized to two, our *3D*-$\mathcal{L}_1$-only method is able to visualize the silhouette of the vertebra giving a coarse shape of the vertebra's body. Further increasing the number of views leads to more details visible in our *3D*-$\mathcal{L}_1$-only results as shown in Figure 6.3. While six views are sufficient to clearly visualize the vertebra's body, eight projection views are enough to represent all relevant structures of the vertebra. I.e. the vertebra's body, the left and right transverse process and the spinous process are clearly indicated and distinguishable structures using eight views and our method, while the results of the *FBP* method still suffers from introduced streaking artifacts.

Up to 60 Views By using 15 projection views, the results of *3D*-$\mathcal{L}_1$-only are already quite good with the transverse and spinous processes being clearly visible as given in Figure 6.3. The *FBP* method is now able to visualize the vertebra and give indications of the largest structures of it, however, streaking artifacts are still introduced and degrade the reconstructed image's quality. Our *3D*-$\mathcal{L}_1$-only method is able to give a very good reconstruction using 30 views showing sharper edges and small details of the vertebra's exact structure as well as the surrounding area as shown in Figure 6.3. The *FBP* method was able to reduce the amount of streaking artifacts to an amount where all relevant structures of the vertebra are clearly shown and some smaller details become visible. Using 60 views leads to very similar results for both methods, while the *3D*-$\mathcal{L}_1$-only results are very similar to those generated using 30 views, the *FBP* method was able to reduce the amount of streaking artifacts to a degree where they are not visible anymore. The quality of the reconstructed images using our *3D*-$\mathcal{L}_1$-only and the *FBP* method can be considered equivalent from that point on.

6.1.2.3 Evaluation of Additional Slices

As of now, we only showed qualitative results of a selected slice, thus, in the following we will also present qualitative results of additional slices. The qualitative results of these slices are presented in Figure 6.4, where the center slice in the axial and sagittal view for

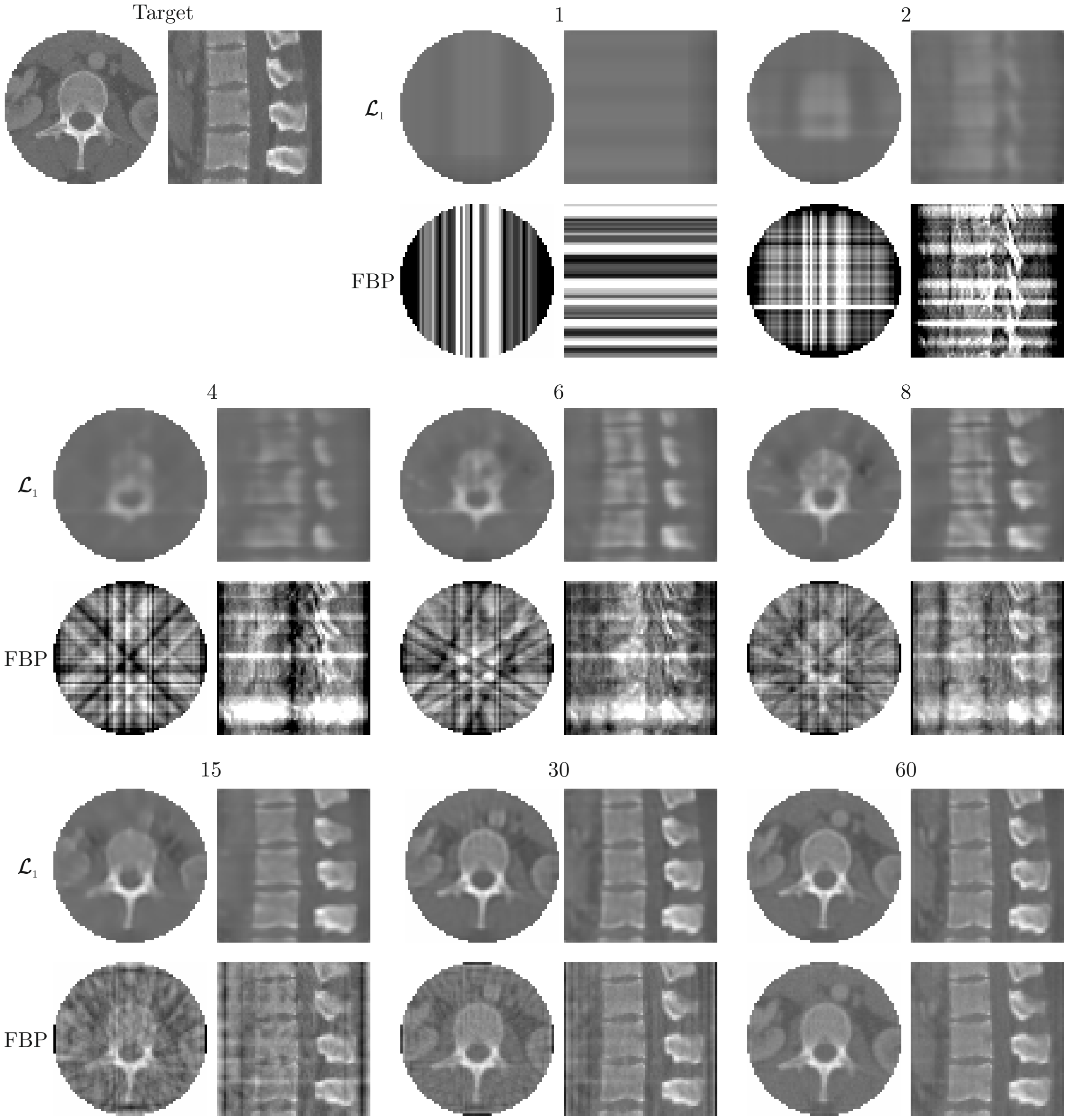


Figure 6.3: Qualtitative results of our *3D* method and the *FBP* method for a different number of views.

different vertebrae is shown. The results have been generated by our *3D*-$\mathcal{L}_1$-only method using eight and 30 projection views.

Eight Views The reconstructed images show that eight projection views are enough to yield a reconstruction that overall corresponds to the target, however, the results look blurry. While the general structure of the spine is visible in all images, the sagittal view shows that vertebrae located in the lung area look blurrier than vertebrae above and below the lung.

30 Views Increasing the number of projection views to 30 yields results of a very similar quality independent of the exact location of the vertebra. The anatomical structure of the reconstructed vertebrae generated using 30 projection views is very similar to the target vertebrae, however, some streaking artifacts are visible in the reconstructed volumes.

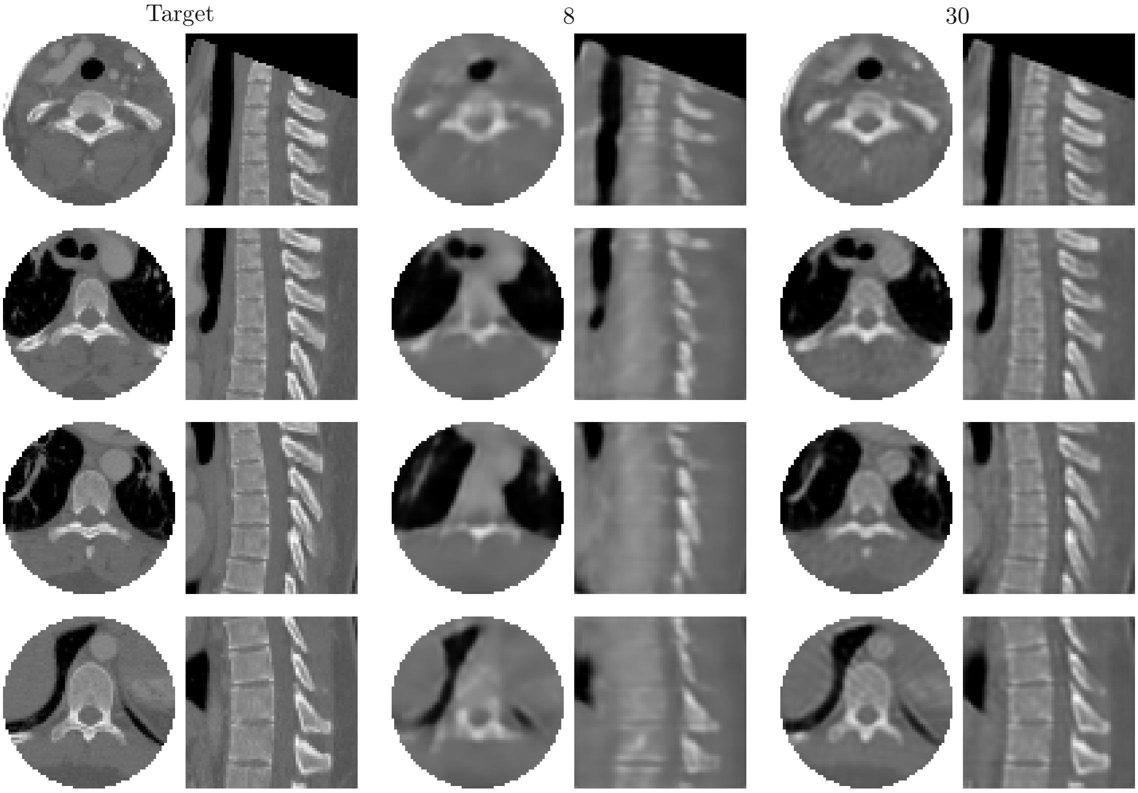


Figure 6.4: Qualtitative results of our *3D* method showing additional slices for eight and 30 views.

N	$\mathcal{L}_1 + \mathcal{L}_{\mathbf{wGAN}}$	$\mathcal{L}_1$	**FBP**
1	4.62 ± 1.8	$\mathbf{3.71} \pm 1.5$	42.14 ± 1.98
2	2.97 ± 1.01	$\mathbf{2.35} \pm 0.79$	37.1 ± 1.71
4	2.4 ± 0.65	$\mathbf{1.85} \pm 0.53$	22.77 ± 1.37
6	2.12 ± 0.55	$\mathbf{1.63} \pm 0.43$	16.89 ± 0.71
8	1.81 ± 0.42	$\mathbf{1.44} \pm 0.34$	12.67 ± 0.65
15	1.39 ± 0.27	$\mathbf{1.25} \pm 0.24$	7.38 ± 0.34
30	1.16 ± 0.2	$\mathbf{0.83} \pm 0.14$	4.81 ± 0.13
60	0.73 ± 0.1	$\mathbf{0.65} \pm 0.09$	3.49 ± 0.04
120	0.51 ± 0.06	$\mathbf{0.46} \pm 0.05$	3.09 ± 0.06
180	0.5 ± 0.06	$\mathbf{0.43} \pm 0.05$	3.06 ± 0.06

Table 6.2: Quantitative results evaluating the *MAE* of our *2D* methods compared to the *FBP* method. The results are multiplied by 10^2.

6.2 Evaluation of 2D Experiments

The evaluation of our *2D* experiments incorporates two methods using different loss functions as explained in Section 5.3. While one method solely utilizes $\mathcal{L}_1$ loss which we call *2D*-$\mathcal{L}_1$-only, the other method uses a combined loss function consisting of $\mathcal{L}_1$ loss and $\mathcal{L}_{wGAN}$ loss called *2D*-$\mathcal{L}_1$-$\mathcal{L}_{wGAN}$. We compare the results of both methods quantitatively and qualitatively to one another as well as to the results of the non-learning based *FBP* method, which is explained in Section 2.5.3. Both proposed methods are evaluated using a different number $N \in \{1, 2, 4, 6, 8, 15, 30, 60, 120, 180\}$ of projection images for reconstruction. Furthermore, for our *2D*-$\mathcal{L}_1$-$\mathcal{L}_{wGAN}$ we conducted experiments using a different weight $\lambda \in 10^{\{-6,-5,-4,-3,-2,-1\}}$ between the $\mathcal{L}_{wGAN}$ loss and the $\mathcal{L}_1$ loss. As default values we chose $N = 8$ and $\lambda = 10^{-3}$. The quantitative evaluation of *2D*-$\mathcal{L}_1$-only and *2D*-$\mathcal{L}_1$-$\mathcal{L}_{wGAN}$ is shown in Section 6.2.1, while the qualitative evaluation is given in Section 6.2.2.

6.2.1 Quantitative Evaluation

The quantitative evaluation of our *2D* methods was conducted by computing the *MAE* as well as the Structural Similarity Index Metric (SSIM) of *2D*-$\mathcal{L}_1$-$\mathcal{L}_{wGAN}$, *2D*-$\mathcal{L}_1$-only and the *FBP* method to the target image. A visualization of the *MAE* measurements of the three methods using a different number of projection views is shown in Figure 6.5, the exact values are given in Table 6.2. Furthermore, we evaluated the results calculating the *SSIM* measurements of each method again using a different number of views as shown in Figure 6.6 – exact numbers are given in Table 6.3. *SSIM* was calculated only from the image area inside of the circle mask described in Section 5.2.2.

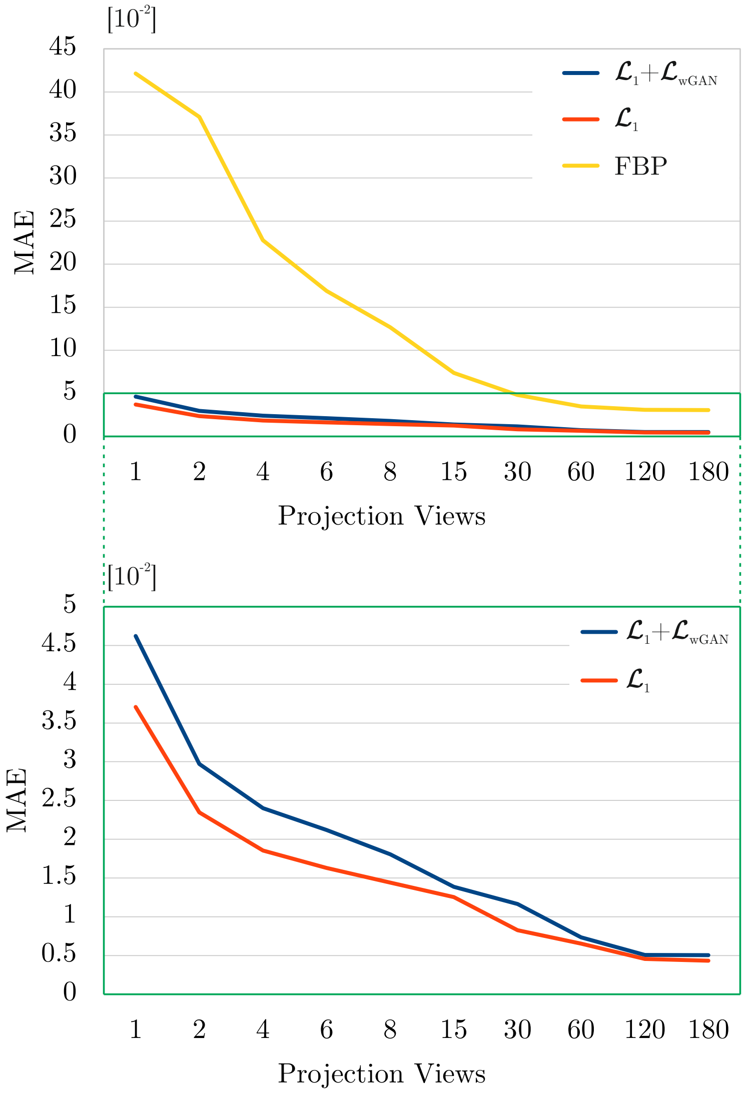


Figure 6.5: Quantitative results evaluating the *MAE* of our *2D* methods compared to the *FBP* method.

6.2.2 Qualitative Evaluation

The qualitative evaluation of our *2D* methods, i.e. *2D*-$\mathcal{L}_1$-$\mathcal{L}_{wGAN}$ and *2D*-$\mathcal{L}_1$-only is conducted in this section. We carefully evaluated the quality of the reconstructed images by comparing them to the target images focusing on the overall anatomical structure as well as on small details and give a detailed description of small changes in the images also evaluating different hyperparameter settings. We scaled all *CT* slices used in this section to a specific brightness range to achieve visual correspondence and a good contrast. However, to accomplish this, we truncated some values which is especially visible for the *FBP*

N	$\mathcal{L}_1 + \mathcal{L}_{\mathbf{wGAN}}$	$\mathcal{L}_1$	**FBP**
1	0.12 ± 0.02	$\mathbf{0.14} \pm 0.02$	0.02 ± 0.01
2	0.22 ± 0.03	$\mathbf{0.25} \pm 0.03$	0.01 ± 0.01
4	0.28 ± 0.04	$\mathbf{0.31} \pm 0.04$	0.08 ± 0.02
6	0.33 ± 0.04	$\mathbf{0.36} \pm 0.05$	0.13 ± 0.02
8	0.37 ± 0.05	$\mathbf{0.39} \pm 0.05$	0.16 ± 0.03
15	$\mathbf{0.48} \pm 0.06$	0.47 ± 0.07	0.25 ± 0.05
30	$\mathbf{0.6} \pm 0.07$	$\mathbf{0.6} \pm 0.08$	0.37 ± 0.06
60	$\mathbf{0.73} \pm 0.06$	$\mathbf{0.73} \pm 0.07$	0.55 ± 0.07
120	$\mathbf{0.83} \pm 0.04$	$\mathbf{0.83} \pm 0.05$	0.75 ± 0.05
180	$\mathbf{0.86} \pm 0.03$	0.85 ± 0.04	0.81 ± 0.04

Table 6.3: Quantitative results evaluating the Structural Similarity Index Measure of our 2D methods compared to the Filtered Backprojection method.

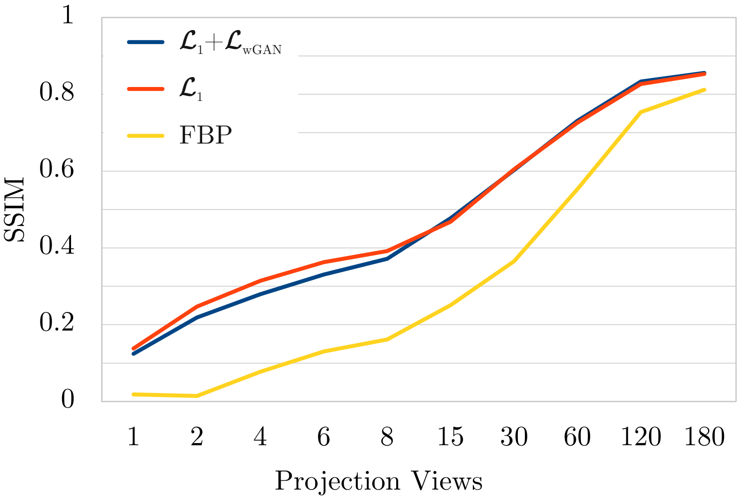


Figure 6.6: Quantitative results evaluating the Structural Similarity Index Measure of our 2D methods compared to the Filtered Backprojection method.

method reconstructed images using a very small number of views. In Section 6.2.2.1 we give an overview of the selected slice used to visualize the qualitative evaluation in this work. Results generated from a varying number of views are given in Section 6.2.2.2 for both methods, while Section 6.2.2.3 focuses on our *GAN* based method *2D*-$\mathcal{L}_1$-$\mathcal{L}_{wGAN}$ showing different results achieved by utilizing a different weight between the two loss functions. In Section 6.2.2.4 we show results for additional slices that have been reconstructed by our method and after stacking the reconstructed *2D* axial slices, we evaluate the quality of the results by observing the coronal and sagittal view in Section 6.2.2.5. Finally, in Section 6.2.2.6 we evaluate the performance of both methods on *2D* head *CT* slices representing a site that the network has never seen during training.

6.2.2.1 Overview of a Selected Slice Used for Evaluation

For the evaluation of the visual reconstruction quality of our *2D* methods we selected a representative slice from the test set which is used to observe the influence of different hyperparameters in this work. The selected slice including visualizations of all relevant structures of that slice are as shown in Figure 6.7. The criteria to observe the various anatomical structures of the given slice include the presence, location, shape and integrity of the respective structure as well as the correspondence to the target image. The structures we observed and described in the following consist of bones, soft tissues and the lung visible in that specific slice. The bones present in the observed slice are the vertebra, two ribs as well as two heads of ribs and the spinous process of the vertebra located above, however, the right rib is occluded due to a zoom box we utilized in the upcoming images for better visibility of small details. Furthermore, we observed the descending aorta, the heart and the pulmonary artery, and the three bronchi present in the image. Also, the two lung lobes as a whole structure as well as the blood vessels contained in them were observed. For the blood vessels we distinguished between vessels having a large, medium and small size.

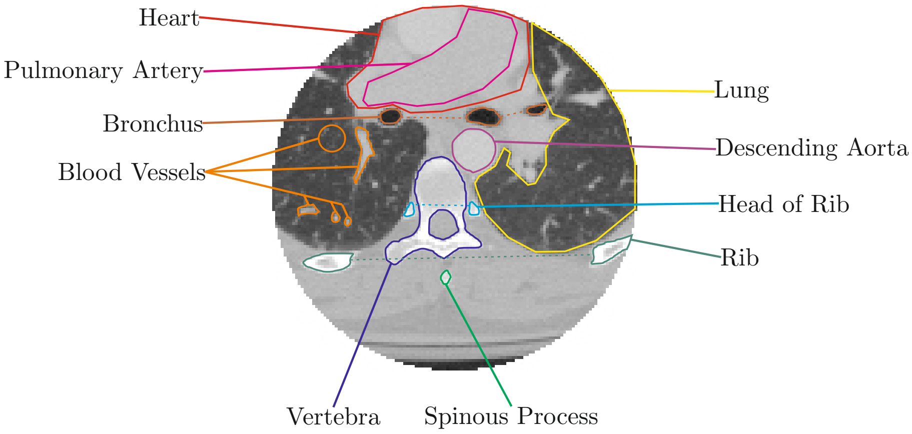


Figure 6.7: A visualization of important structures considered in the qualitative evaluation of our *2D* method.

6.2.2.2 Varying the Number of Projection Views for a Selected Slice

The number of views represents the number of projections used to reconstruct the target image, projection image generation is explained in Section 5.2. The number of views correlates to the amount of information that is available and can be utilized by the *CNN* and thus, has a major influence on the quality of the reconstructed image. In the following we observed a different number of views $N \in \{1, 2, 4, 6, 8, 15, 30, 60\}$, separated into different paragraphs. For the *FBP* method we also appended results generated using $N \in \{120, 180, 360\}$.

One View The visualization of the lung as distinguishable structure from soft tissues and bones shown in Figure 6.8 is accomplished by the two learning based methods *2D*-$\mathcal{L}_1$-only and *2D*-$\mathcal{L}_1$-$\mathcal{L}_{wGAN}$ already by utilizing one projection view, whereas the *FBP* method requires eight views to reduce the amount of streaking artifacts enough to visualize anything that is meaningful. Furthermore, while *2D*-$\mathcal{L}_1$-only and *2D*-$\mathcal{L}_1$-$\mathcal{L}_{wGAN}$ using only one view both place bones near the image borders at the top and bottom of the image due to extreme uncertainty, *2D*-$\mathcal{L}_1$-$\mathcal{L}_{wGAN}$ is able to locate the vertebra as well as both ribs in the image, the reconstructed structures are, however, anatomically not corresponding to the target. While the learning based methods are able to roughly visualize the general anatomical structure of the slice, due to the extremly sparse information, none of the images produces anything clinically meaningful by using only one view.

Two Views The reconstructed image's quality is already much better when two views are utilized for reconstruction, see Figure 6.8. Especially the ambiguity of bone placement for the learning based methods has decreased as bones are not placed at the image borders where they do not belong as observed on the reconstructed images using only one view. By utilizing two views, *2D*-$\mathcal{L}_1$-only is able to visualize the silhoutte of the vertebra and the ribs, while *2D*-$\mathcal{L}_1$-$\mathcal{L}_{wGAN}$ was able to improve on the vertebra's structure and the ribs look already quite believeable though placed incorrectly. Furthermore, the right and left bronchus, that split from the airway to supply both lung lobes with air, are also identified in both learning based reconstructions. While the right bronchus is clearly visible, the left one is indicated by a slightly darker image region in comparison to the rest of the lung. Also, while *2D*-$\mathcal{L}_1$-$\mathcal{L}_{wGAN}$ using two views is already able to visualize structures within the lung representing blood vessels, these introduced structures do not correspond to the target image.

Four Views By utilizing four projection images as shown in Figure 6.9, *2D*-$\mathcal{L}_1$-$\mathcal{L}_{wGAN}$ is able to drastically reduce the amount of larger misplaced structures and thus improves the similarity to the target image. Both learning based methods were able to improve on the vertebra's shape, although neither the result generated from *2D*-$\mathcal{L}_1$-only nor *2D*-$\mathcal{L}_1$-$\mathcal{L}_{wGAN}$ looks like the target image's vertebra. The shape of the right bronchus has improved for both learning based methods and the darker area containing the left bronchus is now even better distinguishable from the rest of the lung area. While *2D*-$\mathcal{L}_1$-only starts to show some blood vessels in the lung, *2D*-$\mathcal{L}_1$-$\mathcal{L}_{wGAN}$ was able to improve on the placement and size of the blood vessels, they look more realistic but, as the rest of the image, do not correspond well to the target image.

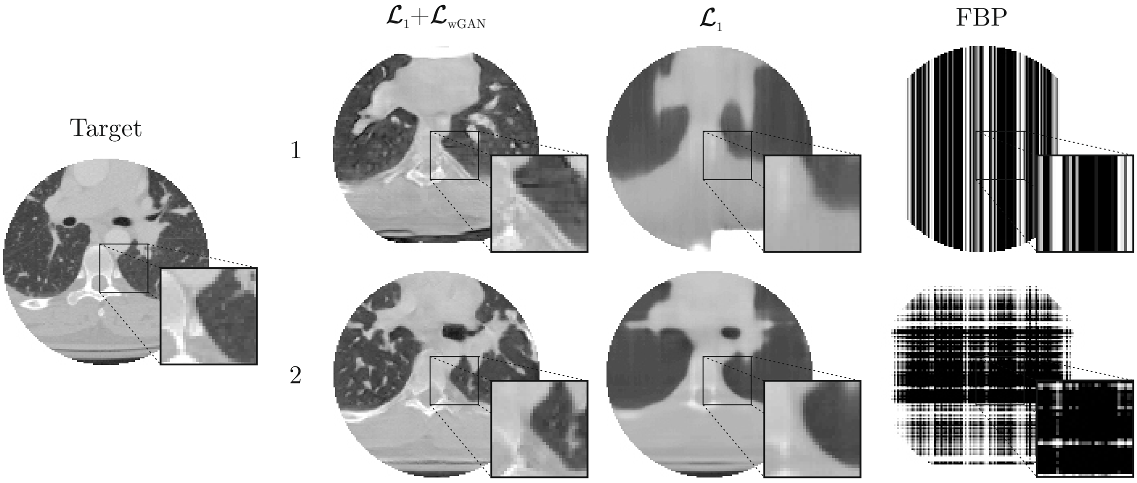


Figure 6.8: Qualtitative results of our *2D* methods and the *FBP* method for one and two views.

Six Views Further increasing the number of views from four to six projection images as shown in Figure 6.9 did not affect the results too much, while the overall anatomical structures improved for *2D*-$\mathcal{L}_1$-only, the reconstructed image generated by *2D*-$\mathcal{L}_1$-$\mathcal{L}_{wGAN}$ has a similar quality as the one generated using four views. *2D*-$\mathcal{L}_1$-only managed to visualize the left rib, which was clearly identified as a bone but, however, slightly misplaced. Both learning based methods are able to indicate the thin soft tissue border of the left bronchus and can distinguish the descending aorta from other soft tissues. Lastly, *2D*-$\mathcal{L}_1$-$\mathcal{L}_{wGAN}$ connected the right bronchus to the rightmost bronchus, although they are not connected in this slice.

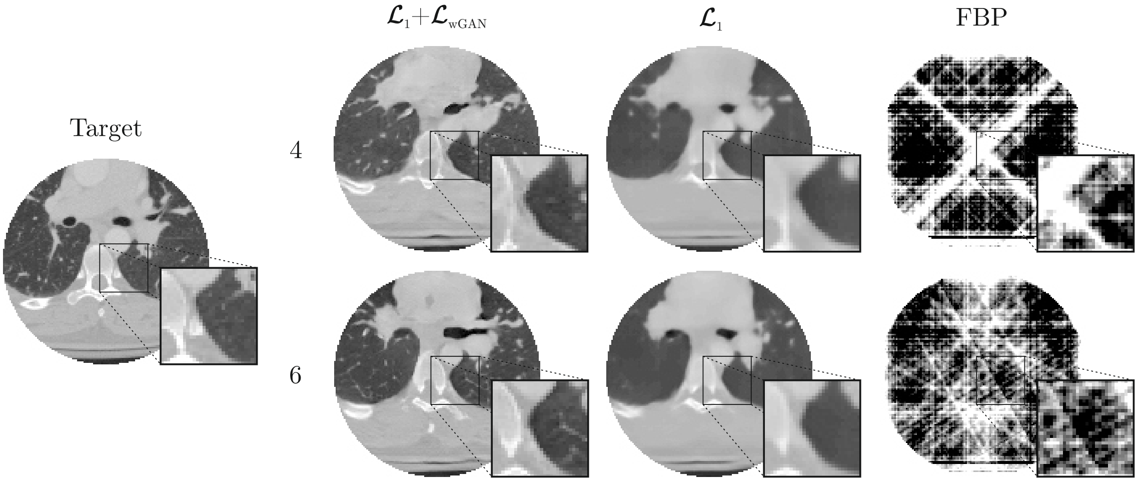


Figure 6.9: Qualtitative results of our *2D* methods and the *FBP* method for four and six views.

Eight Views All relevant structures of the vertebra are visible for *2D*-$\mathcal{L}_1$-only and *2D*-$\mathcal{L}_1$-$\mathcal{L}_{wGAN}$, when reconstruction is done utilizing eight projection images leading to believable vertebra shapes, see Figure 6.10. Furthermore, the left rib is represented well for both methods, which is also true for the left and right bronchus with the left being clearly distinguished from the rest of the lung by both *CNNs*. Also, the rightmost bronchus is now visualized by both methods and additionally, *2D*-$\mathcal{L}_1$-$\mathcal{L}_{wGAN}$ started to differentiate between single substructures of the heart. Both methods approximate larger blood vessels present in the projection images which can be matched to the larger anatomical structures of the target image, nevertheless, while the position of the blood vessels is close to the original position, the exact shape of these fine details is not and some structures are wrongly introduced by *2D*-$\mathcal{L}_1$-$\mathcal{L}_{wGAN}$. At this stage, the reconstructed images generated by both learning based methods are very similar to the target image where all larger structures are represented roughly correct resulting in a pretty clear correspondence to the target image, however, fine details are not reconstructed in a sufficient quality. Up to this point the *FBP* did not result in anything meaningful, however, by using eight views the *FBP* method managed to reduce the amount of streaking artifacts enough to allow the distinction between lung and other tissues.

15 Views Further increasing the number of views to 15 as shown in Figure 6.10 allows the *FBP* method to distinguish the lung and the bronchi from the soft tissues and bones and gives an indication of the vertebra, however, the reconstructed image is still heavily burdenend by streaking artifacts. In contrast to that, the learning based methods improved the quality of the reconstructed image to a point at which the correspondence to the target image is clearly given. In both, *2D*-$\mathcal{L}_1$-only and *2D*-$\mathcal{L}_1$-$\mathcal{L}_{wGAN}$, the bones, i.e. the vertebra as well as both ribs, the bronchi, the descending aorta and even some of the larger blood vessels in the lung are represented very well and contribute to a high similarity to the target image. While *2D*-$\mathcal{L}_1$-only also started to distinguish between the different substructures of the heart, *2D*-$\mathcal{L}_1$-$\mathcal{L}_{wGAN}$ shows further improvements in the heart area. Additionally, the head of rib is also visualized in both learning based methods and clearly separated from the vertebra, which is especially good visible in the result generated by *2D*-$\mathcal{L}_1$-$\mathcal{L}_{wGAN}$. Lastly, both methods managed to represent the spinous process of the vertebra located above.

30 Views The results generated by *2D*-$\mathcal{L}_1$-only and *2D*-$\mathcal{L}_1$-$\mathcal{L}_{wGAN}$ further improved in visualizing the different substructures of the heart when 30 projection views are utilized as given in Figure 6.11, however, some errors in this region are made, e.g. the misinterpretation by *2D*-$\mathcal{L}_1$-$\mathcal{L}_{wGAN}$ leading to a separation of the pulmonary artery. The fine structures of the blood vessels in the lung are represented very well by both learning based methods leading to correctly placed and very similarly shaped blood vessels of large and medium size. By using 30 projection views, the *FBP* method managed to clearly visualize the vertebra and the ribs. Furthermore, it is able to represent all bronchi as well as to

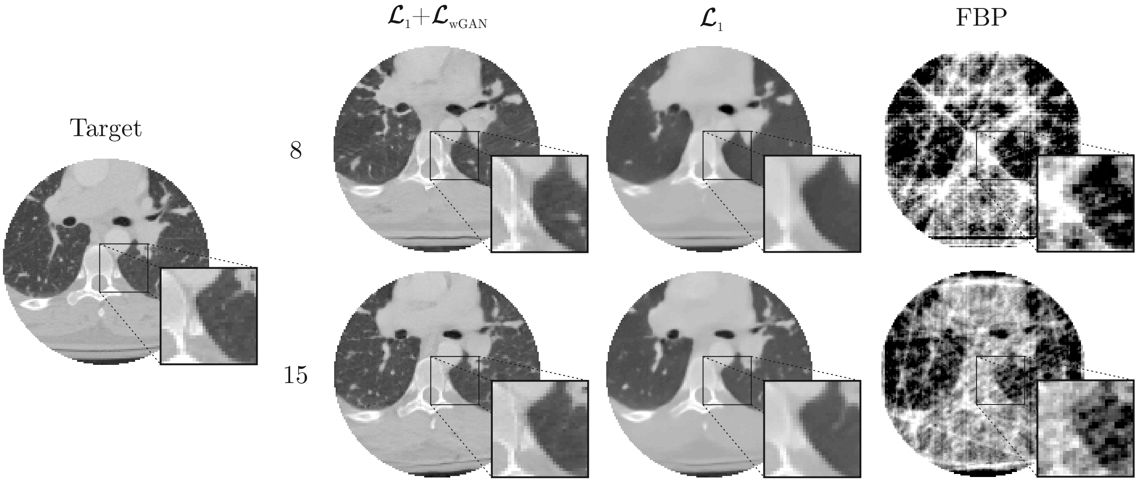


Figure 6.10: Qualtitative results of our *2D* methods and the *FBP* method for eight and 15 views.

separate the left bronchus from the lung and even shows some of the large blood vessels in the lung. However, the *FBP* method still contains many streaking artifacts and thus, a distinction between different soft tissues can not be done.

60 Views Both learning based methods managed to represent the various substructures of the heart by using 60 projection views as shown in Figure 6.11, especially the pulmonary artery is now displayed correctly. The *FBP* method greatly reduced the amount of introduced streaking artifacts allowing a well representation of the target image that manages to visualize the descending aorta and some finer blood vessels in the lung, however, other soft tissues can not be distinguished.

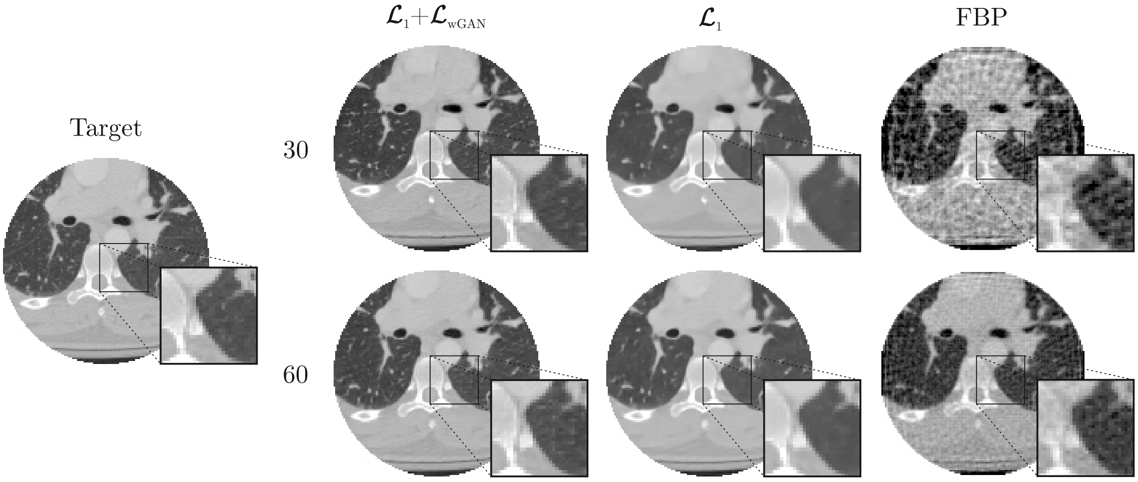


Figure 6.11: Qualtitative results of our *2D* methods and the *FBP* method for 30 and 60 views.

120, 180 and 360 Views By further increasing the number of views shown in Figure 6.12, the *FBP* is able to eliminate the presence of streaking artifacts allowing fine details to be visualized. When increasing to 120 views, the *FBP* method's result represents all discussed fine details like the distinction between the various substructures of the heart as well as medium sized blood vessels in the lung. However, while streaking artifacts are not present anymore, the reconstructed image still contains some artifacts similar to noise, which can be reduced by increasing the number of views to 180 and even further by using 360 views.

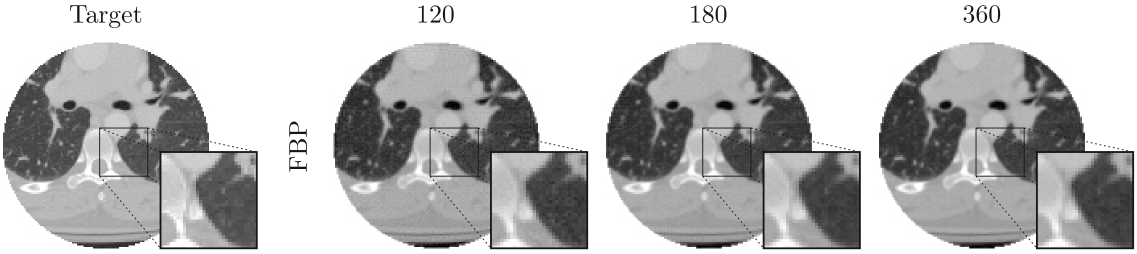


Figure 6.12: Qualtitative results of the *FBP* method for 120, 180 and 360 views.

6.2.2.3 Varying the Weight λ between Losses for a Selected Slice

The weight λ represents a hyperparameter which is required by our *2D*-$\mathcal{L}_1$-$\mathcal{L}_{wGAN}$ method and defines the contributed influence of the $\mathcal{L}_1$ loss and the $\mathcal{L}_{wGAN}$ loss to the utilized loss function defined in Equation (5.5). All results of *2D*-$\mathcal{L}_1$-$\mathcal{L}_{wGAN}$ we showed up to now have been generated using the default weight $\lambda = 10^{-3}$, however, we also experimented with different values. More precisely, we conducted experiments using different weights $\lambda \in 10^{\{-6,-5,-4,-3,-2,-1\}}$, where a low value for λ leads to a low contribution of the $\mathcal{L}_{wGAN}$ loss and a high value leads to a high contribution, shown in Figure 6.13. All reconstructed images using *2D*-$\mathcal{L}_1$-$\mathcal{L}_{wGAN}$ with a different λ as well as *2D*-$\mathcal{L}_1$-only have been generated using eight views, since a smaller number of views leads to a larger variance in the reconstructed images due to a higher uncertainty in value distribution.

Weight λ as 10^{-6} and 10^{-5} The results acquired using a small weight of $\lambda = 10^{-6}$ and $\lambda = 10^{-5}$ are due to the strong influence of the $\mathcal{L}_1$ loss very similar to the result generated by *2D*-$\mathcal{L}_1$-only. The only noticeable difference to *2D*-$\mathcal{L}_1$-only is that the reconstruction generated by $\lambda = 10^{-5}$ connected the right and rightmost bronchus. Both *2D*-$\mathcal{L}_1$-$\mathcal{L}_{wGAN}$ results using a $\lambda = 10^{-6}$ and $\lambda = 10^{-5}$ yield a very similar anatomical structure compared to the target slice, however, most of the fine details are lost due to oversmoothing yielding blurry looking reconstruction images.

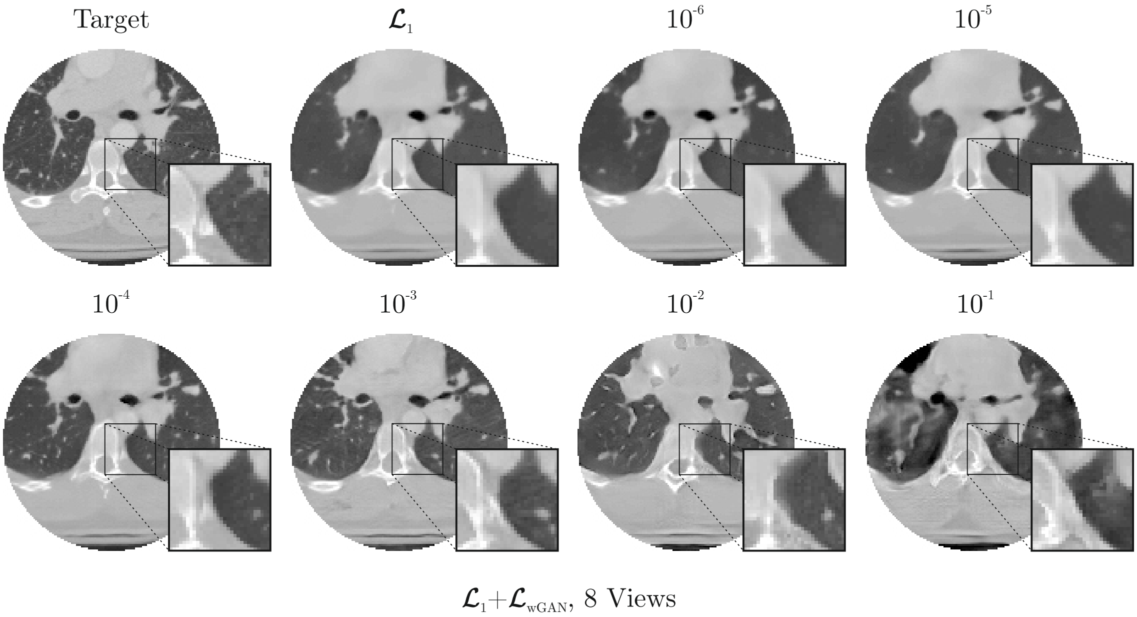


Figure 6.13: Qualtitative results of our *2D*-$\mathcal{L}_1$-$\mathcal{L}_{wGAN}$ method generated using different weights between the loss functions.

Weight λ as 10^{-4} and 10^{-3} Increasing the weight to $\lambda = 10^{-4}$ leads to sharper reconstructed images containing fine details and still yields a clear correspondence to the target slice. In contrast to smaller values of λ, some structures like the head of ribs become distinguishable from the vertebra. Utilizing $\lambda = 10^{-3}$ leads to even more distinguishable structures present in the reconstructed image, e.g. some subregions of the heart, while the correspondence to the target is still pretty clear.

Weight λ as 10^{-2} and 10^{-1} Further increasing the contribution of the $\mathcal{L}_{wGAN}$ loss by setting the weight to $\lambda = 10^{-2}$ leads to a diminishing anatomical correspondence to the target with too many *GAN* specific artifacts being present as well as newly introduced structures that are anatomically not feasible. Using a weight of $\lambda = 10^{-1}$ ultimately destructs the anatomical feasibility leading to bad and unrealistic reconstruction results due to the generative influence and possibilities being to large.

6.2.2.4 Evaluation of Additional Slices

The qualitative results we showed up to now have been limited to one selected slice. To give insight into the reconstruction quality of some additional slices as well, we extracted additional slices from the test data, which are presented in Figure 6.14. The reconstruction results have been generated by our *2D*-$\mathcal{L}_1$-only and *2D*-$\mathcal{L}_1$-$\mathcal{L}_{wGAN}$ method using eight and 30 projection views and a λ setting of 10^{-3}.

Eight Views By using eight projection views it is observable that the results generated using 2D-$\mathcal{L}_1$-only appear to be blurry, while the results generated by 2D-$\mathcal{L}_1$-$\mathcal{L}_{wGAN}$ achieve sharper images that contain more structural details. The correspondence of these structural details to the target image using just eight projection views is, however, not clear. Especially soft tissues cannot be reconstructed well when only eight projection views are used. High contrast structures like bones, the airways and the lung yield better reconstructions, however, some small bone-like structures have been introduced when looking at the results generated by 2D-$\mathcal{L}_1$-$\mathcal{L}_{wGAN}$ using eight projection views in the first row in Figure 6.14.

30 Views Increasing the number of projection views to 30 results in more reliable reconstructions of the target images for both, 2D-$\mathcal{L}_1$-only and 2D-$\mathcal{L}_1$-$\mathcal{L}_{wGAN}$. While high contrast regions yield a good resemblance of the target, low contrast regions like the soft tissues are reconstructed less accurately and also look more blurry when using 2D-$\mathcal{L}_1$-only. In contrast to that, 2D-$\mathcal{L}_1$-$\mathcal{L}_{wGAN}$ achieves sharper reconstruction results.

6.2.2.5 Evaluation of the Coronal and Sagittal View

As explained in Section 5.1.1, for our 2D method we extracted the axial 2D CT slices and reconstructed each slice independently of which we presented the results in the preceding sections. As a next step, these independently reconstructed 2D axial CT slices can be stacked to yield a complete 3D CT volume that represents the patient. In this section, we evaluate the integrity of these 3D CT volumes generated by stacking the reconstructed 2D CT slices by looking at the coronal and sagittal view, which are orthogonal to the axial view as visualized in Figure 2.10. We compare these orthogonal views generated from results using eight and 30 projection views reconstructed by 2D-$\mathcal{L}_1$-$\mathcal{L}_{wGAN}$ and 2D-$\mathcal{L}_1$-only.

Eight Views Using eight projection views leads to clear discontinuities in the stacked image as shown in Figure 6.15, which is especially visible in the sagittal view when looking at the spine. The result generated by our 2D-$\mathcal{L}_1$-$\mathcal{L}_{wGAN}$ method suffers more from discontinuity than 2D-$\mathcal{L}_1$-only due to yielding sharper images that look less blurry. Additionally, these discontinuities seem to be strongest for slices that also contain larger parts of the lung which becomes apparent when comparing individual regions of the spine in sagittal view to the corresponding position in coronal view.

30 Views Increasing the number to 30 projection views as shown in Figure 6.16 solves the problem of discontinuity for both, 2D-$\mathcal{L}_1$-$\mathcal{L}_{wGAN}$ and 2D-$\mathcal{L}_1$-only. When again observing the spine in sagittal view, the result generated by either method now looks continuous and both appear sharp yielding a visually good reconstruction of the target image.

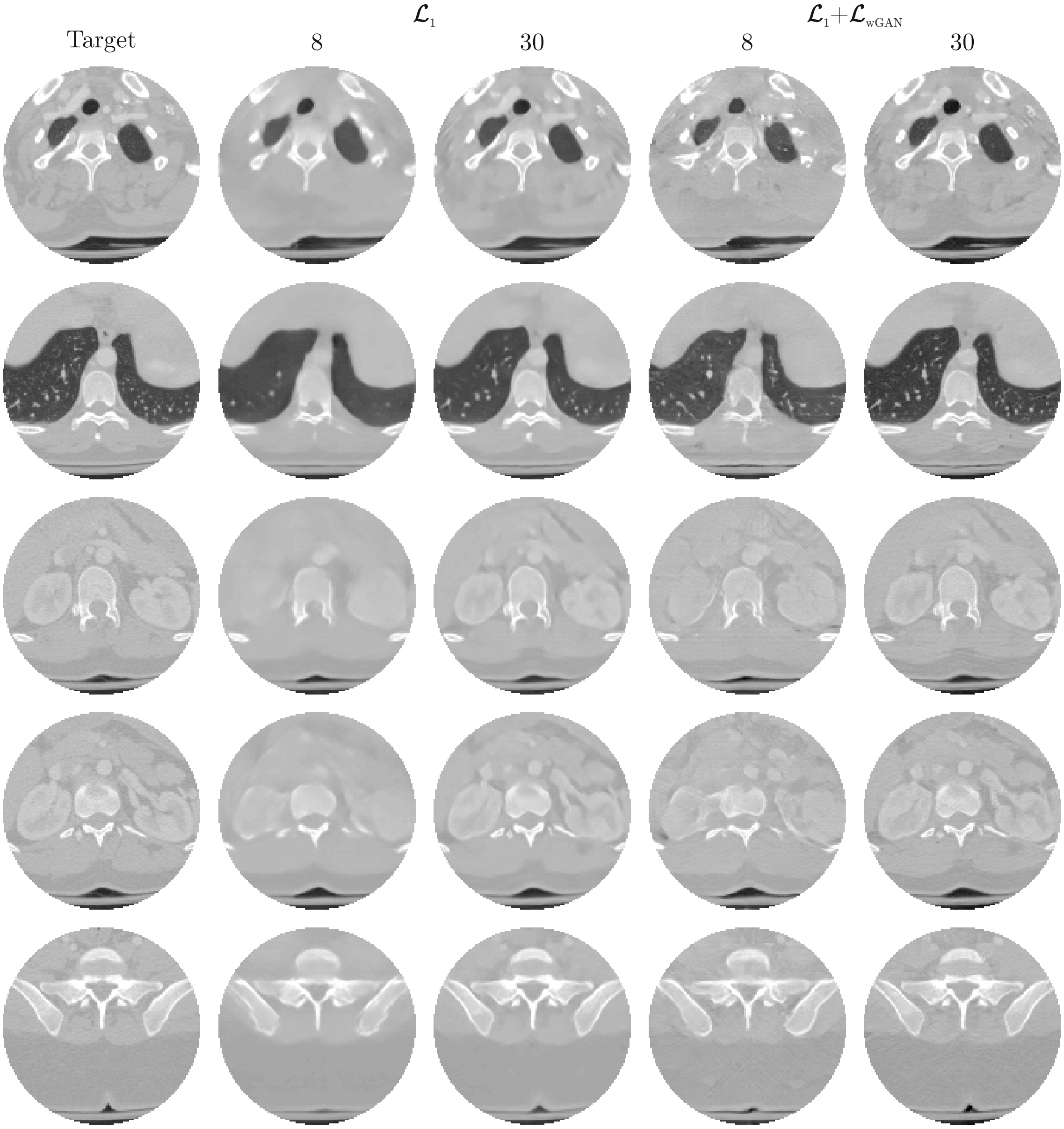


Figure 6.14: Qualtitative results of our *2D* methods showing additional slices for eight and 30 views.

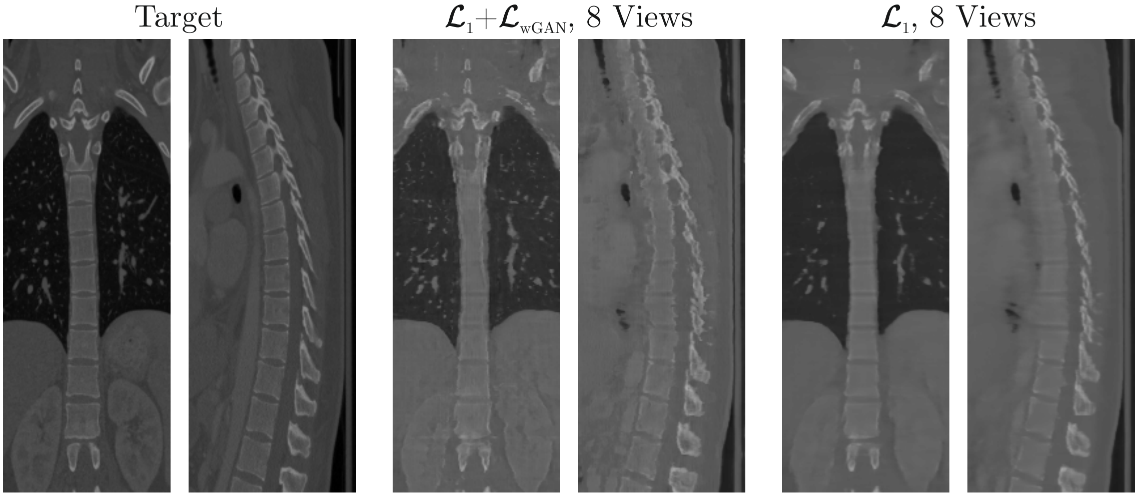


Figure 6.15: Qualtitative results of our *2D* methods when showing the coronal and sagittal view of the stacked 2D results using eight views.

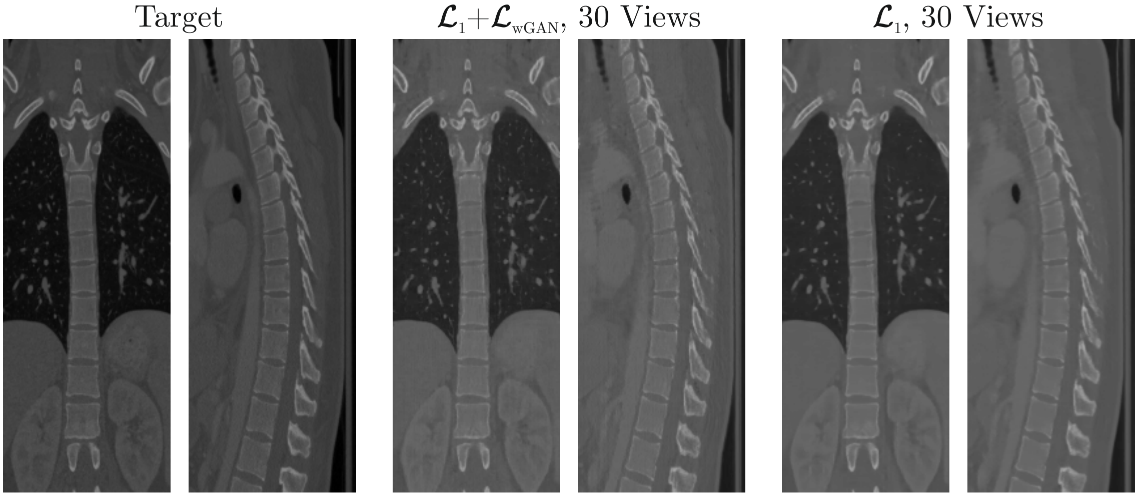

Target
$\mathcal{L}_1+\mathcal{L}_{\mathrm{wGAN}}$, 30 Views
$\mathcal{L}_1$, 30 Views

Figure 6.16: Qualtitative results of our *2D* methods when showing the coronal and sagittal view of the stacked 2D results using 30 views.

6.2.2.6 Evaluation of a Site Unseen During Training

As of now, we have evaluated the performance of our methods using *CT* slices taken from the same region as the networks saw during training. In this section, we give insight into the quality of the reconstruction results of *CT* slices acquired from a different site. As such, we reused the networks from above that have been trained on *CT* slices from the thoracic and abdominal region to generate the reconstruction results of head *CT* slices presented in this section. We compare the results generated by *2D*-$\mathcal{L}_1$-$\mathcal{L}_{wGAN}$ and *2D*-$\mathcal{L}_1$-only using one, eight, 30 and 60 views to the target *CT* image.

One View The results generated using just one view shown in Figure 6.17 and Figure 6.18 do not carry any meaningful information independently of the method used for reconstruction. Both methods, *2D*-$\mathcal{L}_1$-$\mathcal{L}_{wGAN}$ and *2D*-$\mathcal{L}_1$-only seem to reproduce an image that looks more similar to those the networks saw during training.

Eight Views When the number of views used to reconstruct the head *CT* image is increased to eight, the reconstruction results appear to be more meaningful and roughly resemble the target image, see Figure 6.17 and Figure 6.18. However, the reconstructed results are still very bad and contain a lot of flaws.

30 Views Using 30 projection views as an input for *2D*-$\mathcal{L}_1$-$\mathcal{L}_{wGAN}$ and *2D*-$\mathcal{L}_1$-only to reconstruct a *CT* image improves the quality of the reconstructed head *CT* slices. While the reconstruction result of the more complex head *CT* slice shown in Figure 6.17 is burdened by many artifacts, the result of the less complex head *CT* slice in Figure 6.18 is visually very similar to the target image yielding a good reconstruction.

60 Views The results generated from 60 projection views improved the sharpness of the reconstructed head *CT* images as shown in Figure 6.17 and Figure 6.18. However, both *2D*-$\mathcal{L}_1$-$\mathcal{L}_{wGAN}$ and *2D*-$\mathcal{L}_1$-only still introduce artifacts to the reconstructed image and also introduced an additional artifact to the result in Figure 6.18.

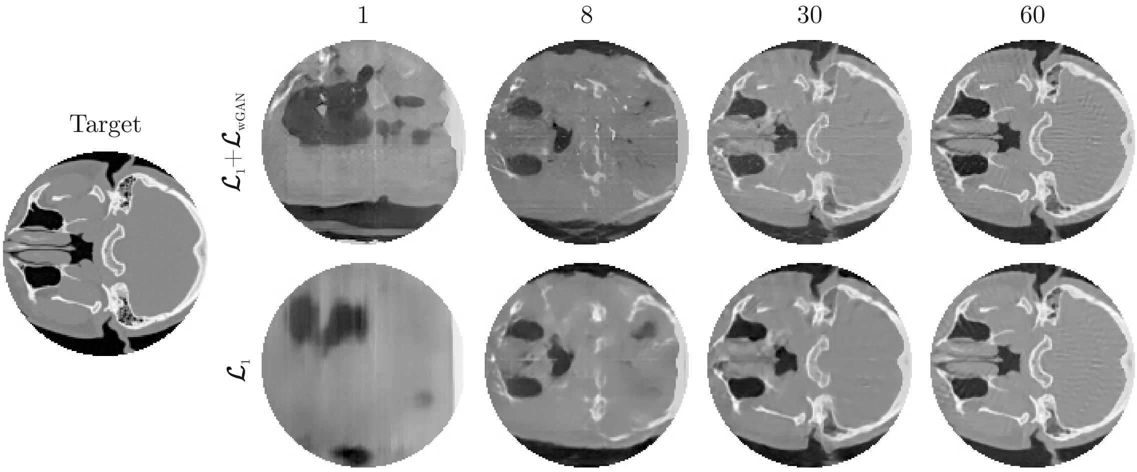


Figure 6.17: Qualtitative results of our *2D* methods when evaluating a site unseen during traing using a different number of views.

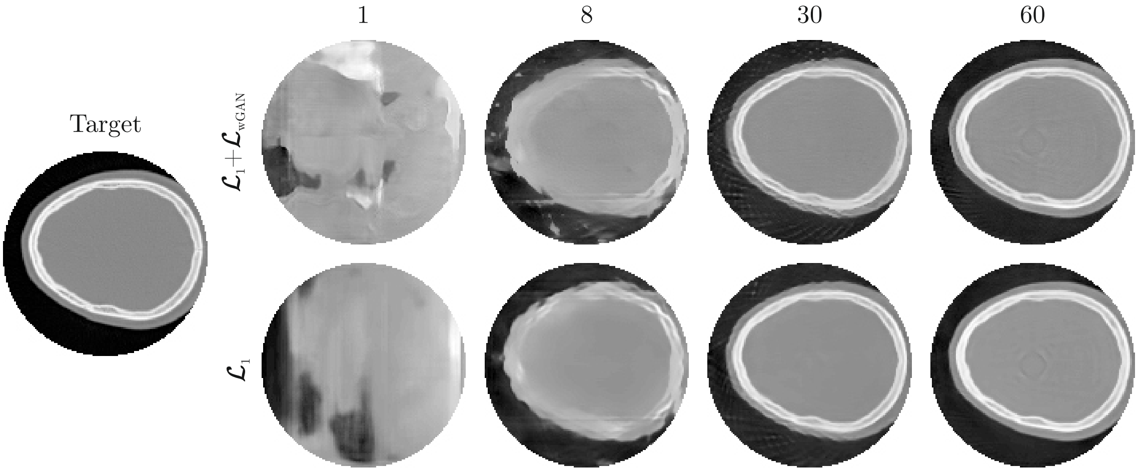


Figure 6.18: Qualtitative results of our *2D* methods when evaluating a site unseen during traing using a different number of views.

7

Discussion

Contents

7.1 Comparison to Related Work 99
7.2 Discussion of Our Method 101
7.3 Future Work 104

Motivated by reducing the amount of ionizing radiation exposed to the patient during Computed Tomography (CT) imaging, we proposed a sparse-view reconstruction method that reduces the number of projection views used to reconstruct a *CT* image, see Chapter 5. Highly reducing the number of projection views leads to a violation of the Nyquist-Shannon sampling theorem explained in Section 2.2.3, which we aim to compensate by using a Convolutional Neural Network (CNN).

7.1 Comparison to Related Work

While *CT* reconstruction in clinical practice still relies on classical reconstruction methods like the analytical Filtered Backprojection (FBP) method, most of the research conducted in the field of medical image reconstruction that was recently published is based on deep learning and especially *CNNs*, see Chapter 4. These approaches can be categorized by distinguishing between them according to the implemented strategy of applying deep learning as well as by the approach used to actually reduce the amount of ionizing radiation exposure.

7.1.1 Strategies to Apply Deep Learning

Most approaches follow the strategy of applying a *CNN* to learn image-to-image reconstruction as explained in Section 4.1.2, which means that the *CNN* receives an already reconstructed low-dose *CT* image and learns an image enhancement procedure to improve the reconstruction quality. Consequently, this type of approaches still relies on a classical reconstruction method like the *FBP* method to provide the input for the *CNN*. Since classical reconstruction methods are known to heavily introduce artifacts when the Nyquist-Shannon sampling theorem is violated, we argue that a *CNN* following the image-to-image reconstruction strategy is required to learn removing the artifacts that have been introduced by the classical reconstruction method.

In contrast to that, our method falls into the category of learning sinogram-to-image reconstruction as given in Section 4.1.4 and as such, our *CNN* directly learns *CT* image reconstruction from the projection data circumventing the need of using any other reconstruction method. Thus, the strategy of sinogram-to-image reconstruction does not suffer from any classical reconstruction method, which we view as an advantage over the image-to-image reconstruction strategy.

7.1.2 Types of Approaches to Reduce Ionizing Radiation

We identified three different types of approaches to accomplish a reduction of ionizing radiation exposure in literature. The first type of approaches is represented by reducing the tube current used when acquiring the projection views as explained in Section 4.2. Since tube current reduced approaches acquire the full projection data, they homogeneously degrade the quality of the reconstructed image when using classical reconstruction methods and to the best of our knowledge, exclusively follow the strategy of image-to-image reconstruction.

A second type of approaches uses beam blockers that act as physical barriers between the X-ray tube and the patient as described in Section 4.3. To be of practical use, this type of approaches requires to define the properties of physical beam blockers which then need to be implemented in *CT* scanners representing an additional difficulty in comparison to the other types of approaches. As of now, beam blocking based approaches are rarely found in Fan Beam Computed Tomography (FBCT) and in most cases used in Cone Beam Computed Tomography (CBCT) to increase the contrast by reducing the amount of beam scatter which is typical to *CBCT*.

The method we proposed falls into the last type of approaches to reduce the amount of ionizing radiation exposure called sparse-view, where the number of projection views is reduced as explained in Section 4.4. Many sparse-view approaches are based on the strategy image-to-image reconstruction and thus, rely on a classical reconstruction method like the *FBP* method which leads to the introduction of streaking artifacts that are inhomogeneous. Since the sparsity of the data leads to inhomogeneously distributed artifacts, we argue that the strategy of image-to-image reconstruction is suboptimal for the type of sparse-view reconstruction methods. Furthermore, sparse-view approaches only require to reduce the number of projection views that are acquired during *CT* imaging which can easily be implemented by defining a sparse-view protocol for image acquisition. In contrast to beam blocking based approaches, changes to the actual hardware are not necessary for sparse-view reconstruction approaches to be of practical use.

7.2 Discussion of Our Method

Many other low-dose *CT* reconstruction approaches that rely on a classical reconstruction technique like the *FBP* aim to remove the artifacts that have been introduced by the classical reconstruction technique. In contrast to that, our method directly learns *CT* image reconstruction from the projection data and circumvents the need of using any classical reconstruction technique. Since publicly available datasets typically do not incorporate the projection data used for *CT* image reconstruction, we decided to simulate the projection views by generating them from the normal-dose *CT* image to proof the concept of our approach. Beneficially, since the number of data samples in medical imaging is usually very low, simulating the projection data allowed us to use data augmentation of the target *CT* images without invalidating the correspondence to the projection data and also allowed us to generate the projection views from any arbitrary angle. To generate the simulated projection data, we implemented a procedure that is very similar to real data acquisition in *CT* imaging explained in Section 5.2.

We conducted a variety of different experiments to evaluate our approach and the performance of the trained *CNNs*. First, we implemented our approach using three dimensional (3D) data, where the *CNN* optimizes the $\mathcal{L}_1$ loss function and directly learns to reconstruct a *3D CT* volume. However, the vast execution time and memory requirements when training *CNNs* to directly learn *3D CT* image reconstruction represents a limitation that forced us to reduce the resolution of the reconstructed images to $64 \times 64 \times 64$. To increase the resolution of the images we adapted our initial approach to use two dimensional (2D) *CT* slices instead of *3D CT* volumes, which allowed us to increase the resolution to 128×128 as well as to increase the number of training iterations of the *CNN*. Additionally, changing to *2D* allowed us to extend the *CNN* used in our approach to a Generative Adversarial Network (GAN) which uses a combined loss function consisting of a content $\mathcal{L}_1$ and an adversarial $\mathcal{L}_{wGAN}$ loss coming from a Wasserstein Generative Adversarial Network (WGAN) as proposed in [28]. While some other approaches nowadays also rely on the contribution of a *GAN* to a combined loss function, in contrast to our approach, most of them also rely on the strategy of image-to-image reconstruction and additionally fall into the category of tube current reduced reconstruction approaches.

7.2.1 3D Experiments

The quantitative results of our *3D* experiments shown in Section 6.1.1 demonstrate that our *CNN* based *3D*-$\mathcal{L}_1$-only method performs significantly better than the non-learning based *FBP* method when heavily reducing the number of projection views available for *CT* image reconstruction. In contrast to the *FBP* method, our *3D*-$\mathcal{L}_1$-only method benefits from prior knowledge that was acquired beforehand by learning from the training data. Increasing the number of views available to both methods reduces the gap in performance between them to a point where the reconstruction quality can be considered the same which is reached when using 60 views and more.

This observation is supported by our qualitative results presented in Section 6.1.2, the results generated by our *3D*-$\mathcal{L}_1$-only method as well as by the *FBP* method using 60 views are visually very similar to one another and represent good reconstructions of the target. When using a smaller number of views, the *FBP* method starts to introduce streaking artifacts degrading the quality of the reconstructed image, while our method is able to achieve a better visual reconstruction quality in comparison to the *FBP* method. E.g., when using eight views for reconstruction, our *3D*-$\mathcal{L}_1$-only method is able to visualize all relevant structures of the vertebra, whereas the image content of the *FBP* method reconstructed image is barely recognizable due to the amount of introduced streaking artifacts.

We show that our *CNN* based sparse-view *CT* reconstruction method is able to achieve a good reconstruction quality from a very small number of projection views without being heavily burdened by artifacts as the *FBP* method is. When using a very small number of views the reconstruction results of our method, however, look increasingly blurry loosing more and more details which we aimed to improve by using a combined loss function which we implemented for our *2D* pipeline.

7.2.2 2D Experiments

The quantitative results of our *2D* experiments show that both *2D*-$\mathcal{L}_1$-only and *2D*-$\mathcal{L}_1$-$\mathcal{L}_{wGAN}$ perform significantly better than the *FBP* method when using a reduced number of projection views to reconstruct the target image, see Section 6.2.1. The observation of the Mean Absolute Error (MAE) shows that *2D*-$\mathcal{L}_1$-only performs slightly better than *2D*-$\mathcal{L}_1$-$\mathcal{L}_{wGAN}$ which is to be expected, since *2D*-$\mathcal{L}_1$-only is directly optimized to minimize $\mathcal{L}_1$ loss, i.e. *MAE*. The Structural Similarity Index Metric (SSIM) result demonstrates that *2D*-$\mathcal{L}_1$-only achieves the best results up to eight projection views, while from that point on, the results generated by *2D*-$\mathcal{L}_1$-only and *2D*-$\mathcal{L}_1$-$\mathcal{L}_{wGAN}$ can be considered equivalent.

When looking at the qualitative results shown in Section 6.2.2, we can see that the *FBP* method increasingly suffers from the introduction of streaking artifacts when reducing the number of projection views. In contrast to that, our *2D* methods are able to visualize the overall structure of the given slice and yield visually more appealing and more realistic looking images. However, reducing the number of views available for reconstruction leads to increasingly blurry looking images when reconstructing using only $\mathcal{L}_1$ loss, while an additional $\mathcal{L}_{wGAN}$ loss leads to sharper reconstruction results that may contain deviated anatomical structures.

When experimenting with a different contribution of the adversarial $\mathcal{L}_{wGAN}$ loss to the combined loss function by changing the weight λ and using the same number of projection views, it becomes apparent that a higher contribution of the adversarial loss leads to a higher disruption of the anatomical correspondence to the target image as shown in Section 6.2.2.3. As such, the *2D*-$\mathcal{L}_1$-$\mathcal{L}_{wGAN}$ reconstructed images gradually lose the correspondence to the target image and in extreme cases, the reconstructed images even become anatomically infeasible. However, when weight λ is chosen carefully, the combined loss function and the influence of the *GAN* allows our *2D*-$\mathcal{L}_1$-$\mathcal{L}_{wGAN}$ method to result in sharper images that look more realistic compared to the results generated using *2D*-$\mathcal{L}_1$-only and still show a clear correspondence to the target image with only minor discrepancies.

Since our *2D* methods learn to reconstruct axial *2D CT* slices independently from one another, we also evaluated the reconstruction quality along the orthogonal axis of the axial slice in Section 6.2.2.5. To conduct this evaluation, we independently reconstructed all axial *2D CT* slices of a *3D CT* volume and stacked them after reconstruction as such, that they again yield a *3D CT* volume. By looking at the coronal as well as the sagittal view, this procedure allows to evaluate the integrity of the reconstruction results in *3D*. The result generated by *2D*-$\mathcal{L}_1$-$\mathcal{L}_{wGAN}$ using eight projection views shows larger discontinuities compared to the target than the result generated by *2D*-$\mathcal{L}_1$-only, however, the result of *2D*-$\mathcal{L}_1$-only is less sharp leading to an obfuscation of the discontinuities. This observation confirms that using $\mathcal{L}_1$ loss leads to a more consistent anatomy, while an additional $\mathcal{L}_{wGAN}$ loss yields a sharper but not necessarily anatomically consistent result. When increasing the number of projection views to 30, both *2D* methods are able to resolve the problem of discontinuity. We argue that the required number of views to solve the problem of discontinuity can be reduced using *3D* methods that provide *3D* data to the *CNN*, like our *3D*-$\mathcal{L}_1$-only method. However, since using *3D* data is very resource intensive, they become infeasible when large image resolutions are involved.

Another experiment we conducted is based on evaluating the performance of our *CNNs* when images from other sites that have been unseen during training are provided as an input. In Section 6.2.2.6 we reused the *CNNs* that have been trained using *2D CT* slices from the thorax and the abdomen and observed their reconstruction quality when the projection data from *2D CT* slices from the head are used as an input. The results generated by *2D*-$\mathcal{L}_1$-$\mathcal{L}_{wGAN}$ and *2D*-$\mathcal{L}_1$-only using a very low number of projection views showed to be bad with both methods having severe difficulties to yield a reconstruction that is not heavily burdened by artifacts. Increasing the number of views to 30 or 60 shows, that both methods are able to yield a reconstruction that resembles the target, however, slices that contain more complex anatomical structures are still burdened by streaking-like artifacts. In contrast to that, slices that contain less complex anatomical structures can be reconstructed better, but still suffer from artifacts in some cases, which leads to the conclusion that our method can not be directly used on data containing sites that have never been seen by the *CNN* during training. Fine-tuning the *CNNs* using data from the target site represents a possible solution to this problem.

7.3 Future Work

Up to now we evaluated our method only using simulated projection data, as such, in our future work we aim to conduct an evaluation when real projection data is used. This, however, opposes the problem that publicly available data does typically not include the projection data used for reconstruction that is required by our *CNN*. For those datasets that include the real projection data, it is very questionable whether they contain enough data samples for our *CNN* to optimize considering that data augmentation possibilities of real projection data are very limited. However, since real and simulated projection data is very similar, this problem can be circumvented by extending the method proposed in this thesis to use an additional fine-tuning step. As such, the *CNN* can be trained on a large amount of simulated projection data, while a small amount of real projection data is sufficient in the fine-tuning step, where the *CNN* is optimized to adapt the features learned from simulated projection data to real projection data. Additionally, to improve the integrity of our *2D* methods in *3D*, we also plan to investigate the performance of a multi-slice approach, where the *CNN* still learns to reconstruct one axial *2D* *CT* slice but is also provided with the projection data of neighboring slices. To further investigate potential real world applications, we also aim to investigate the performance of our method for registration tasks as well as it's performance on locating instruments within the body of a patient, which represents an important task during minimally invasive and image guided surgery. We also plan to contact radiologists to evaluate the reconstruction results generated by our method to aquire a professional and objective opinion on their quality. Finally, we aim to evaluate our method on other datasets consisting of more data samples and also on datasets that contain other sites or multiple different sites.

8

Conclusion

We proposed a machine learning based sparse-view Computed Tomography (CT) reconstruction method from a reduced number of projection views that achieves decent reconstruction results even from undersampled data according to the Nyquist-Shannon sampling theorem and compressed sensing. Many other approaches rely on traditional reconstruction algorithms like the Filtered Backprojection (FBP) method, utilize already reconstructed sparse-view *CT* images and learn to improve the quality of these images by removing the undersampling artifacts that have been introduced when reconstructing the *CT* image using the traditional algorithm. In contrast to that, our approach directly utilizes the projection data to train a Convolutional Neural Network (CNN) that learns to reconstruct the *CT* image itself without relying on any traditional reconstruction algorithm.

The data we utilized consists of ten already reconstructed *CT* images, however, since our method requires not only access to the already reconstructed normal-quality *CT* images but also to the corresponding projection data, we simulated the projection data by generating it from the already reconstructed *CT* images from different angles on the axial plane similarly to *CT* scanners. Simulating the projection data gave us two benefits, first it allowed us to augment the data by rotation, translation and scale completely at will increasing the variety of the available data samples which is important when training *CNNs* to counteract overfitting. Furthermore, the simulation of the projection data enabled us to freely experiment with the number of projection views and the angles from which they are utilized which was beneficial for this work we consider a proof of concept.

The generated projection images are utilized as the input for our *CNN*, while the corresponding *CT* image from which the projection images have been generated is treated as the target image for our *CNN* to optimize. We experimented with different dimensionalities of the image data separately utilizing two dimensional (2D) and three dimensional (3D) *CT* images optimized on $\mathcal{L}_1$ loss. Additionally, on our *2D* data we tested the performance of a combined loss function consisting of a content $\mathcal{L}_1$ loss and an adversarial $\mathcal{L}_{wGAN}$ loss that comes from a Generative Adversarial Network (GAN). For convenience, these methods are called *3D*-$\mathcal{L}_1$-only, *2D*-$\mathcal{L}_1$-only and *2D*-$\mathcal{L}_1$-$\mathcal{L}_{wGAN}$ respectively. The evaluation was conducted by comparing the results generated by our method quantitatively and qualitatively to the well known *FBP* method as well as to each other, however, the results generated by our *2D* and *3D* method have not directly been compared due to larger differences in hyperparameters that would lead to an unfair comparison.

The quantitative results show that our approach performs significantly better than the *FBP* method when reconstructing the *CT* images from heavily undersampled projection data. The qualitative results reveal that the *FBP* method gradually suffers from streaking artifacts when reducing the number of projection views. In contrast to that, the results generated by our methods optimized on $\mathcal{L}_1$ loss convey more useful information than the *FBP* method's results, but look gradually blurrier when the number of projection views is reduced, while the results of *2D*-$\mathcal{L}_1$-$\mathcal{L}_{wGAN}$ look sharper and more realistic. However, the results generated by our *GAN* based method gradually lose the anatomical correspondence to the target image potentially causing the introduction artifacts that look similar to anatomical structures.

We conclude that the results generated from a highly reduced number of views utilizing our methods and especially our *GAN* based method *2D*-$\mathcal{L}_1$-$\mathcal{L}_{wGAN}$ have to be treated with caution especially when used for diagnosis, since particularly small and subtle anatomical details of the target image can not be guaranteed to be reconstructed correctly from highly undersampled data with a large amount of missing information. When a high undersampling rate is used, the *FBP* method heavily introduces streaking artifacts, while our methods optimized on $\mathcal{L}_1$ prefer blurring and our *2D*-$\mathcal{L}_1$-$\mathcal{L}_{wGAN}$ method potentially introduces anatomical structures that do not correspond to the target image. As such, while our *CNN* based methods are able to improve the quality of the reconstructed images, they still suffer from the sparse information available when using high undersampling rates but, in contrast to the *FBP* method, our methods learned to optimize the visual representation of the reconstructed *CT* slice. While it remains questionable whether or not our learning based methods can be utilized in diagnostic clinical practice, we see practical applications in image registration tasks where a reduced amount of ionizing radiation exposure to the patient is necessary or only a reduced number of projection views can be acquired and a perfect reconstruction is not necessary. An exemplary application is represented by locating the instruments inside of the patient during minimally invasive and image guided surgeries.

A

List of Acronyms

1D	one dimensional
2D	two dimensional
3D	three dimensional
ADAM	Adaptive Moment Estimation
ANN	Artificial Neural Network
ART	Algebraic Reconstruction Technique
CBCT	Cone Beam Computed Tomography
CNN	Convolutional Neural Network
CS	Compressed Sensing
CSI	Computational Methods and Clinical Applications for Spine Imaging
CT	Computed Tomography
DFR	Direct Fourier Reconstruction
EM	Earth Mover
FBCT	Fan Beam Computed Tomography
FBP	Filtered Backprojection
FNN	Feedforward Neural Network
FT	Fourier transform
GAN	Generative Adversarial Network
GD	Gradient Descent
HU	Hounsfield Unit
Hz	Hertz
IFT	Inverse Fourier transform
JS	Jensen-Shannon
KL	Kullback-Leibler
kVp	kilovoltage peak
Leaky ReLU	Leaky Rectified Linear Unit

MAE	Mean Absolute Error
MICCAI	Medical Image Computing and Computer Assisted Intervention
MRI	Magnetic Resonance Imaging
MSCT	Multi-Slice Computed Tomography
MSE	Mean Squared Error
MVUS	Many-View Undersampling
PBCT	Parallel Beam Computed Tomography
PGD	Projected Gradient Descent
ReLU	Rectified Linear Unit
SGD	Stochastic Gradient Descent
SSIM	Structural Similarity Index Metric
TV	Total Variation
VGG	Visual Geometry Group
VMAT	Volumetric Modulated Arc Therapy
WGAN	Wasserstein Generative Adversarial Network
WGAN-GP	Wasserstein Generative Adversarial Network with Gradient Penalty

B

List of Publications

Work presented in this master's thesis led to the following peer-reviewed publications. For the sake of completeness of this thesis, they are listed in chronological order along with the respective abstracts.

B.1 2018

Volumetric Reconstruction from a Limited Number of Digitally Reconstructed Radiographs Using CNNs

Franz Thaler, Christian Payer and Darko Štern
In: *Proceedings of the OAGM Workshop 2018*
May 2018, Hall/Tyrol, Austria
(Accepted for oral presentation)

Abstract: We propose a method for 3D computed tomography (CT) image reconstruction from 3D digitally reconstructed radiographs (DRR). The 3D DRR images are generated from 2D projection images of the 3D CT image from different angles and used to train a convolutional neural network (CNN). Evaluating with a different number of input DRR images, we compare our resulting 3D CT reconstruction to those of the filtered backprojection (FBP), which represents the standard method for CT image reconstruction. The evaluation shows that our CNN based method is able to decrease the number of projection images necessary to reconstruct the original image without a significant reduction in image quality. This indicates the potential for accurate 3D reconstruction from a lower number of projection images leading to a reduced amount of ionizing radiation exposure during CT image acquisition.

B.2 2018

Sparse-View CT Reconstruction Using Wasserstein GANs

Franz Thaler, Kerstin Hammernik, Christian Payer, Martin Urschler and Darko Štern
In: *MICCAI Workshop on Machine Learning for Medical Image Reconstruction - MLMIR 2018*
September 2018, Granada, Spain
(Accepted for poster presentation)

Abstract: We propose a 2D computed tomography (CT) slice image reconstruction method from a limited number of projection images using Wasserstein generative adversarial networks (wGAN). Our wGAN optimizes the 2D CT image reconstruction by utilizing an adversarial loss to improve the perceived image quality as well as an L_1 content loss to enforce structural similarity to the target image. We evaluate our wGANs using different weight factors between the two loss functions and compare to a convolutional neural network (CNN) optimized on L_1 and the Filtered Backprojection (FBP) method. The evaluation shows that the results generated by the machine learning based approaches are substantially better than those from the FBP method. In contrast to the blurrier looking images generated by the CNNs trained on L_1, the wGANs results appear sharper and seem to contain more structural information. We show that a certain amount of projection data is needed to get a correct representation of the anatomical correspondences.

Bibliography

[1] Adler, J. and Öktem, O. (2017). Solving ill-posed inverse problems using iterative deep neural networks. *Inverse Problems*, 33(12):124007. (page 59)

[2] Adler, J. and Öktem, O. (2018). Learned primal-dual reconstruction. *IEEE transactions on medical imaging*, 37(6):1322–1332. (page 59)

[3] Ambrose, J. (1973). Computerized transverse axial scanning (tomography): Part 2. Clinical application. *The British journal of radiology*, 46(552):1023–1047. (page 5)

[4] Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., Casper, J., Catanzaro, B., Cheng, Q., Chen, G., et al. (2016). Deep speech 2: End-to-end speech recognition in english and mandarin. In *International Conference on Machine Learning*, pages 173–182. (page 29)

[5] Anirudh, R., Kim, H., Thiagarajan, J. J., Aditya Mohan, K., Champley, K., and Bremer, T. (2018). Lose the views: Limited angle CT reconstruction via implicit sinogram completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6343–6352. (page 60)

[6] Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pages 214–223. (page 40, 42, 43, 44)

[7] Ba, J. L., Kiros, J. R., and Hinton, G. E. (2016). Layer normalization. *arXiv preprint arXiv:1607.06450*. (page 44)

[8] Barrett, J. F. and Keat, N. (2004). Artifacts in CT: Recognition and avoidance. *Radiographics*, 24(6):1679–1691. (page 24)

[9] Boone, J. M., Nelson, T. R., Lindfors, K. K., and Seibert, J. A. (2001). Dedicated breast CT: Radiation dose and image quality evaluation. *Radiology*, 221(3):657–667. (page 7, 25)

[10] Bracewell, R. N. (1956). Strip integration in radio astronomy. *Australian Journal of Physics*, 9(2):198–217. (page 21)

[11] Brenner, D. J. and Hall, E. J. (2007). Computed tomography - an increasing source of radiation exposure. *New England Journal of Medicine*, 357(22):2277–2284. (page 1, 6, 24)

[12] Chambolle, A. and Pock, T. (2011). A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of mathematical imaging and vision*, 40(1):120–145. (page 59)

[13] Chen, H., Zhang, Y., Chen, Y., Zhang, J., Zhang, W., Sun, H., Lv, Y., Liao, P., Zhou, J., and Wang, G. (2018). LEARN: Learned experts' assessment-based reconstruction network for sparse-data CT. *IEEE transactions on medical imaging.* (page 59)

[14] Chen, H., Zhang, Y., Kalra, M. K., Lin, F., Chen, Y., Liao, P., Zhou, J., and Wang, G. (2017a). Low-dose CT with a residual encoder-decoder convolutional neural network. *IEEE transactions on medical imaging*, 36(12):2524–2535. (page 54)

[15] Chen, H., Zhang, Y., Zhang, W., Liao, P., Li, K., Zhou, J., and Wang, G. (2017b). Low-dose CT via convolutional neural network. *Biomedical optics express*, 8(2):679–694. (page 54)

[16] Chen, X., Ouyang, L., Yan, H., Jia, X., Li, B., Lyu, Q., Zhang, Y., and Wang, J. (2017c). Optimization of the geometry and speed of a moving blocker system for cone-beam computed tomography scatter correction. *Medical physics*, 44(9). (page 56)

[17] Chen, Y. and Pock, T. (2017). Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1256–1272. (page 59)

[18] Chen, Z., Jin, X., Li, L., and Wang, G. (2013). A limited-angle CT reconstruction method based on anisotropic TV minimization. *Physics in Medicine & Biology*, 58(7):2119. (page 60)

[19] Cireşan, D., Meier, U., and Schmidhuber, J. (2012). Multi-column deep neural networks for image classification. *arXiv preprint arXiv:1202.2745.* (page 47)

[20] Donoho, D. L. (2006). Compressed sensing. *IEEE Transactions on information theory*, 52(4):1289–1306. (page 9)

[21] Eldar, Y. C. and Kutyniok, G. (2012). *Compressed sensing: Theory and applications.* Cambridge University Press. (page 9)

[22] Elsayed, O., Mahar, K., Kholief, M., and Khater, H. A. (2015). Automatic detection of the pulmonary nodules from CT images. In *SAI Intelligent Systems Conference (IntelliSys), 2015*, pages 742–746. (page 17)

[23] Flower, M. A. (2012). *Webb's physics of medical imaging.* CRC Press. (page 21)

[24] Goldman, L. W. (2007). Principles of CT: Radiation dose and image quality. *Journal of nuclear medicine technology*, 35(4):213–225. (page 7, 25)

[25] Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. (2016). *Deep learning*, volume 1. MIT press Cambridge. (page 31, 32, 33, 34, 35, 36, 37, 38, 41, 46, 47)

[26] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680. (page 2, 33, 40, 41, 42, 43, 44, 45, 60)

[27] Grossberg, S. (1988). Nonlinear neural networks: Principles, mechanisms, and architectures. *Neural networks*, 1(1):17–61. (page 31)

[28] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. (2017). Improved training of Wasserstein GANs. In *Advances in Neural Information Processing Systems*, pages 5769–5779. (page 40, 44, 45, 73, 76, 77, 101)

[29] Gupta, H., Jin, K. H., Nguyen, H. Q., McCann, M. T., and Unser, M. (2018). Cnn-based projected gradient descent for consistent CT image reconstruction. *IEEE transactions on medical imaging*, 37(6):1440–1453. (page 58)

[30] Hammernik, K., Klatzer, T., Kobler, E., Recht, M. P., Sodickson, D. K., Pock, T., and Knoll, F. (2018). Learning a variational network for reconstruction of accelerated MRI data. *Magnetic resonance in medicine*, 79(6):3055–3071. (page 60)

[31] Hammernik, K., Würfl, T., Pock, T., and Maier, A. (2017). A deep learning architecture for limited-angle computed tomography reconstruction. In *Bildverarbeitung für die Medizin 2017*, pages 92–97. (page 60)

[32] Han, Y., Yoo, J., Kim, H. H., Shin, H. J., Sung, K., and Ye, J. C. (2018). Deep learning with domain adaptation for accelerated projection-reconstruction MR. *Magnetic resonance in medicine*, 80(3):1189–1205. (page 61)

[33] He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034. (page 29, 37, 75, 76)

[34] Hounsfield, G. N. (1973). Computerized transverse axial scanning (tomography): Part 1. Description of system. *The British journal of radiology*, 46(552):1016–1022. (page 5)

[35] Hsieh, J. et al. (2009). Computed tomography: Principles, design, artifacts, and recent advances. SPIE Bellingham, WA. (page 11, 12)

[36] Hu, Z., Liu, Q., Zhang, N., Zhang, Y., Peng, X., Wu, P. Z., Zheng, H., and Liang, D. (2016). Image reconstruction from few-view CT data by gradient-domain dictionary learning. *Journal of X-ray science and technology*, 24(4):627–638. (page 58)

[37] Jin, K. H., McCann, M. T., Froustey, E., and Unser, M. (2017). Deep convolutional neural network for inverse problems in imaging. *IEEE Transactions on Image Processing*, 26(9):4509–4522. (page 7, 58)

[38] Kang, E., Min, J., and Ye, J. C. (2017). A deep convolutional neural network using directional wavelets for low-dose X-ray CT reconstruction. *Medical Physics*, 44(10). (page 54)

[39] Kida, S., Nakamoto, T., Nakano, M., Nawa, K., Haga, A., Kotoku, J., Yamashita, H., and Nakagawa, K. (2018). Cone beam computed tomography image quality improvement using a deep convolutional neural network. *Cureus*, 10(4). (page 54)

[40] Kim, K., Ye, J. C., Worstell, W., Ouyang, J., Rakvongthai, Y., El Fakhri, G., and Li, Q. (2015). Sparse-view spectral CT reconstruction using spectral patch-based low-rank penalty. *IEEE transactions on medical imaging*, 34(3):748–760. (page 58)

[41] Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. *International Conference on Learning Representations 2015*, pages 1–15. (page 34, 75)

[42] Kobler, E., Klatzer, T., Hammernik, K., and Pock, T. (2017). Variational networks: Connecting variational methods and deep learning. In *German Conference on Pattern Recognition*, pages 281–293. (page 59)

[43] Kobler, E., Muckley, M., Chen, B., Knoll, F., Hammernik, K., Pock, T., Sodickson, D., and Otazo, R. (2018). Variational deep learning for low-dose computed tomography. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6687–6691. (page 59)

[44] Kotelnikov, V. A. (1933). On the transmission capacity of the 'ether' and of cables in electrical communications. In *Proceedings of the first All-Union Conference on the technological reconstruction of the communications sector and the development of low-current engineering.* (page 8)

[45] Koza, J. R., Bennett, F. H., Andre, D., and Keane, M. A. (1996). Automated design of both the topology and sizing of analog electrical circuits using genetic programming. In *Artificial Intelligence in Design'96*, pages 151–170. (page 27)

[46] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105. (page 29, 32, 47)

[47] Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al. (2017). Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4681–4690. (page 46)

[48] Lee, D., Yoo, J., Tak, S., and Ye, J. C. (2018). Deep residual learning for accelerated MRI using magnitude and phase networks. *IEEE Transactions on Biomedical Engineering*, 65(9):1985–1995. (page 61)

[49] Lee, T., Lee, C., Baek, J., and Cho, S. (2016). Moving beam-blocker-based low-dose cone-beam CT. *IEEE Transactions on Nuclear Science*, 63(5):2540–2549. (page 56)

[50] Li, Y., Chen, Y., Hu, Y., Oukili, A., Luo, L., Chen, W., and Toumoulin, C. (2012). Strategy of computed tomography sinogram inpainting based on sinusoid-like curve decomposition and eigenvector-guided interpolation. *JOSA A*, 29(1):153–163. (page 57)

[51] Liang, K., Yang, H., and Xing, Y. (2018). Comparision of projection domain, image domain, and comprehensive deep learning for sparse-view X-ray CT image reconstruction. *arXiv preprint arXiv:1804.04289*. (page 50, 51, 52)

[52] Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., van der Laak, J. A., van Ginneken, B., and Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88. (page 29)

[53] Loubele, M., Bogaerts, R., Van Dijck, E., Pauwels, R., Vanheusden, S., Suetens, P., Marchal, G., Sanderink, G., and Jacobs, R. (2009). Comparison between effective radiation dose of CBCT and MSCT scanners for dentomaxillofacial applications. *European journal of radiology*, 71(3):461–468. (page 12)

[54] Mansfield, P. (1977). Multi-planar image formation using NMR spin echoes. *Journal of Physics C: Solid State Physics*, 10(3):L55. (page 5)

[55] Mardani, M., Gong, E., Cheng, J. Y., Vasanawala, S., Zaharchuk, G., Alley, M., Thakur, N., Han, S., Dally, W., Pauly, J. M., et al. (2017). Deep generative adversarial networks for compressed sensing automates MRI. *arXiv preprint arXiv:1706.00051*. (page 61)

[56] Mazurowski, M. A., Buda, M., Saha, A., and Bashir, M. R. (2018). Deep learning in radiology: An overview of the concepts and a survey of the state of the art with focus on MRI. *Journal of Magnetic Resonance Imaging*. (page 29)

[57] McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133. (page 29)

[58] Minsky, M. and Papert, S. A. (2017). *Perceptrons: An introduction to computational geometry*. MIT press. (page 31)

[59] Mojica, E., Pertuz, S., and Arguello, H. (2017). High-resolution coded-aperture design for compressive X-ray tomography using low resolution detectors. *Optics Communications*, 404:103–109. (page 56)

[60] Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted Boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814. (page 37, 75)

[61] Nash Jr, J. F. (1950). The bargaining problem. *Econometrica: Journal of the Econometric Society*, pages 155–162. (page 41)

[62] Ng, A. Y. and Jordan, M. I. (2002). On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes. In *Advances in neural information processing systems*, pages 841–848. (page 28)

[63] Niu, S., Gao, Y., Bian, Z., Huang, J., Chen, W., Yu, G., Liang, Z., and Ma, J. (2014). Sparse-view X-ray CT reconstruction via total generalized variation regularization. *Physics in Medicine & Biology*, 59(12):2997. (page 59)

[64] Nyquist, H. (1928). Certain topics in telegraph transmission theory. *Transactions of the American Institute of Electrical Engineers*, 47(2):617–644. (page 8)

[65] Ouyang, L., Lee, H. P., and Wang, J. (2015). A moving blocker-based strategy for simultaneous megavoltage and kilovoltage scatter correction in cone-beam computed tomography image acquired during volumetric modulated ARC therapy. *Radiotherapy and Oncology*, 115(3):425–430. (page 56)

[66] Pauwels, R., Jacobs, R., Bogaerts, R., Bosmans, H., and Panmekiate, S. (2016). Reduction of scatter-induced image noise in cone beam computed tomography: Effect of field of view size and position. *Oral surgery, oral medicine, oral pathology and oral radiology*, 121(2):188–195. (page 56)

[67] Pelt, D. M. and Batenburg, K. J. (2013). Fast tomographic reconstruction from limited data using artificial neural networks. *IEEE Transactions on Image Processing*, 22(12):5238–5251. (page 59)

[68] Priemer, R. (1990). *Introductory signal processing*, volume 6. World Scientific Publishing Company. (page 8)

[69] Prince, J. L. and Links, J. M. (2006). *Medical imaging signals and systems*. (page 13, 14, 15, 17, 18, 19, 22, 23, 24)

[70] Radon, J. (1986). On the determination of functions from their integral values along certain manifolds. *IEEE transactions on medical imaging*, 5(4):170–176. (page 18)

[71] Robb, R. A. (1982). The dynamic spatial reconstructor: An X-ray video-fluoroscopic CT scanner for dynamic volume imaging of moving organs. *IEEE transactions on medical imaging*, 1(1):22–33. (page 12)

[72] Ronneberger, O., Fischer, P., and Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer Assisted Intervention*, pages 234–241. (page 39, 40, 54, 58, 61, 71, 75)

[73] Röntgen, W. C. (1896). On a new kind of rays. *Science*, 3(59):227–231. (page 5)

[74] Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386. (page 31)

[75] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. (2016). Improved techniques for training GANs. In *Advances in Neural Information Processing Systems*, pages 2234–2242. (page 42)

[76] Scarfe, W. C. and Farman, A. G. (2008). What is cone-beam CT and how does it work? *Dental Clinics of North America*, 52(4):707–730. (page 11)

[77] Scarfe, W. C., Farman, A. G., Sukovic, P., et al. (2006). Clinical applications of cone-beam computed tomography in dental practice. *Journal-Canadian Dental Association*, 72(1):75. (page 12)

[78] Schlemper, J., Caballero, J., Hajnal, J. V., Price, A. N., and Rueckert, D. (2018). A deep cascade of convolutional neural networks for dynamic MR image reconstruction. *IEEE transactions on Medical Imaging*, 37(2):491–503. (page 60)

[79] Schulze, D., Heiland, M., Thurmann, H., and Adam, G. (2004). Radiation exposure during midfacial imaging using 4-and 16-slice computed tomography, cone beam computed tomography systems and conventional radiography. *Dentomaxillofacial Radiology*, 33(2):83–86. (page 12)

[80] Schulze, R., Heil, U., Groß, D., Bruellmann, D., Dranischnikow, E., Schwanecke, U., and Schoemer, E. (2011). Artefacts in CBCT: A review. *Dentomaxillofacial Radiology*, 40(5):265–273. (page 12)

[81] Seitzer, M., Yang, G., Schlemper, J., Oktay, O., Würfl, T., Christlein, V., Wong, T., Mohiaddin, R., Firmin, D., Keegan, J., et al. (2018). Adversarial and perceptual refinement for compressed sensing MRI reconstruction. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 232–240. (page 61)

[82] Shannon, C. E. (1949). Communication in the presence of noise. *Proceedings of the IRE*, 37(1):10–21. (page 8)

[83] Shepp, L. A. and Logan, B. F. (1974). The Fourier reconstruction of a head section. *IEEE Transactions on nuclear science*, 21(3):21–43. (page 18)

[84] Simard, P. Y., Steinkraus, D., Platt, J. C., et al. (2003). Best practices for convolutional neural networks applied to visual document analysis. In *Icdar*, volume 3. (page 47)

[85] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*. (page 54, 61)

[86] Smith-Bindman, R., Lipson, J., Marcus, R., Kim, K.-P., Mahesh, M., Gould, R., De González, A. B., and Miglioretti, D. L. (2009). Radiation dose associated with common computed tomography examinations and the associated lifetime attributable risk of cancer. *Archives of internal medicine*, 169(22):2078–2086. (page 6, 24)

[87] Suetens, P. (2002). *Fundamentals of medical imaging*. Cambridge university press. (page 13, 15, 17, 19, 20, 21, 22, 23, 24)

[88] Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112. (page 29)

[89] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9. (page 58)

[90] Tan, L. and Jiang, J. (2018). *Digital signal processing: Fundamentals and applications*. Academic Press. (page 8)

[91] Tarantola, A. (2005). *Inverse problem theory and methods for model parameter estimation*, volume 89. (page 7)

[92] Tian, Z., Jia, X., Yuan, K., Pan, T., and Jiang, S. B. (2011). Low-dose CT reconstruction via edge-preserving total variation regularization. *Physics in Medicine & Biology*, 56(18):5949. (page 58)

[93] Villani, C. (2008). *Optimal transport: Old and new*, volume 338. Springer Science & Business Media. (page 43)

[94] Whittaker, E. T. (1915). On the functions which are represented by the expansions of the interpolation-theory. *Proceedings of the Royal Society of Edinburgh*, 35:181–194. (page 8)

[95] Wolterink, J. M., Leiner, T., Viergever, M. A., and Išgum, I. (2017). Generative adversarial networks for noise reduction in low-dose CT. *IEEE transactions on medical imaging*, 36(12):2536–2545. (page 55)

[96] Wu, D., Kim, K., El Fakhri, G., and Li, Q. (2017). Iterative low-dose CT reconstruction with priors trained by artificial neural network. *IEEE transactions on medical imaging*, 36(12):2479–2486. (page 54)

[97] Würfl, T., Hoffmann, M., Christlein, V., Breininger, K., Huang, Y., Unberath, M., and Maier, A. K. (2018). Deep learning computed tomography: Learning projection-domain weights from image domain in limited angle problems. *IEEE transactions on medical imaging*, 37(6):1454–1463. (page 60)

[98] Xie, S., Zheng, X., Chen, Y., Xie, L., Liu, J., Zhang, Y., Yan, J., Zhu, H., and Hu, Y. (2018). Artifact removal using improved GoogLeNet for sparse-view CT reconstruction. *Scientific reports*, 8. (page 58)

[99] Yang, G., Yu, S., Dong, H., Slabaugh, G., Dragotti, P. L., Ye, X., Liu, F., Arridge, S., Keegan, J., Guo, Y., et al. (2018a). DAGAN: Deep de-aliasing generative adversarial networks for fast compressed sensing MRI reconstruction. *IEEE Transactions on Medical Imaging*, 37(6):1310–1321. (page 61)

[100] Yang, Q., Yan, P., Zhang, Y., Yu, H., Shi, Y., Mou, X., Kalra, M. K., Zhang, Y., Sun, L., and Wang, G. (2018b). Low-dose CT image denoising using a generative adversarial network with Wasserstein distance and perceptual loss. *IEEE Transactions on Medical Imaging*, 37(6):1348–1357. (page 46, 54)

[101] Yang, X., De Andrade, V., Scullin, W., Dyer, E. L., Kasthuri, N., De Carlo, F., and Gürsoy, D. (2018c). Low-dose X-ray tomography through a deep convolutional neural network. *Scientific reports*, 8(1):2575. (page 54)

[102] Yu, H., Liu, D., Shi, H., Yu, H., Wang, Z., Wang, X., Cross, B., Bramler, M., and Huang, T. S. (2017). Computed tomography super-resolution using convolutional neural networks. In *Image Processing (ICIP), 2017 IEEE International Conference on*, pages 3944–3948. (page 54)

[103] Zhang, X., Zhao, J., and LeCun, Y. (2015). Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657. (page 29)

[104] Zhao, C., Chen, X., Ouyang, L., Wang, J., and Jin, M. (2017). Robust moving-blocker scatter correction for cone-beam computed tomography using multiple-view information. *PloS one*, 12(12):e0189620. (page 56)

[105] Zhao, C., Zhong, Y., Duan, X., Zhang, Y., Huang, X., Wang, J., and Jin, M. (2018). 4D cone-beam computed tomography (CBCT) using a moving blocker for simultaneous radiation dose reduction and scatter correction. *Physics in Medicine & Biology*, 63(11):115007. (page 56)

[106] Zhao, J., Chen, Z., Zhang, L., and Jin, X. (2016). Few-view CT reconstruction method based on deep learning. In *Nuclear Science Symposium, Medical Imaging Conference and Room-Temperature Semiconductor Detector Workshop (NSS/MIC/RTSD), 2016*, pages 1–4. (page 58)

[107] Zhou, Y.-T. and Chellappa, R. (1988). Computation of optical flow using a neural network. In *IEEE International Conference on Neural Networks*, volume 1998, pages 71–78. (page 36)