Aus dem Lehrstuhl für Computerunterstützte Klinische Medizin

der Medizinischen Fakultät Mannheim

(Direktor: Prof. Dr. rer. nat. Lothar R. Schad)

Optimized Training Pipeline for Deep Learning Applications in Medical Image Processing

Inauguraldissertation
zur Erlangung des Doctor scientiarum humanarum (Dr. sc. hum.)
der
Medizinischen Fakultät Mannheim
der Ruprecht-Karls-Universität zu Heidelberg

vorgelegt von

Alena-Kathrin Golla geb. Schnurr, M.Sc.

aus

Hamburg

2021

Dekan: Prof. Dr. med. Sergij Goerdt
Referent: Prof. Dr. Ing. Frank G. Zöllner

Optimized Training Pipeline for Deep Learning Applications in Medical Image Processing

Deep learning has revolutionized the field of digital image processing. However, training a Convolutional Neural Network (CNNs) requires a complex pipeline consisting of image normalization, data augmentation, sample mining, parameter updates, performance evaluation and monitoring. Regardless of the image processing task, development of new approaches requires this pipeline to work before any experiments can be performed. For tomographic image data, special care is necessary with regard to the modality-specific image properties.

The work presented in this thesis provides a training pipeline based on the commonly used TensorFlow library. The pipeline is tailored to three medical image processing tasks: image regression, semantic segmentation and image classification. It was utilized to train CNNs in four studies on medical deep learning.

In an initial study the pipeline was used to train CNNs for limited angle artifact correction in circular tomosynthesis. The CNNs were trained on simulated data and were subsequently able to correct artifacts in synthetic and real scans. On the real data an artifact reduction of 30 to 40% was achieved using a 3D ResNet.

In a second study intra-individual volume change analysis in serial T_1-weighted magnetic resonance imaging scans of the brain was realized with a 3D U-Net. The results demonstrated that the deep learning version could approximate the complex Voxel-guided Morphometry mapping at high quality (structural similarity index measure $= 0.9521 \pm 0.0236$) while reducing the computation time by 99.62%.

In a third study, the pipeline was applied to vessel segmentation in contrast enhanced computed tomography (CT). Ratio-based sampling was proposed to counter the class-ratio imbalance. Using the pipeline, 2D and 3D versions of the U-Net, the V-Net and the DeepVesselNet were trained. Well performing networks were combined into an ensemble. The method achieved Dice similarity coefficients of 0.758 ± 0.050 (veins) and 0.838 ± 0.074 (arteries) on the IRCAD data set. Application to the BTCV data set showed a high transfer ability.

In the final study, the pipeline was used to train several CNNs to classify whether CT images show an abdominal aortic aneurysm. Across the whole data set the algorithm achieved an accuracy of 0.856 and area under the receiver operating characteristic curve of 0.926. Using layer-wise relevance propagation, relevance maps were generated that offer interpretable visualization of the CNN's decision process.

The presented framework enables fast prototyping of deep learning applications for medical image processing. Due to the modular design individual components can be switched easily. It is a valuable tool in the development of clinically relevant artificial intelligence algorithms.

Optimierte Trainingspipeline für Deep-Learning-Anwendungen in der medizinischen Bildverarbeitung

Deep Learning hat den Bereich der digitalen Bildverarbeitung revolutioniert. Jedoch erfordert das Training von Convolutional Neural Networks (CNNs) eine komplexe Pipeline aus Normalisierung, Datenaugmentierung, Sample Auswahl, Parameter-Updates, Evaluierung und Überwachung. Die Entwicklung neuer Ansätze erfordert zunächst das Funktionieren dieser Pipeline, unabhängig davon, welche Aufgabe gelöst werden soll. Bei tomographischen Bilddaten ist besondere Sorgfalt im Hinblick auf die modalitätsspezifischen Bildeigenschaften geboten.

In dieser Arbeit wird eine Trainingspipeline vorgestellt, die auf der weit verbreiteten TensorFlow-Bibliothek basiert. Die Pipeline ist auf drei Aufgaben der medizinisch Bildverarbeitung zugeschnitten: Bildregression, semantische Segmentierung und Bildklassifikation. Sie wurde zum Trainieren von CNNs in vier Studien zu medizinischem Deep Learning eingesetzt.

In einer ersten Studie wurde die Pipeline verwendet, um CNNs für die Korrektur von Unterabtastungsartefakten in zirkulärer Tomosynthese zu trainieren. Die CNNs wurden auf simulierten Daten trainiert und waren anschließend in der Lage, Artefakte in synthetischen und echten Scans zu korrigieren. Auf den realen Daten wurde mit einem 3D-ResNet eine Artefaktreduktion von 30 bis 40% erreicht.

In einer zweiten Studie wurde eine intra-individuelle Volumenänderungsanalyse in seriellen T_1-gewichteten Magnetresonanztomographieaufnahmen des Gehirns mit einem 3D-U-Net realisiert. Die Ergebnisse zeigten, dass die Deep-Learning-Version den komplexen Voxel-guided Morphometry Algrithmus mit hoher Qualität (structural similarity index measure $= 0.9521 \pm 0.0236$) approximieren konnte, während die Berechnungszeit um 99.62% reduziert wurde.

In einer dritten Studie wurde die Pipeline zur Gefäßsegmentierung in der kontrastverstärkten Computertomographie (CT) eingesetzt. Es wurde ein verhältnisbasiertes Sampling vorgeschlagen, um dem Ungleichgewicht zwischen den Klassen entgegenzuwirken. Mit der Pipeline wurden 2D- und 3D-Versionen des U-Netzes, des V-Netzes und des DeepVesselNet trainiert. Gut funktionierende Netze wurden zu einem Ensemble kombiniert. Die Methode erreichte Dice-Koeffizienten von 0.758 ± 0.050 (Venen) und 0.838 ± 0.074 (Arterien) auf dem IRCAD-Datensatz. Die Anwendung auf den BTCV-Datensatz zeigte eine hohe Übertragungsfähigkeit.

In der letzten Studie wurde die Pipeline verwendet, um mehrere CNNs zur Klassifikation von abdominalen Aortenaneurysmen in CT-Bildern zu trainieren. Auf dem gesamten Datensatz erreichte der Algorithmus eine Genauigkeit von 0.856 und eine Fläche unter der Receiver-Operating-Characteristic-Kurve von 0.926. Mittels Layer-wise Relevance Propagation wurden Relevanzkarten erzeugt, die eine interpretierbare Visualisierung des Entscheidungsprozesses des CNNs bieten.

Das vorgestellte Framework ermöglicht das schnelle Prototyping von Deep-Learning-Anwendungen für die medizinische Bildverarbeitung. Durch den modularen Aufbau können einzelne Komponenten leicht ausgetauscht werden. Es ist ein wertvolles Werkzeug bei der Entwicklung von klinisch relevanten Deep-Learning-Algorithmen.

Contents

List of Figures **xiii**

List of Tables **xv**

List of Acronyms **xvii**

1 Introduction and Outline **1**
- 1.1 Motivation . . . 1
- 1.2 Outline . . . 2
- 1.3 Citation of Previous Publications . . . 3

2 Background **5**
- 2.1 Medical Imaging . . . 5
  - 2.1.1 Computed Tomography . . . 5
  - 2.1.2 Magnetic Resonance Imaging . . . 7
- 2.2 Image Processing . . . 9
  - 2.2.1 Classification . . . 9
  - 2.2.2 Segmentation . . . 10
  - 2.2.3 Regression . . . 10
- 2.3 Deep Learning . . . 11
  - 2.3.1 Artificial Neural Networks . . . 12
  - 2.3.2 Convolutional Neural Networks . . . 13
  - 2.3.3 Training . . . 16

3 Training Pipeline Implementation **23**
- 3.1 Basis Module . . . 23
- 3.2 Specialized Modules . . . 24

4 Simulation-Based Deep Artifact Correction **27**
- 4.1 Introduction . . . 27
- 4.2 Material & Methods . . . 29
  - 4.2.1 Interventional Tomosynthesis . . . 29
  - 4.2.2 Real Data . . . 29
  - 4.2.3 Simulation Data . . . 30
  - 4.2.4 Networks . . . 31
  - 4.2.5 Training . . . 33
  - 4.2.6 Experiments . . . 34
- 4.3 Results . . . 34

4.3.1 Anthropomorphic Simulations 34
4.3.2 Catheter Simulations 35
4.3.3 Simplex Simulations 35
4.3.4 Proof of Principle on Real Data 35
4.4 Discussion 38
4.5 Conclusion 42

5 Deep VGM 45
5.1 Introduction 45
5.2 Materials and Methods 45
5.2.1 Image Data 45
5.2.2 Voxel-Guided Morphometry 46
5.2.3 Preprocessing 46
5.2.4 Deep VGM: Architecture 46
5.2.5 Deep VGM: Training and Loss Functions 47
5.2.6 Evaluation 49
5.2.7 Implementation 49
5.3 Results 49
5.3.1 Quantitative Evaluation 49
5.3.2 Qualitative Evaluation 49
5.4 Discussion 50
5.5 Conclusion 51

6 Convolutional Neural Network Ensemble Segmentation 53
6.1 Introduction 53
6.1.1 Related Work 53
6.1.2 Contribution 54
6.2 Materials and Methods 55
6.2.1 Data 55
6.2.2 Preprocessing and Sample Mining 56
6.2.3 Networks 57
6.2.4 Training and Loss Function 58
6.2.5 Prediction and Ensembles 59
6.2.6 Evaluation Metrics 59
6.3 Experiments and Results 60
6.3.1 Experiment 1 60
6.3.2 Experiment 2 61
6.3.3 Experiment 3 65
6.4 Discussion 65
6.5 Conclusion 68

7 Abdominal Aortic Aneurysm Screening 71
7.1 Introduction 71
7.1.1 Related Work 71
7.1.2 Contribution 72
7.2 Materials and Methods 72
7.2.1 Patients and Data Set 72
7.2.2 Networks, Preprocessing and Training 72

7.2.3 Layer-wise Relevance Propagation 74
7.2.4 Evaluation 74
7.3 Experiments and Results 75
7.4 Discussion 78

8 Summary 81

9 Outlook 85

10 Bibliography 87

11 Appendix 103
11.1 Supplementary Material Chapter 6 103
11.2 Supplementary Material Chapter 7 104

12 Publications 107

13 Curriculum Vitae 111

14 Acknowledgements 113

List of Figures

2.1 MRI and CT images 5
2.2 CT detectors 7
2.3 HU scale 7
2.4 Classification 10
2.5 Segmentation 11
2.6 Regression 11
2.7 Deep learning categorization 12
2.8 Perceptron 13
2.9 CNN architecture 13
2.10 CNN layers 14
2.11 CNN training pipeline 16

4.1 Artifact correction: circular tomosynthesis 29
4.2 Artifact correction: networks architectures 32
4.3 Artifact correction: simulation results 36
4.4 Artifact correction: comparison simulated images 37
4.5 Artifact correction: real data results 39
4.6 Artifact correction: comparison porcine phantom 40
4.7 Artifact correction: HU-artifacts 41

5.1 Deep VGM: example data 47
5.2 Deep VGM: architecture 48
5.3 Deep VGM: results and difference maps 50
5.4 Deep VGM: lesion examination 51

6.1 Vessel segmentation: network architectures 58

6.2 Vessel segmentation: experiment 1 - visual comparison 61
6.3 Vessel segmentation: experiment 2 - visual comparison 63
6.4 Vessel segmentation: experiment 2 - comparison challenging cases . . 64
6.5 Vessel segmentation: experiment 3 - visual comparison 67

7.1 Abdominal aortic aneurysm screening: example cases 74
7.2 Abdominal aortic aneurysm screening: network architectures 75
7.3 Abdominal aortic aneurysm screening: experiment 1 - roc curves . . 76
7.4 Abdominal aortic aneurysm screening: relevance analysis 77

11.1 Vessel segmentation: training and validation accuracy 104
11.2 Abdominal aortic aneurysm screening: training and validation accuracy 105

List of Tables

2.1 Loss functions 19

4.1 Artifact correction: acquistion parameters 30

5.1 Deep VGM: quantitative evaluation 50

6.1 Vessel segmentation: related work 55
6.2 Vessel segmentation: data set statistics 56
6.3 Vessel segmentation: number of trainable parameters 57
6.4 Vessel segmentation: sampling parameters 59
6.5 Vessel segmentation: experiment 1 - quantitative comparison 62
6.6 Vessel segmentation: experiment 2 - ensemble weights 63
6.7 Vessel segmentation: experiment 2 - quantitative comparison 65
6.8 Vessel segmentation: experiment 3 - quantitative comparison 66

7.1 Abdominal aortic aneurysm screening: data set statistics 73
7.2 Abdominal aortic aneurysm screening: experiment 1 - quantitative comparison 76
7.3 Abdominal aortic aneurysm screening: experiment 3 - quantitative comparison 78

11.1 Vessel segmentation: vascular structures 103

List of Acronyms

3-CCD	Three Charge Coupled Devices
AAA	Abdominal Aortic Aneurysm
AI	Artificial Intelligence
ANN	Artificial Neural Network
API	Application Programming Interface
ASSD	Average Symmetric Surface Distance
AUC	Area Under the Curve
C	Connectivity
CBCT	Cone-Beam Computed Tomography
CE-CT	Contrast Enhanced Computed Tomography
CNN	Convolutional Neural Network
CSF	Cerebrospinal Fluid
CT	Computed Tomography
CTA	Computed Tomography Angiography
cTS	Circular Tomosynthesis
DAC	Deep Artifact Correction
DL	Deep Learning
DSC	Dice Similarity Coefficient
ECMO	Extracorporeal Membrane Oxygenation
ELU	Exponential Linear Unit
EVAR	Endovascular Aneurysm Repair
FCNN	Fully Convolutional Neural Network
FN	False Negative
FP	False Positive
GAN	Generative Adversarial Network
GPU	Graphical Processing Unit

HU	Hounsfield Unit
LCC	Largest Connected Component
LRP	Layer-wise Relevance Propagation
MAE	Mean Absolute Error
ML	Machine Learning
MLP	Multi-Layer-Perceptron
MRA	Magnetic Resonance Angiography
MRI	Magnetic Resonance Imaging
MS	Multiple Sclerosis
MSCT	Multi-slice Computed Tomography
MSE	Mean Squared Error
PACS	Picture Archiving and Communication System
PET	Positron Emission Tomography
ReLU	Rectified Linear Unit
RF	Radio Frequency
RIS	Radiology Information System
RMSE	Root Mean Squared Error
ROC	Receiver Operating Characteristic
ROI	Region of Interest
SSIM	Structural Similarity Index Measure
TACE	Transarterial Chemoembolization
TIPS	Transjugular Intrahepatic Portosystemic Shunt
TN	True Negative
TP	True Positive
TV	Total Variation
VCR	Vessel Confusion Rate
VGM	Voxel-Guided Morphometry

1. Introduction and Outline

1.1 Motivation

Deep learning has lead to unprecedented advances in the field of digital image processing since the reemergence of the technology in 2012. While initial works focused on the photo domain, the technology quickly translated to the medical image domain [1]. Medical imaging is an essential part of the clinical routine. The number of Magnetic Resonance Imaging (MRI) and Computed Tomography (CT) scans continuously increased in the last years [2]. This increase comes with a growing demand for professional assessment of the resulting image data by radiologists. A thorough assessment is a time consuming task which often still requires the manual delineation of structures for volumetry. High performance algorithms for detection and delineation offer considerable automation of the assessment and reduce the required time while simultaneously supporting the radiologist with objective measurements.

Accelerating patient care is a focus point of the research campus "Mannheim Molecular Intervention Environment" (M^2OLIE). Here, methods to improve the treatment cycle of oligometastatic liver cancer are explored. Key aspects of this cycle are the initial multimodal imaging, the subsequent minimally invasive image-guided interventions with robotic guidance and the molecular characterization of tissue samples. The multimodal imaging protocol includes CT, MRI and Positron Emission Tomography (PET), as these provide a basis for an initial diagnosis which can then be further differentiated by additional examinations such as biopsies [3]. Interventions are performed under the guidance of Cone-Beam Computed Tomography (CBCT) supported by intelligent assistance systems. This requires the prior analysis of the diagnostic multimodal data. Segmentation of the target structures as well as organs at risk are required to safely plan the puncture path in needle-based interventions.

M^2OLIE includes several sub-projects working on a broad range of scientific questions based on different imaging modalities. The barrier of integrating Deep Learning (DL) into these sub-projects can be substantially reduced if a framework with all required basic functionalities such as image normalization, data augmentation, sample mining, parameter updates, performance evaluation and monitoring is established. Such a framework enables fast prototyping and allows researchers to focus on the scientific question. Furthermore, medical DL applications need to be tailored to the distinct requirements of the domain. Tomographic medical images, for instance, have a clearly defined extent for each voxel. This real world spacing information has to be accounted for, when preparing the data. Additionally, different imaging modalities require different approaches for normalizing value ranges depending on the physical effect they are based on.

The access to medical image data is tightly regulated due to patient data protection laws [4]. Disease incidence varies strongly and for some rarely occurring diseases only a few cases are imaged each year. Furthermore, many imaging studies initially only examine a small cohort of patients. These causes limit the availability of large data sets, which are required for DL. Due to the limitation of data in the medical domain, algorithmic approaches to facilitate training deep learning solutions even on limited data sets are required. To achieve this, data preprocessing as well as the training step itself have to be extended.

In this thesis, I present a training pipeline that is tailored to the processing of medical image data of different modalities and can be applied to a variety of tasks. The pipeline is optimized to enable training even on small data sets. Due to its modular design, individual components such as the network architecture can easily be switched.

1.2 Outline

This thesis is written cumulatively. The training pipeline has been applied to four medical image processing tasks. Chapter 4 to Chapter 7 each present a self-contained scientific study. Therefore each of these chapters is comprised of an introduction, the description of the materials and methods, the presentation of the results as well as a discussion and a conclusion. Furthermore, a statement of contribution is given at the end of each of these chapters.

Chapter 2 aims to provide a short introduction to medical imaging with focus on CT and MRI. An introduction to image processing is given as well as a description of the fundamentals of DL.

Chapter 3 provides a short description of the pipeline implementation and available options.

In Chapter 4, the application of the pipeline to image regression is presented. In this work different Convolutional Neural Networks (CNNs) were trained to correct limited angle artifacts in Circular Tomosynthesis (cTS) images. Simulated data was used for the training and it was shown that the networks were able to correct real image data subsequently.

In Chapter 5, a second application of the pipeline to image regression is presented. In this study the pipeline was used to train a 3D CNN to predict quantified change maps between two T1 MRI images based on Voxel-Guided Morphometry (VGM).

Chapter 6 presents the application of the pipeline to semantic segmentation in CT data. The aim was the segmentation of the arterial and venous vessel system in the abdomen. Several state of the art architectures using 2D and 3D operations were compared. The improved accuracy of multi-architecture ensembles has also been shown.

Chapter 7 presents the use of the pipeline to image classification. 3D extensions of literature networks were used to classify CT images to show or not to show an Abdominal Aortic Aneurysm (AAA). The decision process of the best performing network was analyzed using Layer-wise Relevance Propagation (LRP).

Chapter 8 gives an overview of the entire thesis as well as a detailed overview of the results of the scientific studies presented in Chapter 4 to Chapter 7.

In Chapter 9 future research directions as well as the relevance of the presented work are discussed.

1.3 Citation of Previous Publications

Several chapters of this thesis have already been published or are currently submitted for publication. The citations for these chapters are:

Chapter 4: A.-K. Schnurr, K. Chung, T. Russ, L. R. Schad and F. G. Zöllner. Simulation-Based Deep Artifact Correction with Convolutional Neural Networks for Limited Angle Artifacts. Zeitschrift für Medizinische Physik, 29 (2), p.150-161, doi: 10.1016/j.zemedi.2019.01.002, 2019.

Chapter 5: A.-K. Schnurr, P. Eisele, C. Rossmanith, S. Hoffmann, J. Gregori, A. Dabringhaus, M. Kraemer, R. Kern, A. Gass and F. G. Zöllner. Deep Voxel-Guided Morphometry (VGM): Learning regional brain changes in serial MRI. Proc. International Workshop on Machine Learning in Clinical Neuroimaging, Held in Conjunction with MICCAI, Lima, Peru, LNCS 12449, p.159-168, doi: 10.1007/978-3-030-66843-3_16, 2020 © Springer Nature Switzerland AG 2020.

Chapter 6: A.-K. Golla, D. F. Bauer, R. Schmidt, T. Russ, D. Nörenberg, K. Chung, C. Tönnes, L. R. Schad, and F. G. Zöllner. Convolutional Neural Network Ensemble Segmentation with Ratio-based Sampling for the Arteries and Veins in Abdominal CT Scans. IEEE Transactions on Biomedical Engineering, in press, doi: 10.1109/TBME.2020.3042640, 2021 © 2020 IEEE.

Chapter 7: A.-K. Golla, C. Tönnes, T. Russ, D. F. Bauer, M. F. Frölich, S. J. Diehl, S. O. Schönberg, M. Keese, L. R. Schad, F. G. Zöllner and J. S. Rink. Automated Screening for Abdominal Aortic Aneurysm in CT scans under clinical conditions using Deep Learning. under review at European Radiology, submitted 23.03.2021.

2. Background

2.1 Medical Imaging

Medical imaging includes a broad spectrum of methods, which use physical mechanisms to portray a patient's individual anatomy. The resulting images can be used for diagnosis and therapy, but also to monitor diseases over time. As the different modalities yield complementary information about the imaged tissue, multimodal imaging, meaning the use of different imaging methods is often applied. In the following section CT and MRI will be presented as they are the relevant modalities for this work.

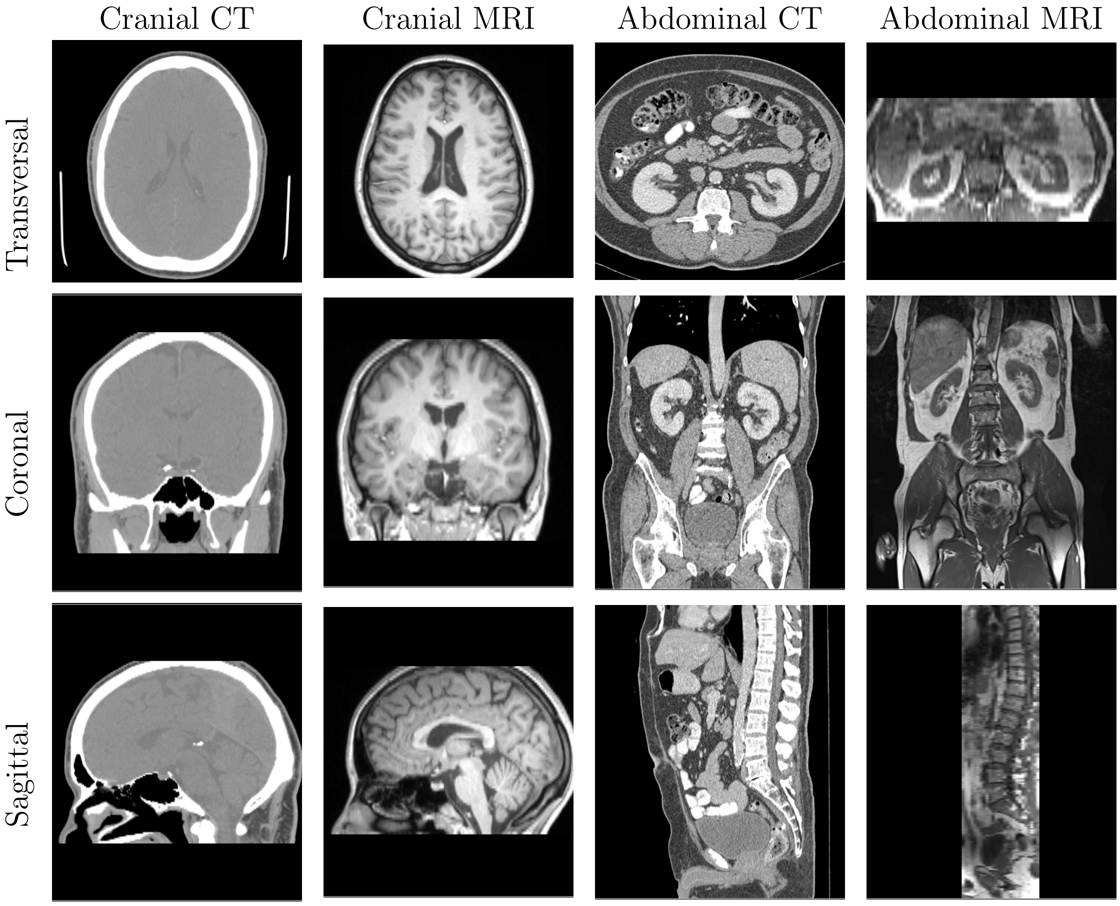


Figure 2.1: Example of MRI and CT images. The two left columns show cranial images, while the two right columns show abdominal images. For each image volume one exemplary slice of each principal anatomical plane is shown.

2.1.1 Computed Tomography

CT utilizes the attenuation of electromagnetic waves (X-rays) to acquire images of the internal human anatomy. CT is a tomographic extension of the principle used for conventional X-ray imaging, which does not suffer from superposition of distinct anatomical structures as it reconstructs volumetric 3D data.

A CT imaging system consists of an X-ray source and a detector, which are mounted onto a gantry that can be rotated around a movable patient table. In the source X-rays are generated when fast electrons enter a solid metal anode and are decelerated in the process. The wavelength of the resulting photons and thus also the radiation energy depends on the velocity of the electrons, which in turn depends on the voltage applied for acceleration. For diagnostic imaging, acceleration voltages between 25kV and 150kV are used [5]. The overall intensity of the generated X-ray spectrum, however, is dictated by the anode current.

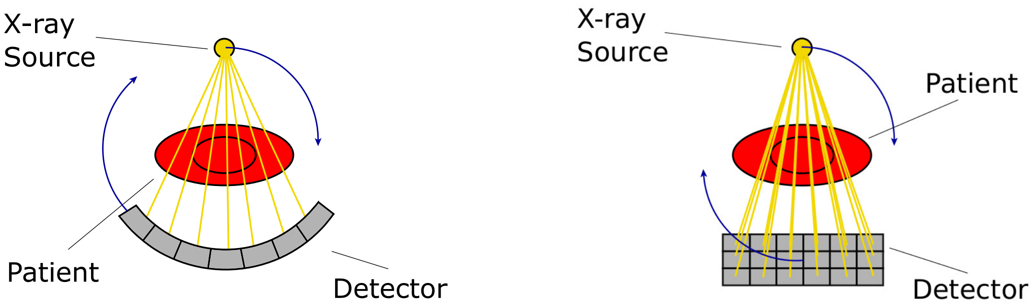
When X-rays pass through a patient, they are attenuated resulting in a decrease in radiation intensity. The attenuation is exponential along the incident direction. Several physical interactions cause this decrease in radiation intensity: Rayleigh scattering, Compton scattering and photoelectric absorption. The influence of each of these processes governing the attenuation is dependent on the penetrated material and the X-ray wavelength. In the diagnostic energy window photoelectric absorption and Compton scattering are responsible for the majority of the absorbed energy, while the effect of Rayleigh scattering is negligible.

After passing through the patient, the X-rays reach the detector. As some X-rays are deflected due to scattering, a collimator is placed in front of the detector. This grid filters the photons according to their directions and stops all that are not coming directly from the the source. Each scattered photon would otherwise contribute to the noise in the resulting image. The detector itself consists of an array of detector elements. Each element is made of a scintillator medium, which converts the high energy X-rays into lower energy photons in the visible spectrum and a photon detector, which converts the light into an electrical signal.

The exact arrangement of detector elements depends on the type of CT scanner. Modern Multi-slice Computed Tomographys (MSCTs) use several rows of detector elements, which are arranged curve, in combination with a fan-shaped X-ray beam. The scanning process is repeated for multiple rotation angles around the patient table resulting in a series of radiographic projections. While in a single projection several structures inside the patient are mapped to the same point on the detector, using different angles enables estimation of the spatial distribution of the attenuation coefficients. To scan several slices, the patient table is moved in small increments and the scan is repeated for each position.

CBCTs on the other hand use flat-panel detectors in combination with a a cone-shaped X-ray beam. CBCT systems are more compact than MSCTs and can be mounted on C-arms instead of closed gantries. To acquire an image volume, CBCTs require less rotations around the patient as the flat-panel detectors capture a projections image and not only a few rows. This results in shorter scan times and lower radiation exposure of the patient. However, the radiodensity values measured in CBCT are inaccurate. MSCT and CBCT are visualized in Figure 2.2.

Modern CT systems implement the filtered back projection algorithm or iterative solutions to reconstruct the CT slices from the projections [6]. Each resulting voxel represents a small subvolume of the patient and its average attenuation. The size of this subvolume is defined by the slice thickness and the in-plane spatial resolution.



(a) Multi-slice CT **(b)** Cone-beam CT

Figure 2.2: Arrangement of detector elements in two different types of CT systems.

The radiodensity in CT data is given in Hounsfield Units (HUs) as introduced by Hounsfield [7]. It is a linear transformation of a voxel's average attenuation coefficient μ based on the attenuation coefficients at standard pressure and temperature of distilled water μ_{water} and μ_{air}. High values of the attenuation coefficient μ result from a high density or high atomic number of the imaged material. The HU for an image voxel is given by:

$$\text{CT-Number}(\mu) = \frac{\mu - \mu_{\text{water}}}{\mu_{\text{water}} - \mu_{\text{air}}} \cdot 1000\,\text{HU} \tag{2.1}$$

Different tissue classes fall into specific intervals on the HU scale as can be seen in Figure 2.3. Calibration of the HU scale based on water and air enables the direct comparison of CT images more or less independent of the scanner manufacturer. Two exemplary CT images are displayed in Figure 2.1. The first column shows a cranial CT, where the high density cortical bone shows a strong contrast to the low HU brain tissue. The third column shows an example for an abdominal CT scan. While the HUs vary for the internal organs, the different soft tissues show a low contrast in comparison to the high density spine and the air filled lungs.

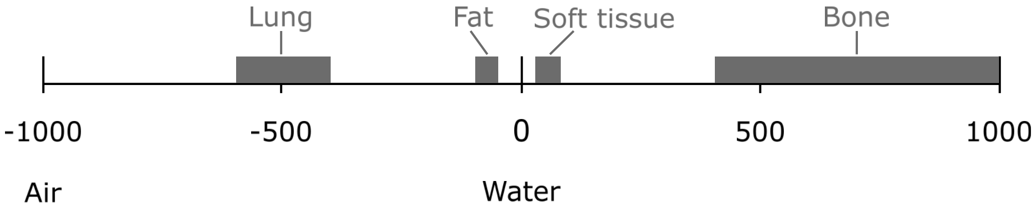


Figure 2.3: The HU scale and intervals of tissue classes on it.

2.1.2 Magnetic Resonance Imaging

MRI is based on the principle of nuclear magnetic resonance, which describes the behavior of atomic nuclei absorbing and re-emitting electromagnetic radiation when placed in a magnetic field [8].

The patient is placed in a strong magnetic field B_0, which leads to the predominant alignment of the protons' spin in the patient body along this field. The direction of

B_0 is commonly referred to as the z-axis of the MRI system. The aligned protons themselves also produce a magnetic field along the z-axis. As this magnetization is not static, but moves in phase with the precessing protons, it can induce an electrical current in coils. The frequency of the proton's precession around the z-axis is called Larmor frequency ω. It depends on the strength of the magnetic field B_0 and the gyromagnetic ratio γ.

$$\omega = -\gamma B_0 \tag{2.2}$$

Via a Radio Frequency (RF) pulse the precession axis of the protons can be changed. However, the protons can only absorb energy from a pulse with a frequency equal to ω. The angle, by which the precession axis is flipped, is determined by the duration of the RF pulse. The axis modification also causes the direction of the proton's magnetic field to change. After applying the excitation pulse, the magnetization converges back to the equilibrium state, where the spins are aligned with B_0. How fast a proton realigns, depends on how it is present in the body. Two relaxation effects are differentiated: T_1 and T_2.

When the direction of the precession axis of the protons is flipped away from the z-axis, the magnetization along this axis M_z is reduced. Over time the longitudinal magnetization increases again as the protons realign with B_0. For $t \to \infty$, M_z reaches the initial magnitude before the pulse $M_{z,0}$. This relaxation is characterized by the tissue-specific time constant T_1. The longitudinal magnetization at a specific time point after excitation is given by:

$$M_z(t) = M_{z,0} - (M_{z,0} - M_z(0))\, e^{-\frac{t}{T_1}}. \tag{2.3}$$

When the precession axis is flipped by 90°, it causes a magnetization in the x,y-plane perpendicular to B_0. This transversal magnetization $M_{x,y}$ decreases when the protons realign with the z-axis. For $t \to \infty$, it regresses back to zero. This relaxation is characterized by the tissue-specific time constant T_2. $M_{x,y}$ at a given time point after the initial magnetization $M_{x,y}(0)$ is given by:

$$M_{x,y}(t) = M_{x,y}(0)\, e^{-\frac{t}{T_2}}. \tag{2.4}$$

To be able to differentiate the location of origin for each current measured in the coils, the field strength can be altered via a linear magnetic field gradient G_x being applied continuously. This results in the Larmor frequency of each proton being determined by its position x on the x-axis (see Equation 2.5). This method is called frequency encoding. Additionally, the scan volume can be encoded via phase encoding. For this a gradient field is temporarily superimposed between the RF pulse and the signal measurement. This then causes the aligned spins to dephase as well as their measurement signal.

$$\omega = -\gamma(B_0 + G_x) \tag{2.5}$$

Depending on the strength and order of radio frequency pulses as well as additional imaging parameters, MRI can produce a variety of different contrasts. Established combinations are called sequences. In general, MRI is not a quantitative imaging method, but rather a qualitative one. This means that the resulting image intensity values are dimensionless and not comparable. Quantitative MRI sequences exist, but they are not as commonly employed. As MRI targets protons, the main source of signal in the body is water. Tissues with low water content such as bone, show low to no MRI signal. Two exemplary T_1 images are displayed in Figure 2.1. The second column shows a cranial MRI, where different brain tissues show high contrast, but the skull is only identifiable as a black ring around the brain. The fourth column shows an example for an abdominal MRI scan. The different soft tissues show a high contrast.

2.2 Image Processing

In the time of digitization the demand for automated systems to interpret large amounts of image data has increased strongly. The field of (digital) image processing uses computers in combination with specialized algorithms to analyze images. The following section details three major image processing tasks.

2.2.1 Classification

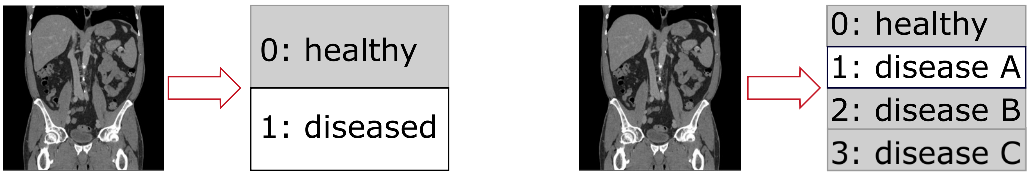
Classification is the task of assigning the correct category to an image sample based on its characteristics. The possible categories are limited to a known set of classes. Based on the number of classes and how many classes a sample can belong to, classification problems can further be categorized into

- binary classification, where the set of categories is limited to two classes and each sample only belongs to one class,
- multi-class classification, where the set of categories has more than two classes and each sample only belongs to one class
- and multi-label classification, where a sample can belong to multiple classes.

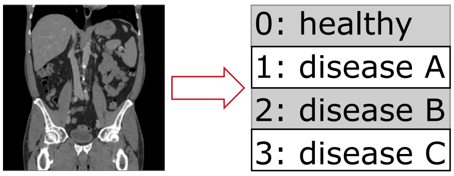
A binary classification in the medical domain would e.g. be the classification of a CT image into the two classes *healthy* or *diseased* (see Figure 2.4a). An example for multi-class classification would be the classification of the CT image into one disease out of a possible set (see Figure 2.4b). Finally a multi-label classification would be the classification of a CT into several disease categories that can occur simultaneously (see Figure 2.4c).

Classes are encoded via a class index, meaning that each class is assigned a unique integer identifier. For binary and multi-class classification the labels are encoded via *one-hot encoding*. This means the label consists of a vector with a length equal to the number of classes. All entries are zero, except for the correct class, where the entry is one. This results in a Boolean encoding which can also be interpreted as class probabilities, where the correct class has a probability of 100%. For multi-class problems several classes can be assigned the value one.

The quality assessment of classifiers is commonly based on correctly and incorrectly classified samples. The majority of metrics are defined using the binary-based categorization of True Positives (TPs), False Positives (FPs), True Negatives (TNs) and False Negatives (FNs). Additionally, the ranking performance can be evaluated based on the predicted probabilities [9].



(a) Binary **(b)** Multi-Class



(c) Multi-Label

Figure 2.4: Schematic representations of medical domain examples for the three classes of classification problems. The CT image on the left is mapped to the classes on the right. Each class has an index and a class name. Label encoding is visualized as gray boxes for zero and white boxes for one.

2.2.2 Segmentation

Segmentation is the process of partitioning a digital image into multiple segments. In the case of *semantic segmentation* the goal is to assign an object class to each pixel or voxel. Objects of the same class are not differentiated. Separating individual objects of the same class is called *instance segmentation*.

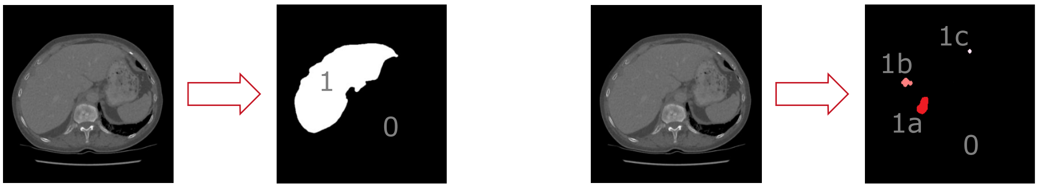
In the medical domain most problems fall into the category of semantic segmentation. This includes for example the segmentation of organs in a CT image. Here, each target organ is assigned a class index, which is then set as the voxel value in the label. For voxels not belonging to any of the target classes a background class (usually class zero) is assigned. Which object classes are defined depends on the application. For example, for some applications the differentiation of arteries and veins might be useful while for others a single vessel class is sufficient.

To determine the performance of segmentation algorithms a ground truth is required. The current gold standard to obtain such ground truth labels is via manual annotation by an experienced physician. The similarity between the generated segmentation and the ground truth is commonly determined based on overlap, volume or spatial distance [10].

As an example for semantic segmentation, liver segmentation is shown in Figure 2.5a. Instance segmentation is visualized in Figure 2.5b. Here, the goal is to differentiate between individual liver lesions.

2.2.3 Regression

While classification and segmentation predict a category, other tasks like reconstruction or denoising require the prediction of continuous values. For simplicity these tasks are only introduced here as regression tasks. This broad category includes all problems where a single continuous scalar value or a set of continuous values is to be derived from an image.



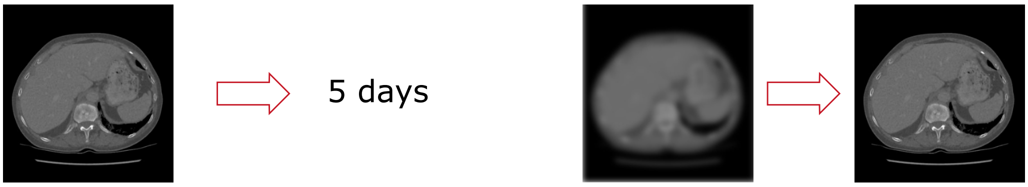
(a) Semantic segmentation

(b) Instance segmentation

Figure 2.5: Schematic representations of medical domain examples for the two disciplines of segmentation problems. On the left, the task is to assign each voxel to the background or liver class. On the right, lesions are not only supposed to be differentiated from the background, but also from other lesion instances. In this example, there are three instances (a,b,c) of the lesions class (1).

Similar to classification where the goal is to predict a single class, it is possible to predict a single parameter based on an input image. An example for this would be the prediction of patient survival in days based on a CT scan as shown in Figure 2.6a. Other applications require the prediction of a whole image. Examples for image regression are image enhancement (see Figure 2.6b), reconstruction and generation.

The evaluation criteria for regression tasks depend on the application. The absolute and mean squared error are usually applicable. In the field of image regression more sophisticated metrics for image comparison and quality can be used [11].



(a) Single value regression

(b) Image regression

Figure 2.6: Schematic representations of medical domain examples for two types of regression problems. The CT image on the left is mapped to a single scalar, which represents the estimated days of patient survival. The CT image on the right is blurry and is mapped to an enhanced version of the image.

2.3 Deep Learning

In the following section an overview of the concepts and methods of DL is given. How DL relates to other fields is visualized in Figure 2.7. DL belongs to the broad domain of Artificial Intelligence (AI). AI can be defined as the task ”to program computers to carry out tasks that would require intelligence if carried out by human beings” [12]. It is commonly further distinguished into *weak* and *strong* AI. Weak refers to ”the use of software to study or accomplish specific problem solving or reasoning tasks that do not encompass the full range of human cognitive abilities” [12]. Strong AI on the other hand aims at the synthetization of consciousness and programs which are truly able to reason and solve arbitrary problems.

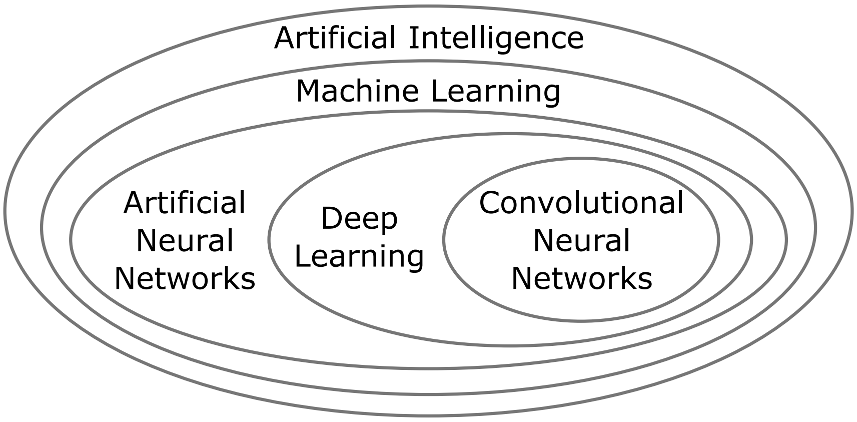


Figure 2.7: Hierarchy of fields related to deep learning.

Machine Learning (ML) algorithms are a type of weak AI, which aims to improve model performance automatically based on experience [13]. The learning problems, that can be solved using ML, can be categorized into three main types:

- supervised learning, where the model learns a mapping between input data and label from training examples,
- unsupervised learning, where the model learns to describe or extract relationships only from the input data,
- and reinforcement learning, where the model learns to operate in an environment from feedback.

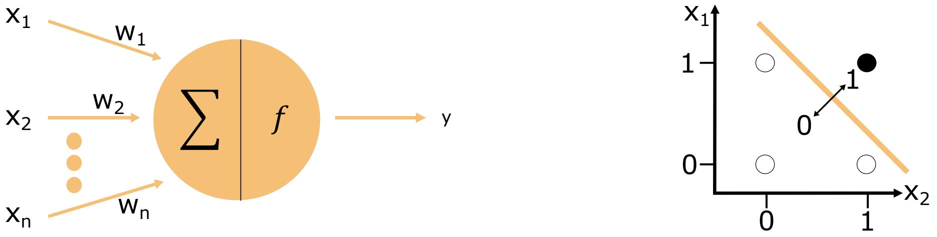
Artificial Neural Networks (ANNs) are a specialized class of supervised ML and include DL. DL differs from other ANN methods in that it employs a higher number of layers, thus learning feature representation at successively higher, more abstract layers [14]. In this work CNNs are used for DL.

2.3.1 Artificial Neural Networks

ANNs are a supervised learning system built from simple elements, called artificial neurons or *perceptrons*. The design of an individual perceptron is based on the structure of a biological neuron. A biological neuron receives electrical impulses from axons of other nerve cells via the so called dendrites. If the combined electric pulses from the dendrites are high enough to pass the firing threshold, it will activate an action potential in the neuron and send it down its axon to the dendrites of the following neurons.

Similar to this, a perceptron receives a series of input variables x_i. Each variable is weighted with a weight w_i. The weighted inputs are summed up and passed to the activation function. If the condition for activation is not full-filled the perceptron output y is zero, which would be equal to a biological neuron not firing. Otherwise the activation function return a non-zero output. A schematic representation of the basic set up of a perceptron is shown in Figure 2.8a. There is one major limitation when using a single perceptron: It can only solve linearly separable problems (see Figure 2.8b as an example).

To be able to solve more complex problems, multiple perceptrons can be combined into Multi-Layer-Perceptrons (MLPs). The perceptrons are organized in layers. All



(a) Perceptron (artifical neuron) **(b)** Linearly separable problem

Figure 2.8: (a) shows the structure of a perceptron with the input variables on the left, followed by the weighted sum, the activation function f and finally the output on the right. (b) shows a graphical interpretation of the logical conjunction of two binary variables x_1 and x_2. This problem is linearly separable as shown by the orange decision border.

perceptrons in a layer receive the same inputs, but have their respective weight for each input. The outputs from all perceptrons in one layer are passed as inputs to all perceptrons in the consecutive layer.

2.3.2 Convolutional Neural Networks

CNNs are a specialization of ANNs for efficient processing of Cartesian data, such as images [15]. Their structure is based on the description of the mammalian vision system by Hubel and Wiesel [16]. As the name indicates, these networks employ so called *convolutional layers*. An ANN with at least one convolutional layer is referred to as a CNN. CNNs commonly feature *pooling layers*. They can also include *fully connected layers*, which are equal to the layers in a MLP. A specific arrangement of layers is referred to as a network *architecture*. An example for a simple classification architecture is shown in Figure 2.9. Individual layers as well as specific architectures are explained in the following paragraphs. More in depth information can be found in [17].

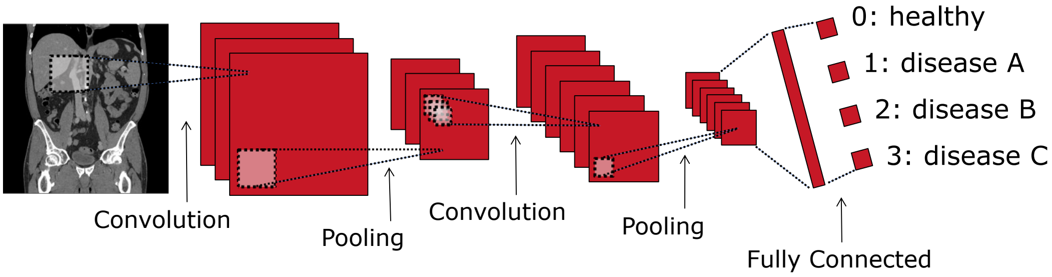


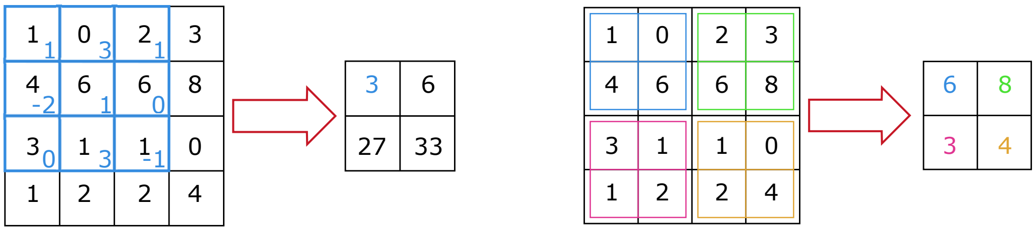
Figure 2.9: Basic CNN architecture for image classification consisting of two convolutional layers, two pooling layers and a fully connected layer.

2.3.2.1 Convolutional Layer

In contrast to MLPs, CNNs do not learn individual weights for each input value, but filter kernels instead. A filter kernel is a grid of weights. The size of this grid is defined by the number of input channels times predefined values for each spatial dimension. The kernel extend for the spatial dimensions is usually chosen to be three. The number of input channels in the first layer is equal to number of color channels (three, when using RGB color space, one for grey value images such as CT or MRI).

The filter kernel is moved across the input with a predefined step size called stride. At each position, the kernel weights are multiplied element-wise with the input values. The sum of these products is one entry of the resulting feature map. The size of the feature map depends on the kernel size, the stride and the padding pattern. If no padding is applied the feature map is smaller than the input, as the filter can only be applied at positions where the entire kernel is contained entirely within the image. Zero padding can be applied to increase the size of the input, so that the final feature map equals the original input in size. An example for a 2D convolutional layer is shown in Figure 2.10a.

Each convolutional layer learns not only one, but a whole set of kernels. The number of kernels determines the number of resulting feature maps, which is also the number of input channels for the next layer. The number of kernels in a CNN dictates the capacity of the network and therefore also the complexity of the tasks it is able to learn. As the kernel weights are used repeatedly across the whole input, instead of having a single weight for each input value, convolutional layers have far less parameters than MLPs. They also limit the interaction of features spatially.



(a) Convolutional layer **(b)** Max-pooling layer

Figure 2.10: (a) shows the application of a 2D convolution with a 3×3 kernel and a stride of one without zero-padding to a one channel input. The kernel, the kernel weights and the result of the shown kernel position are displayed in blue. (b) shows max-pooling using a 2×2 window and a stride of two. Each window position and the respective result is color coded.

2.3.2.2 Activation Function

The convolution operation alone is linear. To achieve non-linearity, the resulting feature map is processed via an activation function. While earlier ANNs often applied the tangens hyperbolicus, CNNs mostly employ Rectified Linear Units (ReLUs) (Equation 2.6) or extensions like LeakyReLUs (Equation 2.8) or Exponential Linear Units (ELUs) (Equation 2.7).

$$ReLU(x) = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{if } x \leq 0 \end{cases} \tag{2.6}$$

$$ELU(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha(e^x - 1)\text{ , where: } \alpha \geq 0 & \text{if } x \leq 0 \end{cases} \tag{2.7}$$

$$LeakyReLU(x) = \begin{cases} x & \text{if } x > 0 \\ 0.01x & \text{if } x \leq 0 \end{cases} \tag{2.8}$$

The final layer of a CNN for classification or segmentation should return probabilities. This is achieved by using the softmax activation function (Equation 2.9). It returns normalized class probabilities for all C classes. In binary cases, the sigmoid function (Equation 2.10) can also be used.

$$\text{softmax}(x) = \frac{e^x}{\sum\limits_{i=1}^{C} e^{x_i}} \tag{2.9}$$

$$\text{sigmoid}(x) = \frac{x}{1 + e^{-x}} \tag{2.10}$$

2.3.2.3 Pooling Layer

Pooling layers reduce the spatial size of feature maps by replacing the values with a statistic of the nearby outputs. The most commonly applied pooling layer is max-pooling [18]. Similar to the convolutional layer, a window is slid across the input. The maximum input value in the window is returned for the current position. To reduce the size of the data the stride is set to a value higher than one. Pooling reduces memory requirements of the following layers and makes the network invariant to small translations of the input. Max-pooling using a 2×2 window and a stride of two is shown schematically in Figure 2.10b.

2.3.2.4 Architectures

Specific combinations of layers and operations are referred to as architectures. Although CNNs were invented in the 1980s, their breakthrough only came in 2012 when the *AlexNet* architecture [19] was developed and won the Image Net challenge for image classification. AlexNet consists of five convolutional layers, three max-pooling layers and three fully connected layers. Since then, CNNs have been dominating image classification. Due to the increased processing abilities of Graphical Processing Units (GPUs) together with the availability of large scale data sets [14] the training of deeper networks like *VGG-16* [20] and *ResNet* [21] became possible. A more detailed look at these three architectures is given in Chapter 7.

Translation into segmentation followed quickly with the use of Fully Convolutional Neural Networks (FCNNs) [22]. These architectures are build based on the previously mentioned classification architectures, but don't contain fully connected layers. Thus they only contain sliding-window operations, which allow for training on smaller samples and subsequent application to arbitrary lager images. Higher performance was achieved by auto-encoder architectures. These networks consist of an encoder and a decoder path. The encoder's structure is similar to a FCNNs. It's purpose is to extract features from the image data and condense it. The result of the encoder is a set of many feature maps with a spatial resolution lower than the original image. The decoder upsamples the information again, so that in the end a segmentation mask can be returned.

The upsampling can be enhanced by passing information from the encoder to the decoder as proposed in the SegNet [23]. A major breakthrough for medical image segmentation was the U-Net [24], which extends the known auto-encoder architecture with skip connections, where entire feature maps are passed from encoder to decoder. The U-Net was also adapted to 3D image segmentation [25, 26]. U-Net and it's 3D extension V-Net are described in more detail in Chapter 6. Auto-encoder architectures are also commonly applied for image regression tasks such as image enhancement. A substantial number of specialized architectures have been proposed in recent years for applications such as image generation [27] and style transfer [28].

2.3.3 Training

CNN training is supervised, therefore requires labeled training data. The examples used for training are usually a major subset (60%-80%) of the whole data set. The remaining data is later used to test the network. In addition to test and training data, samples can be used for validation during the training process. To robustly estimate the performance of a network on a limited data set, k-fold cross validation can be employed. The parameter k states the number of disjoined test sets the data is split into. The model is trained k times. In each iteration the data, which is not in the test set, is used for training (and validation).

For the training process, samples are preprocessed, which usually includes normalization, data augmentation and batching. A more detailed description of these steps is given in Section 2.3.3.1. The preprocessed samples are fed into the network to get a prediction. This prediction is compared to the labels via a loss function (see Section 2.3.3.2). This metric is then minimized by adapting the network parameters. More details on the optimization are explained in Section 2.3.3.3. The CNN training pipeline is visualized schematically in Figure 2.11.

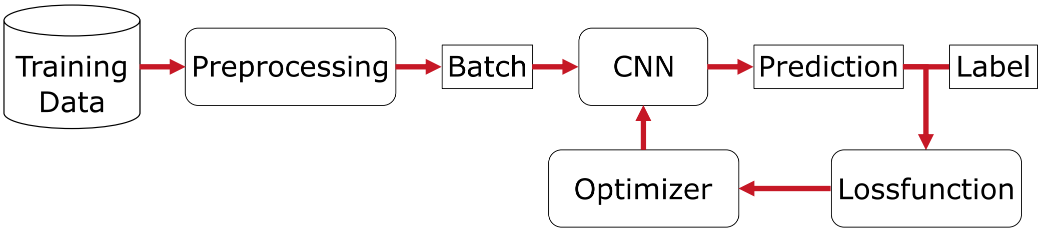


Figure 2.11: Schematic pipeline of CNN training: Preprocessed data is continuously fed to the training loop. The network prediction is then compared to the label via the loss function. Based on the loss, the optimizer computes the parameter updates. After the update the cycle is repeated using the next batch.

How long a network is trained is measured in *epochs*. An epoch has finished once the network has seen every training sample once. Ideally the network should learn to solve the given task well on the training data and then achieve a similar performance when applied to the test data. However, if the training was too short, the model might have *underfitted*. This means that the performance is low on the test as well as on the training data. In contrast to this, the model can also *overfit* on the data. In this case the network achieves high performance on the training data, but performs poorly on the test data. This lack of generalization can be combated by including regularization into the training. Several methods are introduced in Section 2.3.3.4.

2.3.3.1 Preprocessing

The first step in prepossessing is the standardization of the image data. Tomographic data is measured using a specific spatial resolution, which can differ between scans. As only the image values are passed to the CNNs during training, the networks are not able to interpret varying resolutions. Image data is therefore resampled to a common spacing for CNN training.

Similarly, the intensity range of the samples should be similar. To achieve this the image values are normalized. The exact method depends on the modality. As the densities in CT data reflect physical properties, they are often windowed. This means that values outside the relevant interval are mapped to the minimum and maximum values respectively. The truncated values are then mapped into the interval $[-1, 1]$ or $[0, 1]$. Alternatively, the image data can be normalized to have a zero mean and a standard deviation of one.

Data in the medical domain is often limited. Therefore, data augmentation is used to simulate a larger data set and thereby prevent the network from overfitting [29]. This includes a range of methods to generate diverse samples. Data augmentation can either be done before the training process and the individual samples can be saved (*offline*) or it can be done during the training process (*online*). Depending on the data set and how strongly the data set is inflated, offline data augmentation may require extensive amounts of storage. Online data augmentation on the other hand results in slower training due to the additional computational effort.

Horizontal and vertical flipping are common augmentation techniques in the photo domain. In the medical domain however they should be used with caution as the human body in not symmetric. For example, horizontal flipping abdominal images in the transveral or coronal plane simulates a rare medical condition called situs inversus. However, in cranial data horizontal flipping may be suitable depending on the task. For image regression and segmentation the image as well as the label have to be flipped.

Artificial noise can be added to the data. This augmentation should be used in consideration of the imaging system. Based on the system, the noise distribution as well as the distribution parameters have to be determined. The value range of the entire sample can be altered by adding a constant value to all image intensities. This intensity jittering should be performed in a magnitude suitable to the value range of the data. For example, when working on CT data jittering should not exceed single-digit HUs as otherwise tissues would be mapped onto a false section of the HU scale.

Geometric transformations are a commonly applied data augmentation method. However, changing the geometry requires resampling of the image data, which should be performed with an appropriate interpolator and in consideration of the image spacing in medical data. This differs from preprocessing in the photo domain, where interpolation is not dependent on the resolution grid. Care has to be taken, when applying these methods to segmentation as this requires the adaption of the image data as well as the label. This is also the case for image regression tasks. Affine transformations such as scaling, rotation and translation are often used, but elastic deformations are also possible. Each of these methods is governed by parameters which should be chosen to mimic possible real-world transformations. Extreme data augmentation, that generates samples that do not represent realistic extensions of the data set, can deteriorate the network performance.

Cropping the images or extracting patches from them has a similar effect as translation. Training on patches also usually allows for a larger batch size when working close to the memory limit of the GPU. Managing which image region patches are sampled from can be beneficial for the training. For example, peripheral image areas which mostly show the air around the patient can be excluded from sampling. Classification and especially segmentation tasks often suffer from an imbalance between the classes. This problem can be alleviated using an appropriate sampling scheme.

The samples are fed to the network in small subsets called batches. The number of samples in a batch is referred to as the batch size. Similar to the sample extraction, the batching process can also be adjusted to combat the class imbalance in segmentation or classification.

2.3.3.2 Loss Function

In each training step the current batch predictions are compared to the batch labels via a loss function. This function returns a single loss value for the batch, describing how well the predictions match the labels. During the optimization this value is minimized, therefore the loss function should return high values for incorrect predictions and low values for correct predictions.

Choosing an appropriate loss function is essential for the successful training of a CNN. For classification the cross-entropy loss is the state of the art. This function can also be used for segmentation. Another commonly deployed loss functions for segmentation is the Dice loss [25]. In classification and segmentation, the loss is computed between the predicted probabilities and the one-hot encoded labels. For regression tasks the mean absolute error loss as well as the mean squared error loss are commonly used. These compute the error between the predicted values and the label values. Training can also be performed with a combination of different loss functions [30].

In Table 2.1 state of the art loss functions are listed with their definitions and areas of application. Predicted probabilities are denoted as P, predicted values as $\hat{Y}$ and the labels as Y. The number of voxels in a batch is denoted by N. Probabilities, values and labels of the individual voxels and classes are identified via the voxel index i and the class index c.

2.3.3.3 Optimization

During each training step the parameters of a CNN are adapted to decrease the loss value on the current batch. How strongly and in which directions each parameter should be changed is computed using *stochastic gradient descent*. This requires the computation of the gradient with respect to the loss value for every single parameter in the network, which is done via *backpropagation*.

Gradient descent is an iterative optimization algorithm that minimizes an objective value by changing the parameters along the negative gradient direction. The gradient

Loss Name	Definition	Application
cross-entropy	$\mathcal{L}_{CE}\left(Y,P\right) = -\sum_{c=1}^{C}\sum_{i=1}^{N} y_{i,c}\, log(p_{i,c})$	Classification, Segmentation
Dice	$\mathcal{L}_{DSC}\left(Y,P\right) = 1-\frac{2\cdot\sum_{c=1}^{C}\sum_{i=1}^{N} p_{i,c}\cdot y_{i,c}}{\sum_{c=1}^{C}\sum_{i=1}^{N} p_{i,c}+\sum_{c=1}^{C}\sum_{i=1}^{N} y_{i,c}}$	Segmentation
mean absolute error	$\mathcal{L}_{MAE}\left(Y,\hat{Y}\right) = \frac{1}{N}\sum_{i=1}^{N} \lvert y_i-\hat{y}_i\rvert$	Regression
mean squared error	$\mathcal{L}_{MSE}\left(Y,\hat{Y}\right) = \frac{1}{N}\sum_{i=1}^{N} \left(y_i-\hat{y}_i\right)^2$	Regression

Table 2.1: Commonly applied loss functions for different image processing tasks.

is computed across all samples which becomes very expensive with large data sets. To economize on the computational cost, the gradient can instead be approximated on a small subset of the observations. This approximation is called stochastic gradient descent and is used to train CNNs. Each iteration step is computed on a small subset of samples (called a batch as introduced in Section 2.3.3.1).

Let $\mathcal{L}[W_t]$ be the loss value of the batch in iteration t using the set of weights W_t in the network. The gradient $\nabla\mathcal{L}[W_t]$ is the vector of partial derivatives of $\mathcal{L}$ with respect to the individual weights $w_{i,t}$:

$$\nabla\mathcal{L}[W_t] \equiv \left[\frac{\partial\mathcal{L}}{\partial w_{0,t}}, \frac{\partial\mathcal{L}}{\partial w_{1,t}}, \cdots \frac{\partial\mathcal{L}}{\partial w_{n,t}}\right]. \tag{2.11}$$

$\nabla\mathcal{L}[W_t]$ shows the direction and rate of the fastest increase of the loss value. As the goal is to minimize the loss, the negative gradient is used to define the weight changes ΔW_t:

$$\Delta W_t = -\eta\nabla\mathcal{L}[W_t]. \tag{2.12}$$

This update step is scaled by the learning rate η. η is set to a value smaller than one, so that the weights are adapted in small steps. To apply these weight changes, ΔW_t is simply added to the current weights W_t. The update step for an individual weight therefore is:

$$\begin{aligned} w_{i,t+1} &= w_{i,t} + \Delta w_{i,t}, \\ \text{where:} \quad \Delta w_{i,t} &= -\eta\frac{\partial\mathcal{L}}{\partial w_{i,t}}. \end{aligned} \tag{2.13}$$

Several extensions to stochastic gradient descent have been proposed. Using a momentum term, the next update is defined as a linear combination of the current gradient and an exponential moving average of the last gradients. Momentum improves the speed of convergence as it prevents oscillations [31].

Various solutions suggest the use of per-parameter learning rates. These include Adagrad (normalization of learning rates by sum of all previous gradients) [32], Adadelta (normalization of learning rates by sum of window of previous gradients) [33] and RMSProp (normalization of learning rates by sum of a moving average of the squared previous gradients). In 2014, the Adam optimizer was proposed, which combines the advantages of Adagrad and RMSprop [34]. Adam requires three parameters to be set in addition to the learning rate: the exponential decay rate for the first moment estimates (β_1), the exponential decay rate for the second-moment estimates (β_2) and an ϵ to prevent division by zero. The Adam update step is demonstrated in Equation 2.14 to Equation 2.18.

$$M_t = \beta_1 \, M_{t-1} + (1 - \beta_1) \, \nabla \mathcal{L}_t(W_t) \quad \text{(1st moment estimate)} \tag{2.14}$$

$$R_t = \beta_2 \, R_{t-1} + (1 - \beta_2) \, \nabla \mathcal{L}_t(W_t)^2 \quad \text{(2nd moment estimate)} \tag{2.15}$$

$$\hat{M}_t = \frac{M_t}{(1 - \beta_1^t)} \quad \text{(1st moment bias correction)} \tag{2.16}$$

$$\hat{R}_t = \frac{R_t}{(1 - \beta_2^t)} \quad \text{(2nd moment bias correction)} \tag{2.17}$$

$$W_{t+1} = W_t - \eta \, \frac{\hat{M}_t}{\sqrt{\hat{R}_t} + \epsilon} \quad \text{(update)} \tag{2.18}$$

The individual computation of the partial derivatives for all network weights would be extremely computationally expensive. Backpropagation is employed to reduce the computational cost and compute the partial derivatives simultaneously. The algorithm consist of two main steps: the forward step and the backward step. In the forward step, the current batch is put into the network and the intermediate result of every single computation in the network is saved. These intermediate results are then used in the second step. Backpropagation gets it's name from the basic principle of propagating the error from the end of the network back to the front. Starting from the loss value the partial derivatives of each computation operation in the network with respect to the input variables are determined based on the intermediate results from the forward step. The partial derivatives of all parameters in the network with respect to the loss value, can then be computed using the chain rule.

2.3.3.4 Regularization

Regularization refers to techniques that can be used to prevent overfitting. These methods intent to minimize the error by adapting the training to learn an appropriate fit on the training data.

Weight regularization is performed by adding a penalty term based on the network weights to the loss function. The penalty term is scaled to have a suitable magnitude in reference to the loss value. *L1 regularization* uses the sum of the absolute values of the weights. More commonly the *L2 regularization*, which computes the Euclidean norm of the weights, is applied for DL. L2 regularization encourages the network to learn small weights, which in turn results in a potentially more stable model, that is less likely to overfit.

Network generalization can also be improved using *drop out* [35]. This method simulates a set of different architectures by probabilistically dropping out nodes in the network. Dropping out refers to ignoring a random subset of layer outputs in each training step. Due to the constantly changing subset, each training step is performed with a different network configuration. By doing so the co-adaptation of neurons is prevented and a more robust model achieved. Drop out has originally been proposed for fully connect layers, but extension for convolutional layers have since been developed.

Simultaneously adapting all parameters introduces an internal covariate shift. This refers to the change of the distribution of network activations due to the update step and therefore a change of the input distribution for subsequent layers. This can be averted using *batch normalization* [36]. These layers learn to normalize the activation distribution of each batch to a mean of zero and a standard deviation of one.

How long a network should train depends on the task as well as the data and the complexity of the model. Choosing the right amount of training epochs is not trivial. Generally, CNNs should only be trained as long as the network performance still increases. This can be measured using an appropriate metric on the validation set. Using this information, *early stopping* can be applied to the training process. This requires the formulation of convergence criterion. This criterion should be fullfilled, once the performance on the validation set no longer improves, but it should take into account fluctuations during training. A common approach is to average the validation accuracy over several epochs. Upon convergence the training is stopped. For further improvement the best of the recent networks can be chosen for testing based on the validation accuracy.

3. Training Pipeline Implementation

The training pipeline is implemented as a Python framework. For the DL functionalities the TensorFlow library [37] is used, while handling of the medical image data is done with SimpleITK [38]. The following description is focused on the latest version of the pipeline implemented with TensorFlow 2.0. The main functionalities of the framework are available from two main classes: a loader class, that directly loads image data from medical image files and performs preprocessing and batching, and the network class that encompasses functionalities for training and inference. Both classes are implemented in a basis module that is applicable to all tasks. More specialized loader and network classes for segmentation, regression and classification inherit from the basis classes. Project-specific versions can then be generated from these classes, should further alterations be necessary.

3.1 Basis Module

The basis module provides the basic functionalities for the two main classes *data loader* and *network*. The basis module further contains implementations of loss functions, evaluation metrics, commonly used building blocks of CNN architectures and commonly required image processing methods

The basic data loader wraps data loading and preprocessing via Simple ITK with TensorFlow's *tf.data.Dataset* Application Programming Interface (API). Sample mining and processing are done online, So each file is loaded in each epoch. This also enables online data augmentation, where new samples are generated each epoch. The data set is passed to the loader as a list of filenames of the image files. New samples are generated as long as the sample buffer is not full. In this case a filename is passed from the *tf.data.Dataset* API to a wrapper function. This wrapper function is implemented in the task specific loaders. While the workflows differ slightly, they generally consist of the following steps: loading data and label, resampling, normalization, sample extraction and data augmentation. The samples are returned to the *tf.data.Dataset* API and converted into tensors. The sample tensors are stored in the sample buffer and batches are generated from the buffered samples. All operations in the data loader are implemented for 2D and 3D sample generation. It is also possible to extract several 2D slices and return them as channels, which is referred to as 2.5D. The data loader can be run in one of three modes:

- *train*, which includes sampling, data augmentation with batching for a given number of epochs,
- *validate*, which samples and batches the validation data once and can then be reinitialized,
- and *apply*, which samples the volume for inference.

The basic network provides three main functionalities: training, fine-tuning and application. Upon initialization network options are checked for compatibility. When a model is initialized for training, a new model is build. For fine-tuning and application an existing model is loaded, however for application the weights are not trainable. The model is build and stored as a *tf.keras.Model()*. The basic network additionally provides implementations of several optimizers and regularizers. All CNN building blocks are implemented in 2D and 3D and the correct dimension is chosen automatically based on the input. Specific architectures are implemented in the specialized modules. The basic network serves as a super class from which the task networks inherit. A specific architecture only implements the model building function and inherits all other functionalities from its parent classes.

During training the average training accuracy across the epoch is tracked. At the end of the epoch the current weights are saved and the network is tested on the validation data set. TensorBoard, TensorFlow's visualization toolkit, summaries are generated at the end of the epoch for a variety of images, scalars and distributions. Which specific summaries are generated is task depend and therefore implemented in the specialized modules. If early stopping is enabled and the minimum number of epochs has been passed, the convergence criterion is checked at the end of each epoch and should it be fulfilled, the training stops.

3.2 Specialized Modules

The functionalities of the basis module are extended in three specialized modules for image regression, semantic segmentation and classification. Parameters and flags for operation modes are stored in a configuration file. For image regression and segmentation the data loader reads the image data and the label image. For classification the data image is read together with the class label. For segmentation and classification the label is converted to one-hot encoding. Rotation, scaling, deformation, flipping, intensity jittering and noise addition are available data augmentation methods for all tasks, that can be activated and parameterized in the configuration file.

During training, sample selection is distributed through the image volume. Possible sample centers are first identified across the z-axis and then in the x-y-plane. The specialized segmentation module extends the loader by three sample mining schemes:

- *uniform*, which uses a uniform random distribution for center selection in the x-y-plane,
- *constrained* $\mu\sigma$, which constrains the possible centers to the area of the average label coordinates,
- and *constrained label*, which restricts sample centers to non-background voxels.

The constrained label mode is also available for image regression. It restricts the sample centers to regions in the label image with values above a selected threshold. For semantic segmentation, a ratio-based data loader is available (see Chapter 6). In this case the sample mining process selects background and object samples based on a ratio-parameter. This specialized loader, is only available for the modes train and validate, as it is designed to combat the class imbalance during training.

Each of the specialized modules integrates suitable loss functions into the network class. Additionally, the final activation function is set according to task and loss. For all three tasks, the scalar summaries include loss, accuracy, regularizer and objective (loss together with regularizer). Image summaries for regression include the sample input, label and prediction. For segmentation the image summaries include sample input, label, prediction and predicted probability maps of the individual classes. For classification the sample image is visualized with the label class included as text and the prediction color coded (correct: green, incorrect: red). In the case of 3D samples the center slice is used for the image summaries. For multi channel data, each input channel is shown in an individual image summary. For all tasks the histogram of the input, labels and predictions are included in the summaries. For segmentation and classification, the histograms of the predicted probabilities are also included.

Due to memory restrictions of the GPU, the image volume is split into smaller sections for application in image regression and segmentation. The network predictions are then fused and resampled before image export. For classification only a single sample is loaded and the prediction is written into a text file.

4. "Simulation-Based Deep Artifact Correction with Convolutional Neural Networks for Limited Angle Artifacts", *Z Med Phys, doi: 0.1016/j.zemedi.2019.01.002*

4.1 Introduction

Minimal invasive surgery is one of the fastest growing medical disciplines because they are less stressful for patients. The drawback of these procedures is the heavy reliance on imaging modalities such as X-ray and thus a radiation exposure of patients and medical personnel. In particular, 3D imaging protocols in CBCT expose patients to a high amount of ionizing radiation. But on the other hand, CBCT systems offer physicians great flexibility. In contrast to MSCT or MRI, C-arm systems do not limit the access to the patient and can be moved around in the intervention room, which is crucial for interventions. Because of the necessity for CBCTs in interventions, low dose imaging is one of the main research goals in the CT community [39–42]. One possible approach to reduce dose is to manipulate the scan trajectories [43, 44]. We have implemented an experimental framework to sample arbitrary scan trajectories using an experimental C-arm device and a step-and-shoot technique [45]. In particular, we investigated cTS trajectories. These trajectories have been well examined in literature and are one of the simpler two-axis rotation scan trajectories [46–48]. By manipulating the acquisition strategy, we can directly manipulate the Fourier domain sampling and thus the reconstruction quality distribution in the three spacial directions. This allows to scan and detect critical objects at lower dose exposure than conventional CBCTs at the cost of reduced morphological details and limited angle artifacts [49]. In medical applications where the accurate detection of objects such as needles or contrast enhanced blood vessels is more important than the morphological structure, interventional tomosynthesis is a promising low-dose approach. In conventional CT, limited angle artifacts are also prevalent in undersampled scans. Therefore, in the last years, many approaches have been proposed to address this. In particular, Total Variation (TV) was successful [50–52]. Newer publications on TV-regularization have improved the isotropical TV-filters by introducing an adaptive step width [53] or better sensitivity for anisotropical structures [54, 55]. Also the combination of prior knowledge with TV has been successfully demonstrated [56].

Another approach to mitigate limited angle artifacts, is the use of machine learning methods. In recent years deep learning methods have been successfully applied to a wide range of medical image processing problems [1, 57]. Reducing the two-dimensional streaking artifacts and noise in image reconstruction of MSCT has been a focus point, especially in the 2016 Low Dose CT Grand Challenge [58]. FCNNs can be trained to mitigate artifacts by directly predicting the corrected images [54, 59]. However, Han *et al.* have shown that the residual artifact manifold is topologically simpler than the corrected image manifold [60]. They refer to the learning of correction maps/artifact residuals as Deep Residual Learning. To avoid confusion in regards to residual network architectures, we refer to this approach as Deep Artifact Correction (DAC). Han *et al.* have demonstrated the successful application of DAC for limited angle artifact correction in sparse-view MSCT scenarios [61] and the superiority in comparison to TV [62]. Similarly, Lee *et al.* applied DAC to remove aliasing artifacts in accelerated MRI [63].

As FCNNs are a supervised machine learning method they require a label for every training sample. However, a common problem in the field of medical imaging is the inaccessibility of a ground truth. Han *et al.* derived their labels from reconstructions using the full number of available projections. We briefly tested a similar approach for three-dimensional limited angle artifacts using high-dose CBCT reconstructions, but we saw that simultaneously to learning to correct the limited angle artifacts, the network also started to introduce CBCT artifacts. An alternative approach to evade the missing ground truth problem is using simulations of the imaging procedure to generate the training data [64]. Simulation-based DAC has shown to be highly effective for scatter correction [65, 66]. For these approaches the ground truth is generated by simulating noise-free reconstructions. Simulated training data has also proved adequate for the estimation of physiological parameters from multispectral images [67]. CNNs trained on CT data augmented with simulated metal artifacts have been shown to correctly reduce real metal artifacts [68].

We propose DAC using simulations based on digital phantoms to reduce the three-dimensional limited angle artifacts in cTS scans. Our simulations were generated with an anthropomorphic phantom that shares the same morphological structures as human patients. In this paper, we focus on the mitigation of the large-scale three-dimensional limited angle artifacts of cTS. Although, noise and scattering are two main error sources, they cause no systematic edges shifts or locally large distortions. Therefore, we decided not to consider these error sources in this work.

In this research work, we want to address three main issues:

1. Can a FCNN-approach mitigate three-dimensional limited angle artifacts caused by a cTS scan trajectory?
2. Are the features learned by the FCNN generic enough to correct limited angle artifacts in arbitrary images?
3. Is it possible to sufficiently train a FCNN with generic simulation models for the application on real cTS scans?

To investigate these issues, we apply DAC using an adaptation of the U-Net architecture to learn correction maps for limited angle artifacts [24]. We train the network using simulations with an anthropomorphic digital phantom.

4.2 Material & Methods

In the following section we give an overview of interventional tomosynthesis, the used simulation data, DAC and the experiments we performed.

4.2.1 Interventional Tomosynthesis

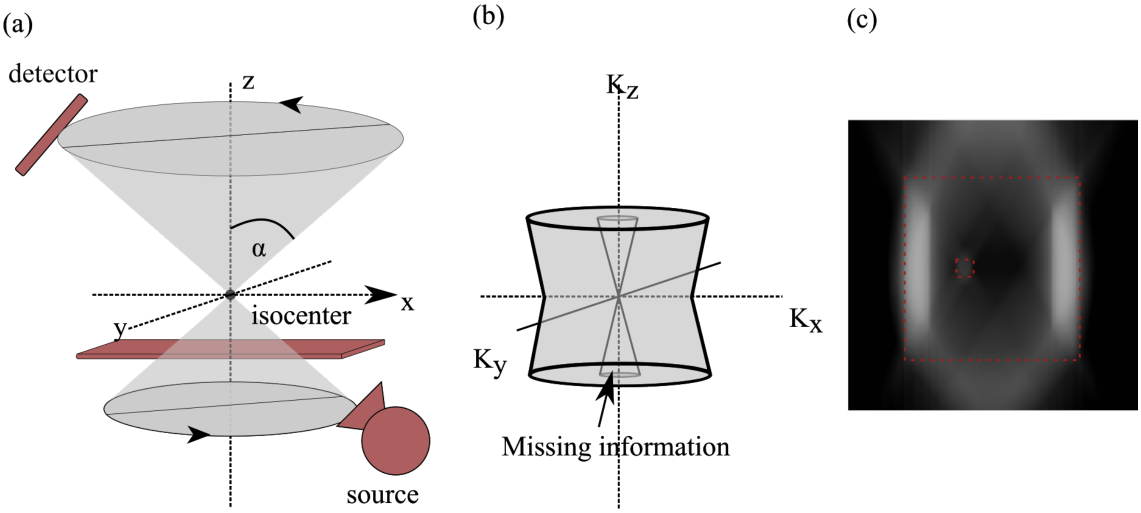


Figure 4.1: Circular Tomosynthesis. a) shows the scan trajectory, b) the corresponding Fourier space sampling with the missing information due to a limited angle scan, c) the resulting artifacts, the dotted red lines indicate the artifact-free geometries of the boxes

In contrast to conventional CBCT scan trajectories, cTS is a two axis rotation movement. Source and detector are actuated on two circles above and beneath the patient, leaving a movement-free space in the middle. Figure 4.1a shows such a cTS scan trajectory. The image reconstructions shows limited angle artifacts because the Tuy-Smith condition is violated. Typically, limited angle artifacts are known from sparse view cTS. In these scenarios, the artifacts are almost exclusively two-dimensional and are shaped like streaks (and thus are also known as streaking artifacts). In cTS the limited angle artifacts are visible throughout two planes and have a large-scale three-dimensional structure. Another challenge in cTS, is the difference of absorption values between CBCT and cTS. In cTS the Fourier domain is not sufficiently scanned (Figure 4.1b). Therefore, the values in cTS are significantly lower than in CBCT.

4.2.2 Real Data

For the acquisition of the real cTS, we used our previously published workflow [45]. The scan trajectories were implemented on an ARTIS zeego (Siemens Healthineers, Germany, Forchheim) with a step-and-shoot technique. Due to the experimental nature of the cTS scan-trajectories, the acquisitions are time-consuming (about 30 minutes). Therefore, it was not possible to perform any measurements on patients. To test and evaluate our network, a porcine shin and a porcine rib slab were used as phantoms. The porcine shin was about 3 cm thick and contained a large bony structure in the middle. The porcine rib slab showed a similar thickness and had a

	Tomosynthesis	High-dose CBCT
Half-tomo angle α	25°	-
Number of projections	100	496
Acquisition time	30 minutes	20 seconds
Source-detector distance	120 cm	120 cm
Source-isocenter distance	78.5 cm	78.5 cm
Voltage	70 kVp	70 kVp
Current	55 mA	17 mA
Pulse width	4 ms	3.5 ms
Detector	960 x 1240 pixel	960 x 1240 pixel
Reconstruction volume	512 x 512 x 388 voxels	512 x 512 x 388 voxels
Detector pixel size	0.308 μm	0.308 μm
Voxel size	0.48 mm	0.48 mm

Table 4.1: Real data acquistion parameters

size of about 6 cm x 13.5 cm. Every phantom was scanned twice: Once with cTS and once with high-dose CBCT. The tomosynthesis data sets were then corrected with different networks. The high-dose CBCT data sets with 496 projections were used as ground-truth for the evaluations. The parameters we used for the acquisition are listed in Table 4.1.

4.2.3 Simulation Data

For the simulated data, we used three models: (1) an anthropomorphic phantom, (2) a catheter phantom and (3) a simplices phantom consisting of 15 generic geometric objects:

1. Anthropomorphic phantom: We used the anthropomophic XCAT phantom [69]. We generated phantoms with a size of $512 \times 512 \times 388$ voxels of randomly selected body regions. The shapes of extremities, torso and skull were also randomly deformed. All data sets were then scaled into HU. Additionally, in one half of our training data sets, we set the outer 30 pixels of the axial plane to background (-1000 HU). This minimizes the truncation artifacts in the reconstruction and introduces sharp edges in half the data sets. The rational for artificially enforcing edges was to prevent the network from training only smooth anthropomophic structures. By splitting the training data sets, both sharp edges and smooth anthromophic structures are included in the training data set.
2. Catheter phantom: For this phantom type, we added catheter-like objects to the anthropomorphic phantoms. For the catheter-like objects first a blood vessel map was created. The map was then reduced to a center line representation. We selected randomly a single blood vessel with a minimum length of 50 pixels and highlighted it in the anthropomorphic phantom with a fix HU-value of 5000 HU. These catheter phantoms were used to test whether the networks are able to correctly detect the catheter-like objects, even though they are not included in the training data sets.

3. Simplices phantom: To further test the robustness of the network, we generate geometric simplices with random shapes, sizes and attenuation coefficients. The geometrical shapes included spheres, ellipsoids, cubes and slabs. Every phantom contains 15 inserts that could overlap and/or be hollow. The HU range of the inserts was limited to $[-200, 1000]$ HU to mimic soft-tissue and bone-like tissue.

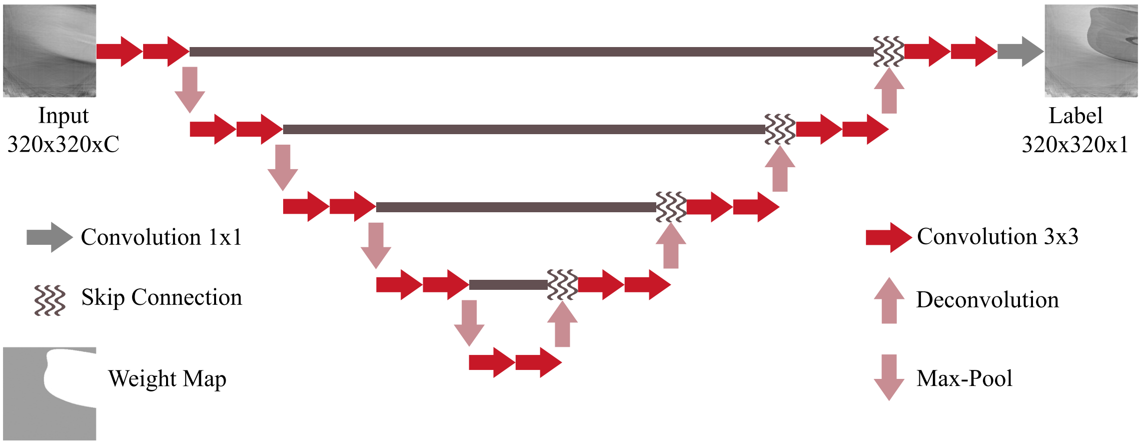
For every phantom a cTS and a CBCT with 100 projections each were simulated. For the image reconstruction, we used the ASTRA toolbox [70, 71].

In our first tests, we figured out that the networks are HU-sensitive and the training was not sufficient when all images edge distortions are exclusively caused by the cTS. Therefore, we deliberately included two error sources to mimic the real imaging system more sufficiently. The heel effect affects every real X-ray imaging system. Due to different absorption length inside the X-ray source the detector is not illuminated isotropically. Instead, the center of the detector is highlighted and a direction dependent gradient is apparent. We imitated this by applying an intensity gradient to all simulated projection data prior to reconstruction. The second error source we incorporated in our simulation is an imperfect calibration of the imaging system. In our simulation, we explicitly calculated the positions of the source, the center of the detector and the basis vectors of the detectors. With a chance of 1/3 each of the 12 parameters were randomly distorted by up to 2%.

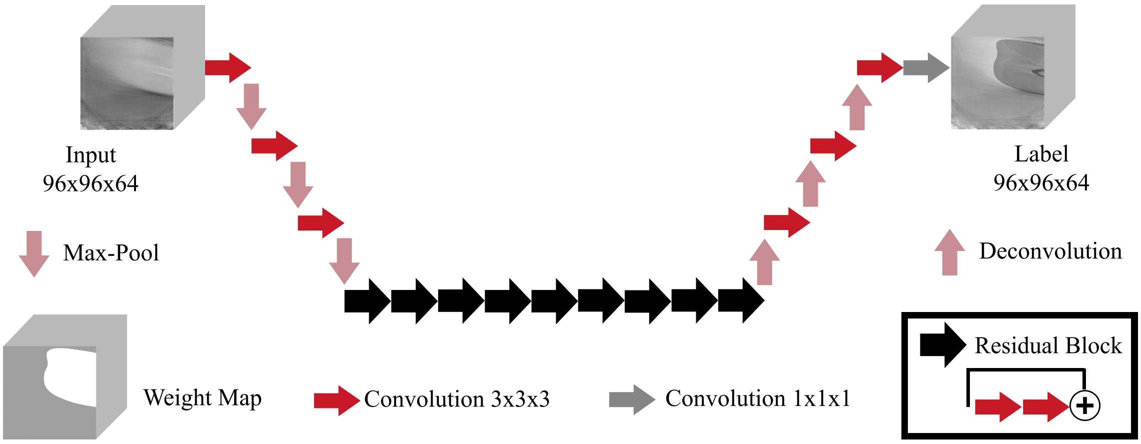
4.2.4 Networks

We explore two architectures for DAC. Similar to Han *et al.* , we adapted the U-Net architecture [24]. The input is first processed by four encoding stages which consist of two convolutional layers each with 3×3 kernels, followed by downscaling layers using 2×2 max-pooling. After two additional convolutional layers in the bottom stage the information is then processed by four decoding stages. In each of these, the features are first upscaled via a deconvolution layer and the result is then concatenated with the feature maps from the corresponding encoding stage forming a skip connection. Each upscaling layer is then followed by two convolutional layers with 3×3 kernels. The stages use 64, 128, 256 and 512 channels. A 1×1 convolution operation is appended to the last stage to generate the final output. We use ReLUs [72] to introduce non-linearity. As limited angle artifacts have a three-dimensional structure, giving the network a volume instead of a single slice could potentially enhance the correction. However, using higher dimensional input also means that the network has to learn a more complex manifold. We therefore compare three versions of this network, referred to as DAC-C1, DAC-C3 and DAC-C5 in the following text. All three learn the prediction of a single-channel label y, but the number of channels C of the input x is $C \in \{1, 3, 5\}$. The center channel is the slice which should be corrected and the adjacent slices in each direction are added to provide the network with depth information. The scan is padded with the edge values, so that three and five slices are also available for the peripheral slices. A visualization of the U-Net architecture can be seen in Figure 4.2a.

While this 2.5D approach does provide some depth information, it is limited because it only considers up to five adjacent slices. To explore the potential of three-dimensional operations on the available hardware, we additionally train a Residual



(a) U-Net (2.5D)



(b) ResNet (3D)

Figure 4.2: U-Net and ResNet Architecture used for DAC. The U-Net receives $C \in 1, 3, 5$ slices from the cTS and is trained to predict the correction map for the central slice. The ResNet receives an image volume and predicts the correction map for the complete volume. The second loss function for both networks is weighted based on the phantom.

Network (ResNet) [21]. The spatial size is first reduced by three encoding stages, which consist of a $3 \times 3 \times 3$ convolutional layer followed by $2 \times 2 \times 2$ max-pooling. The feature maps are then processed by nine residual blocks, which each consist of two $3 \times 3 \times 3$ convolutional layers with a residual connection. This residual connection is formed by an addition of the input feature map to the resulting feature map. To return to the original spatial size the feature maps are then processed by three decoding stages, which consist of a deconvolution and a $3 \times 3 \times 3$ convolutional layer each. The ResNet does not have skip connections. The stages use 16, 32, 64, and 128 channels. Similar to the U-Net, a $1 \times 1 \times 1$ convolution operation is appended at the end. This network predicts the whole correction map for a given input volume. A visualization of the 3D-ResNet architecture can be seen in Figure 4.2b. DAC with the 3D-ResNet is referred to as DAC-3D in the following text.

4.2.5 Training

We use 240 simulations based on the anthropomorphic phantom for training. The cTS scan are preprocessed by truncating the values to the interval $[-2000, 1000]$ HU and then mapping them into the interval $[-1, 1]$. For training we use randomly selected patches. The corresponding label is the correction map between the cTS and the phantom. For the U-Nets we use a patch size of 320x320 voxels, for the 3D-ResNet 96x96x64 voxels.

The networks were trained using two loss functions. In the initial three epochs the voxel-wise L1-norm between labels Y and predictions Y' was minimized:

$$\mathrm{L}_A\left(Y, Y'\right) = \sum_{i=1}^{|Y|} \left|y_i - y_i'\right| \tag{4.1}$$

This loss function allows the network adapt to the correct output domain. However, for the correction we need exact correlation between the edges in the images and the correction maps. The result of the initial three epochs is therefor used as a robust initialization for the training with the second loss function. This loss is a weighted sum of three terms:

$$\begin{aligned} \mathrm{L}_B\left(Y, Y', W, \Phi\right) = &\left(1 - \frac{\sum_{i=1}^{|Y|}\left(y_i - \bar{y}\right)\left(y_i' - \bar{y}'\right)}{\sqrt{\sum_{i=1}^{|Y|}\left(y_i - \bar{y}\right)^2}\sqrt{\sum_{i=1}^{|Y'|}\left(y_i' - \bar{y}'\right)^2}}\right) \\ &+ \lambda_1 \frac{1}{n}\sum_{i=1}^{|Y|} w_i\left(y_i - y_i'\right)^2 \\ &+ \lambda_2 \sum_{j=1}^{|\Phi|}\phi_j^2 \end{aligned} \tag{4.2}$$

Correlation losses have been shown to enhance details in image generation [73]. The Pearson correlation coefficient enforces linear correlation between Y and Y'. It therefore enhances the image structure of the correction maps. $\bar{y}$ and $\bar{y}'$ denote the mean values of the two sets. The weighted mean squared error is necessary to achieve exact regression. The weights W for this term are dependent on the phantom. The masks of object voxels (HU> -1000) is dilated and weighted with $w_i = 1$ while the background has $w_i = 0.5$. This prioritizes object voxels and simultaneously counteracts the class imbalance between objects and background. The third term of the loss function is an L2 regularizer across the network weights Φ. For training we used $\lambda_1 = 10^{-3}$ and $\lambda_2 = 10^{-4}$.

We train the proposed networks for 30 (U-Nets) and 20 (3D-ResNet) epochs using stochastic gradient descent with mini batches of size 8. The Adam optimizer with an initial learning rate of 10^{-4} is used with an epoch-wise learning rate decay. As we use FCNNs the trained filters can be directly applied to the full images in the inference phase. To ensure that after the correction the image no longer contains HUs lower than -1000, we truncate the intensities at this value as a post-processing step.

4.2.6 Experiments

We evaluated the performance of our networks on the three data classes mentioned in Section 4.2.3 and Section 4.2.2. For each experiment we had the following number of test cases:

1. Anthropomorphic Simulations: 60 test cases
2. Catheter Simulations: 6 test cases
3. Simplices Simulations: 25 test cases
4. Proof of Principle on Real Data: 2 test cases

For experiments 1, 2 and 3, the quality was compared to the uncorrected scans and simulated CBCTs. In experiment 4, the corrections were compared to a ground truth generated from a real high-dose CBCT. Image quality was also compared to a TV-reconstruction. Our used TV implementation is a modified isotropic 3D TV-approach as presented in [74].

To evaluate the difference between the DAC corrected cTS and the digital phantom we calculated the Root Mean Squared Error (RMSE) (Equation 4.3). To analyze in more detail where the errors stem from, we also computed separate RMSEs for the objects and the background artifacts. For the simulations we differentiated objects and background based on the phantom data. On the real data scans we used thresholding in combination with a connected component analysis to extract the object mask. We selected all voxels from the background which have a higher HU than -1000 in the cTS as artifact voxels. HUs over 2000 HU were neglected to be more robust against outliers. For the evaluation of the real scans, we neglected the voxels close to the reconstruction volume borders to mitigate artificial sharp edges. All voxels outside of a region of interest around the object were set to -1000 HU.

$$\mathrm{RMSE}\,(y, y') = \sqrt{\frac{1}{n}\ \sum\nolimits_{i=1}^{m} (y_i - y_i')^2} \tag{4.3}$$

4.3 Results

The proposed method was implemented in TensorFlow 1.7 using Python 3.5. Training and inference for the U-Nets were performed on an NVIDIA TITAN Xp. For the 3D-ResNet an NVIDIA Quadro P5000 was used. Training took two days per U-Net and four days for the 3D-ResNet. The correction of a full image volume took 20 seconds on average for U-Nets and 5 seconds for the 3D-ResNet. 3D-ResNet inference had to be divided into subvolumes of 96 slices due to memory constrains.

4.3.1 Anthropomorphic Simulations

The RMSE across the 60 test cases is visualized in Figure 4.4a. The cTS had an average RMSE of 305.63 HU. In comparison, the simulated CBCTs only had a RMSE of 119.93 HU. Using three or more slices further improved the RMSE in comparison to the single slice approach. The difference in RMSE from three to five slices however was smaller than one HU. DAC-C5 achieved a RMSE of 124.24 HU, which is similar to the quality of simulated CBCT. All three U-Nets were able to reduce the cTS error by over 57%. DAC-3D resulted in RMSE=144.07 HU and performed worse than the 2.5D-DAC. A visual comparison between cTS, DACs, CBCT and phantom of one test case is shown in the first two rows of Figure 4.3. 2.5D-DAC strongly improves high contrast edges such as lung and cortical bone borders, while differences in soft tissue could not be restored. DAC-3D was also able to restore most high contrast edges, but the correction maps were more blurrier than the U-Net version. DAC-3D was however better at estimating the body contour. Visual inspection of the data did not show any new structures that were augmented into the image by the networks. Only the 2.5D-U-Nets occasionally overestimated the body contour.

As can be seen in Figure 4.4d and Figure 4.4e, the cTS error of objects is far higher than the background artifact error. The correction of objects benefits from the use of multiple slices, as the error decreases by 4.53% using DAC-C3 and by 9.78% using DAC-C5 compared to the single slice approach. For the object correction DAC-3D achieved the lowest error with 208.01 HU. The background artifacts correction does benefit from multi-slice input. DAC-C3 slightly outperformed DAC-C5. DAC-3D however performed considerably worse than the 2.5 networks. While the background was considerably corrected, DAC-3D failed to mitigate individual strong streaking artifacts, which the U-Nets were able to correct.

4.3.2 Catheter Simulations

On the 6 catheter data sets the cTS had an average RMSE of 309.90 HU and the CBCT in comparison had 159.54 HU. Figure 4.4a shows the RMSE across the 6 test cases. The U-Nets all performed similar with an error reduction of about 49%. While the reduction with DAC-3D was slightly lower with 44%, the 3D-ResNet performed more robustly. Close the catheter DAC-C1 to DAC-C5 failed to restore the HUs of the soft tissue. The 3D approach however was able to estimate the correction maps for this region. Additionally, DAC-C1 and DAC-C3 could not restore the complete catheter, while DAC-3D generated images with the exact catheter location. In Figure 4.3 a visual comparison between cTS, DACs, CBCT and phantom for one test case is shown in the two central rows.

4.3.3 Simplex Simulations

The cTS simulations of the simplices phantoms had an average RMSE of 314.82 HU, while the simulated CBCTs had a RMSE of 129.23 HU. All three U-Nets reduced the RMSE by about 35%. While DAC-C3 performed slightly better, the differences were in the order of 1 HU. DAC-3D reduced the error by 30%. The RMSE across all 25 test cases is shown in Figure 4.4c. The cTS, DACs, CBCT and the phantom of one test case are shown in the last two rows of Figure 4.3.

4.3.4 Proof of Principle on Real Data

The three image planes of the cTS scans as well as of the DAC-corrections and the CBCT-based ground truth are shown in Figure 4.5. The networks were applied to a porcine shin and a porcine rib slab phantom.

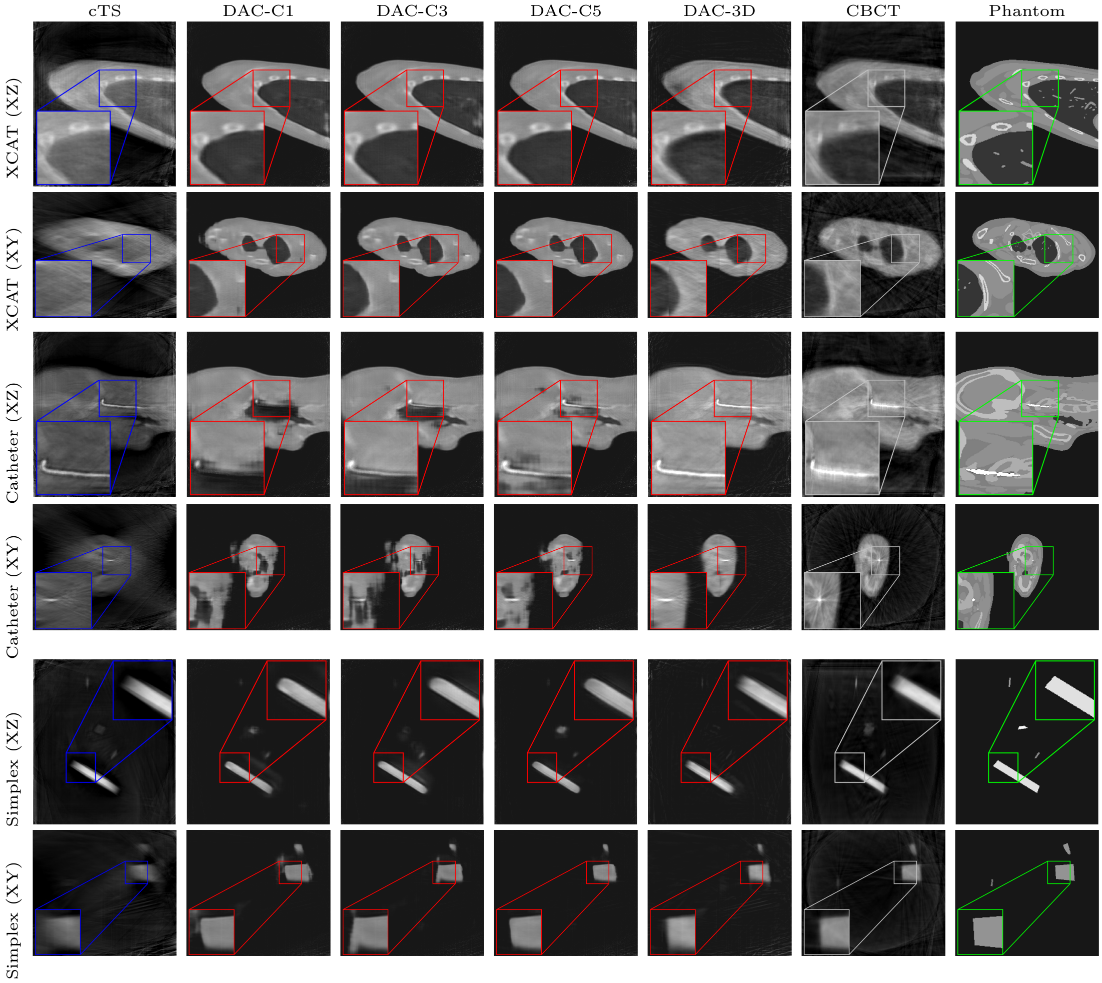
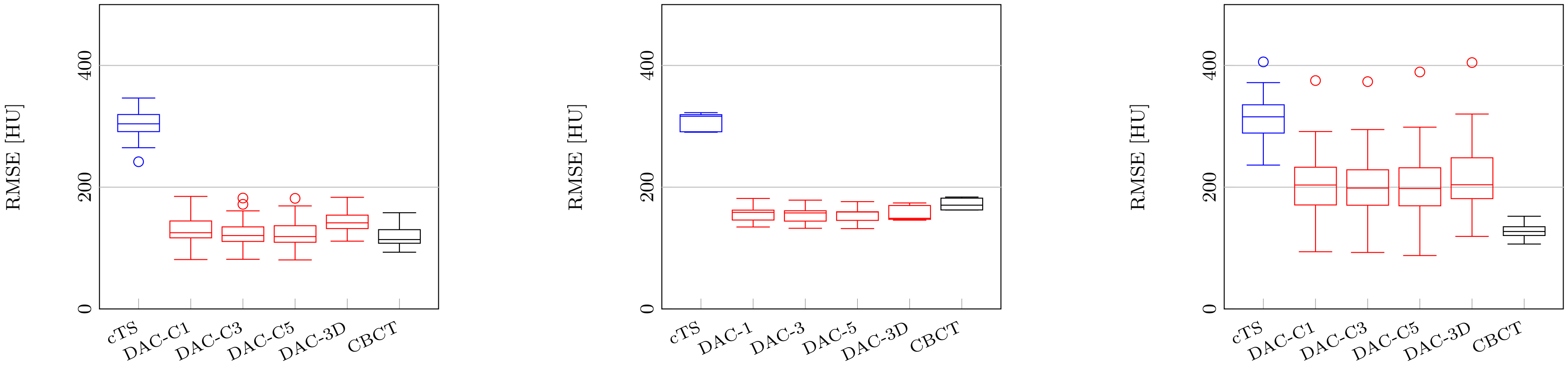
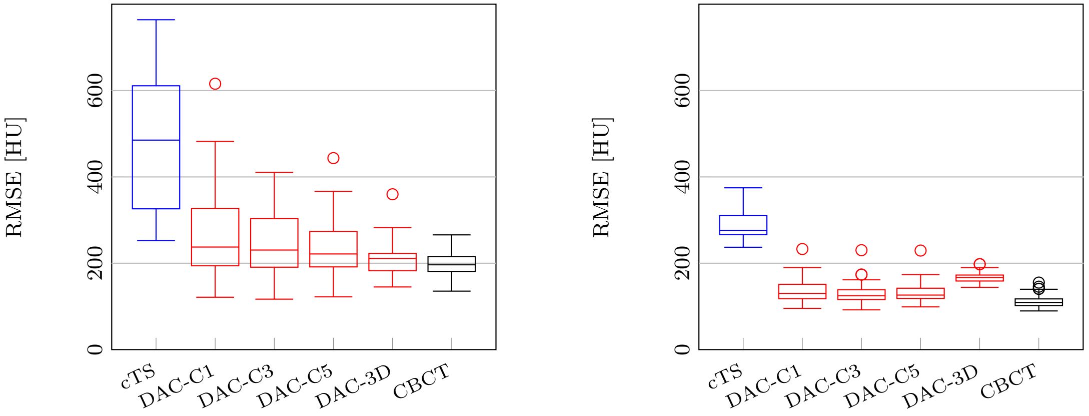


Figure 4.3: Simulation Results: The cTS (first column) shows strong limited angle artifacts. After the subtraction of the predicted correction map, the artifacts are reduced in the preferred plane (first, third and fifth row) as well as in the orthogonal plane (second, forth and sixth row). Image quality and contrast are enhanced in the DAC corrected scans (second to fifth column). A simulated CBCT (sixth column), which requires a higher dose, has a similar image quality. We compare all reconstructions to the digital phantoms (last column).



(a) XCAT **(b)** Catheter **(c)** Simplices



(d) XCAT - Objects **(e)** XCAT - Background Artifacts

Figure 4.4: Comparison of simulated images and DAC corrections to digital phantoms.

While all U-Nets removed the majority of the artifacts in the periphery of the porcine shin, none of them managed to completely eliminate the spotlights which resulted at the openings of the hollow bone tube. All three 2.5D-networks correctly detected the outer tissue, but had difficulties directly next to the bone and the central region. As in the previous experiments each DAC was able to restore edges while mitigating artifacts. However, DAC-3D had less problems in the central region, but more difficulties with some limited angle artifacts. TV on the other hand smoothed artifacts and low contrast features alike. It did enhance existing high contrast edges, but was not able to restore them.

The porcine rib slab correction showed similar results. All networks were able to detect the high-contrast ribs and outer regions of the soft tissue, even though this phantom was more challenging due to its more complex structure and the higher magnitude of the artifacts. The restoration of the soft tissue in between the ribs was difficult for all networks with DAC-3D showing the best performance. In contrast to the porcine shin, the artifacts could not be fully suppressed in the periphery (see YZ-plane).

The RMSE in comparison to the CBCT-based ground truth is visualized in Figure 4.6. Accordingly to the visual inspection of the porcine shin, DAC-3D achieved the lowest background artifact error, while DAC-C5 had the best result for the object. As the used image mainly consists of background, the lowest error across the whole image (148.34 HU) was achieved by DAC-3D. In comparison to the original cTS the DAC-3D correction reduced the error by 45.18%. The TV implementation did not reduce the numerical error. For the porcine rib slab, the RMSE of the whole image and the artifacts was also reduced. The lowest RMSE for DAC was achieved by DAC-3D with 746.40 HU. However, the RMSE of the objects was increased as the network miscorrected the soft tissue to -1000 HU. The networks successfully mitigated limited angle artifacts. DAC-3D achieved the highest error reduction by 26.4%. However, the reconstruction error (left lower corner in the YZ plane) proved to be difficult for the networks and led to an overall higher image-wide error than TV. TV strongly overestimated the object HUs and similarly to the porcine shin amplifies some of the limited angle artifacts. TV handled the recosntruction error better and resulted in a lower image error than the DACs with 541.28 HU.

4.4 Discussion

We implemented four DAC-versions and evaluated each network on simulated cTS scans in three experiments: Anthropomorphic Simulations, Catheter Simulations and Simplices Simulations. In a fourth experiment the networks were applied to real cTS scans.

Across the simulation experiments, all networks were able to mitigate the limited angle artifacts. On the Anthropomorphic Simulations, which were most similar to the training data, the maximal RMSE reduction was 57%. On the Catheter phantom, which differed more from the training data, the maximal error reduction was 49%. Even on the Simplices Simulations which deviated the most from the training data, the error was still reduced by 35%. These results are within our range of expectations.

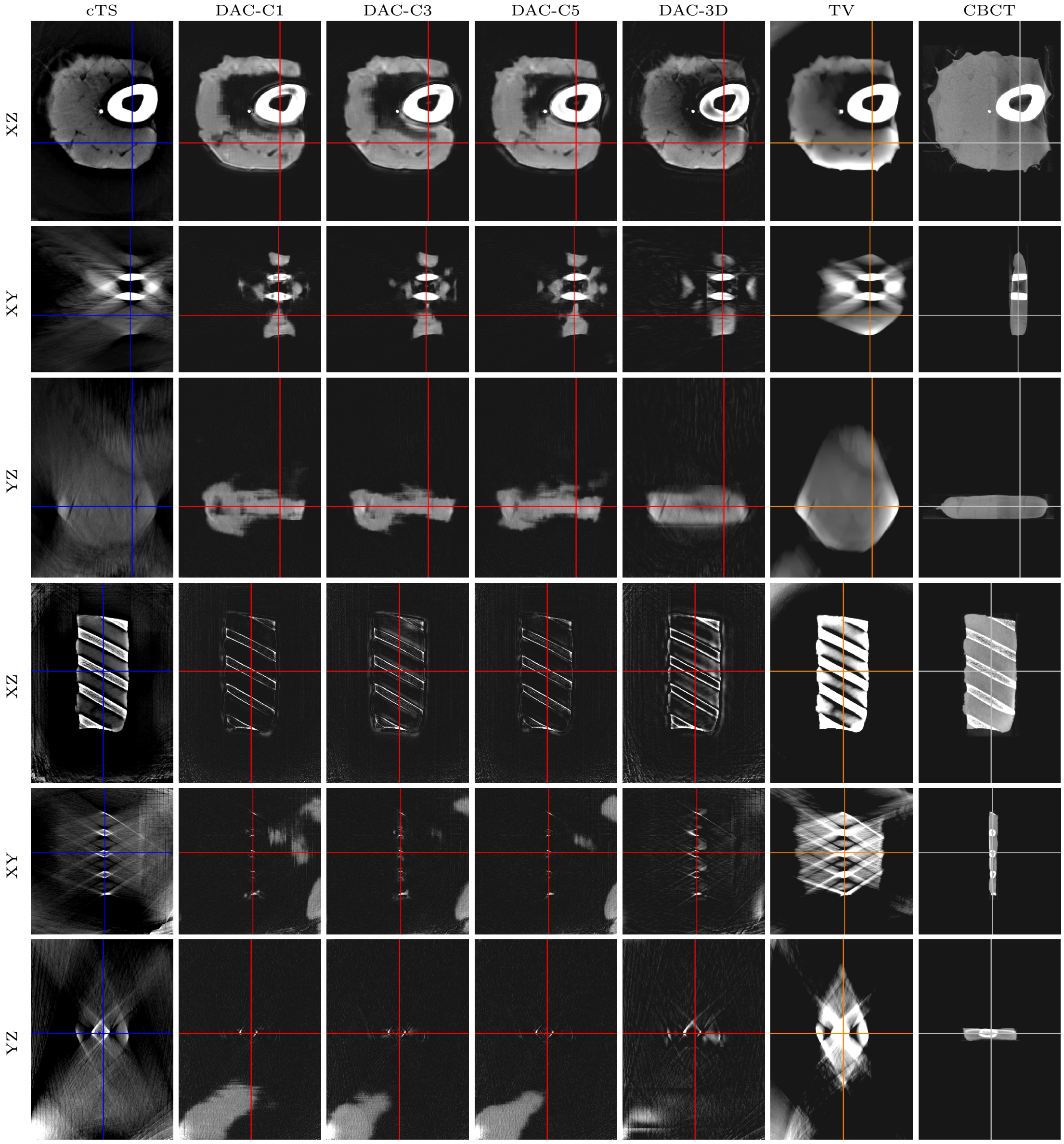
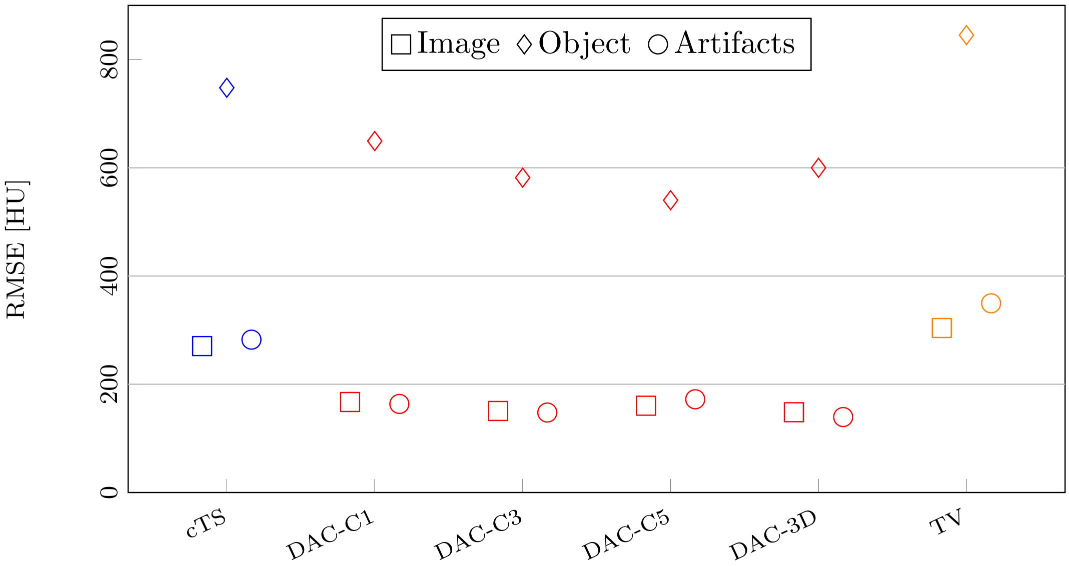
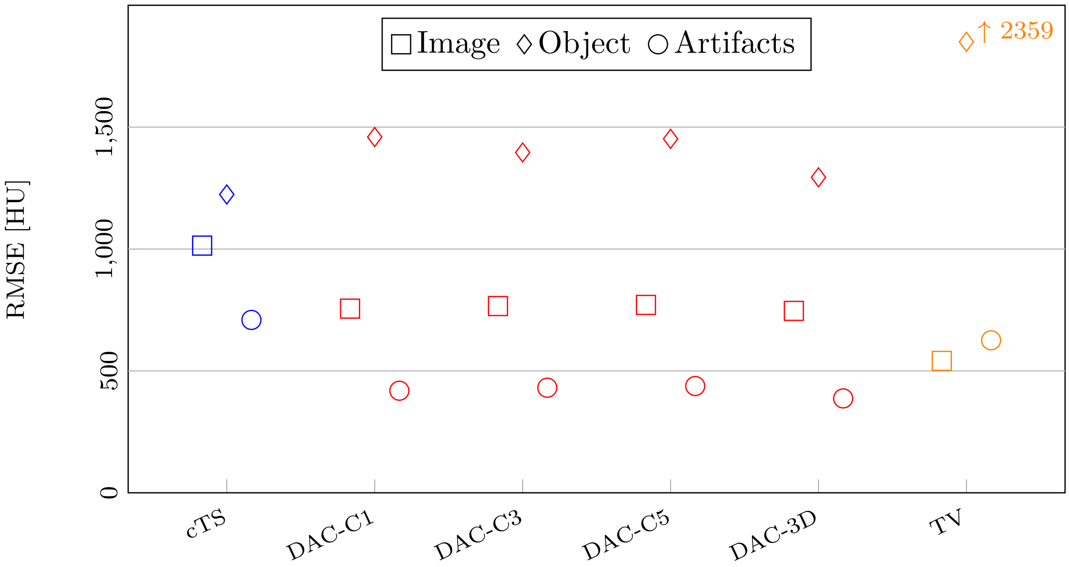


Figure 4.5: Real Data Results: The cTS (first column) shows strong limited angle artifacts in the XY and YZ plane. Using the DAC corrections the artifacts are strongly reduced and edges restored (second to fourth column), while TV smooths object features and artifacts equally (fifth column). A high-dose CBCT-based groundtruth (sixt column) is shown for comparison.



(a) Shin



(b) Rib Slab

Figure 4.6: Comparison of real cTS of the porcine phantom, DAC- and TV-corrections to a high-dose CBCT based ground truth.

The U-Nets and the 3D-ResNet correction maps had different properties. Due to the skip connections the U-Nets were able to estimate more detailed and localized mitigations, which resulted in a better performance in the background. However, as seen especially in the first experiment the three-dimensional operations of DAC-3D led to a better correction of the objects. The 3D-ResNet was also more robust against unknown structures like the catheters.

The resulting RMSE in the background was generally smaller than in the objects. This seems reasonable as for the background the networks only had to learn the correction to -1000 HU, while in the objects the correction is dependent on the tissue and therefore more complex.

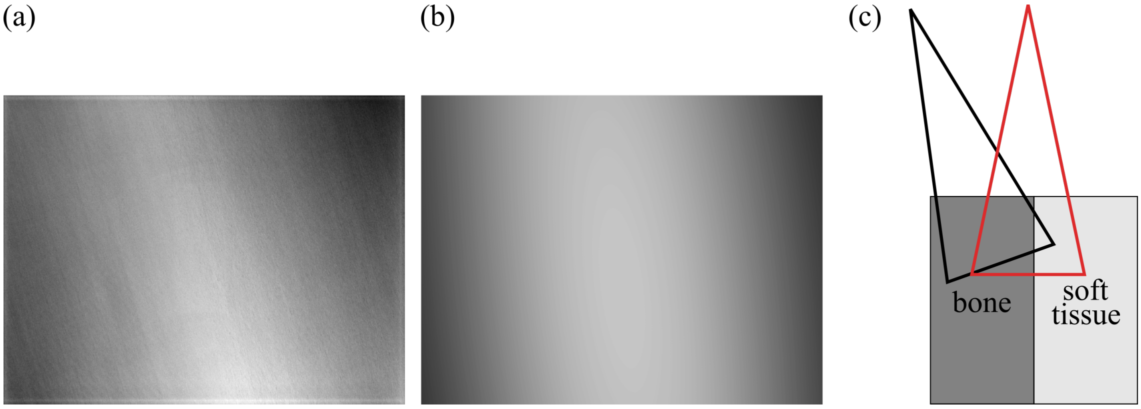


Figure 4.7: HU-Artifacts. a) Shows the anisotropical illumination of the detector due to the heel-effect and b) our gradient mask to mimic the heel-effects in the simulations. c) At locations with HU-discontinuities, a perpendicular view (marked in red) is needed to sharply reconstruct the edge. In cTS we only have the black projections. Therefore, the high-contrast objects have void areas nearby.

In the final experiment, one of the main difficulties of the application to real data we observed is the high dependency of the networks on the accuracy of the HUs, which is not given in cTS. For example, a systematic HU-deviation is caused by the processing of the individual projections. For the reconstructions of the real data sets, we have performed a mean value I_0 log-normalization on each of the radiographies. Due to the heel effect, the detector is not isotropically illuminated as Figure 4.7a displays. Therefore, in certain areas the HUs are either systematically under- or overestimated. Although, we took the heel effect into account in our simulations (Figure 4.7b), our implementation of the heel effect seems to be still too crude and needs refinement.

In particular challenging for DAC was the soft tissue of the porcine data sets. For example, the central region of the shin slice could not be fully restored. This seems to be due to the void area around the bony structures. These stem from reconstruction singularities, which were not present to this extend in the simulations. Singularities can appear at positions with discontinuities in tissue and HUs. A sharp edge can only be reconstructed if an X-ray projection perpendicular to the discontinuity is available (Figure 4.7c). Otherwise, like in our case, the high-contrast objects cast a shadow and the edge is corrupted. As seen before in the simulation data, the 3D-

ResNet performed more robustly in these scenarios, but it also had more problems mitigating the peripheral artifacts.

Another problem with the porcine shin slice is that the tube-like bone produces an interference of artifacts. Every point of the high-contrast bone generates a cone-shaped limited angle artifact. All the artifacts accumulate above and beneath the bone, thereby creating a bright signal. Such tube-like high-contrast objects and the resulting spotlight artifacts were not included in the simulation data. Nevertheless, our networks managed to dim the spotlight-like artifacts, but failed to remove them completely. As deep learning is dependent on the variability of training data, we will further extend the data set.

The real data test also showed that the performance of the networks is dependent on the scanned objects. The artifact estimation of the porcine ribs proved to be more challenging for all implemented networks than the porcine shin slice. This is because firstly the multiple high-contrast ribs cause more extensive artifacts interference pattern than the porcine shin slice and secondly the soft tissue singularities are also more severe. Nevertheless, the networks managed to correctly detect the edges of the ribs. With the beforehand proposed improvements of the training data simulations, the performance should be enhanced.

In general, comparisons of CBCT with diagnostic cTS (gold standard), have shown that the HUs of CBCTs differ significantly from the real values [75, 76]. The effect also translates to cTS. However, due to phrasing the artifact correction as DAC, our solution also intrinsically corrects the HUs.

To combat the remaining limitations the simulation models have to be further refined. For example, our simulations did not include any kind of noise or scatter because we focused on the large-scale limited angle artifacts. In the future, these error sources have to be considered and implemented.

4.5 Conclusion

In this work, we presented a novel simulation-based DAC correction of the three-dimensional limited angle artifacts from cTS reconstruction. Several networks have been trained on simulated imaging data to predict correction maps. The performances of the networks have been evaluated and compared with data sets that were not included in the training sets. A proof of principle on real data has provided with the scan of two porcine phantoms.

The application to 60 simulated test data cases in the first experiment proved that FCNNs can successfully learn to mitigate the limited angle artifacts. Using three-dimensional instead of two-dimensional data input increases the network's ability to remove artifacts. While, numerically DAC-C3 achieved the lowest error, DAC-3D proved more robust. All networks reduced the RMSE to a level similar to the simulated undersampled CBCTs.

As we trained our networks on anthropomorphic images, we investigated how the correction performs on images which strongly differ from the training data. As cTS is used in an interventional context, the presence of non-anthropomorphic structures such as catheters or surgical instruments in the image has to be considered.

Therefore, the features learned for the artifact correction should be as general as possible and not only fitted to anthropomorphic shapes. The network should also not generate artificial structures in case of unknown input. As seen in the second and third experiment, all four networks were able to correct images with arbitrary geometric objects and anthropomorphic shapes with simulated catheters. The U-Nets performed better for artifact mitigation. However, due to the three-dimensional operations the ResNet was more robust in regions with unknown structures.

In the fourth experiment with real acquired data, simulation-based DAC proved to be able to restore high contrast edges such as the body-air-barrier while simultaneously removing artifacts as previously seen in the simulation experiments. Again, DAC-3D proved more robust. Our implementation of an isotropic TV approach on the other hand was not suited as it smooths low contrast edges, while it only removes a fraction of the artifacts. The successful proof of principle of our networks on real data shows that features learned on simulation data can be directly applied to artifact correction on real cTS scans.

Using imaging simulations further eradicates the dependence on large imaging studies to gather a large cohort of patient scans. Highly variational images can automatically be generated and are not subjected to data protection regulations. However, the current version of the correction does not yet yield satisfactory results across the whole image domain.

In conclusion, across all four experiments DAC-correction was able to strongly enhance the image quality. In particular, the networks were able to sufficiently correct the background despite our simple assumptions in the simulations. Based on our experiments we recommend the 3D-ResNet architecture. With regard to the motivation of interventional tomosynthesis to locate critical objects, the reconstructions are mostly sparse. Therefore, most of the artifacts are apparent in the background which our networks can correct efficiently. The presented method has the potential to be used as a post-processing step or included as a regularization term in the iterative reconstruction of cTS.

5. ”Deep Voxel-Guided Morphometry (VGM): Learning regional brain changes in serial MRI”, *MLCN-LNCS, doi: 10.1007/978-3-030-66843-3_16*

5.1 Introduction

MRI has become a fundamental part of the routine clinical management of individual patients with Multiple Sclerosis (MS) [77]. Analysis of subtle changes between examinations is important for the assessment of disease activity and development over time. This includes the analysis of white matter lesions, Cerebrospinal Fluid (CSF)-compartment enlargement, but also of grey matter atrophy [78].

While the progression of MS can be monitored via segmentation of lesions [79], this approach neglects the impact on the surrounding tissue and requires (manual) lesion segmentation. The automatic computation of complete maps which quantify the structural change of the brain tissue offers clinicians more detailed information concerning global and regional morphological changes including appearance of new lesions and information about lesion activity. Change analysis generally requires the images to be registered to the same geometry. They can then be analyzed e.g. via subtraction [80], feature comparison [81], or a high-dimensional deformation field [82], also called Voxel-Guided Morphometry (VGM). The latter allows for the analysis of large spatial deformations, simultaneously achieving sub-voxel accuracy.

DL approaches have become the state of the art solution in many medical image processing tasks. They have been applied to MRI acquisition, segmentation and disease prediction [57]. CNNs have been successfully used to predict the progression of multiple sclerosis [83] and to detect new and enlarging lesions in longitudinal brain MRIs [84].

We present a novel CNN approach combined with an optimized loss function called Deep VGM to analyze brain changes between MRI examinations. We investigate whether Deep VGM can sufficiently approximate VGM for fast clinical usage.

5.2 Materials and Methods

5.2.1 Image Data

In this retrospective study 71 patients with MS were included according to the 2010 diagnostic criteria [85]. The study was approved by the local ethics committee. Each patient underwent two MRI exams, one baseline imaging and a follow-up after 12 months. Imaging was performed using a 3T scanner (Magnetom Skyra, Siemens Healthineers, Erlangen, Germany) and a 3D T1-weighted Magnetization Prepared Rapid Gradient Echo (MPRAGE) sequence with parameters TE = 2.49 ms, TR = 1900 ms, TI = 900 ms, field-of-view 240 × 240 mm^2, and spatial resolution = $0.94 \times 0.94 \times 2.00$ mm^3. The data set includes a total number of 444 lesions with an average number of 6.25 lesions per subject.

5.2.2 Voxel-Guided Morphometry

VGM is a 3D-image alignment method generating maps which show global and regional brain changes between two 3D-MRI data sets from different time points. Usually, T_1-weighted data sets are used for alignment, as these have the most accurate morphological resolution. The algorithm needs high quality brain masks as a prerequisite, which can be obtained using the freesurfer software package [86]. It then proceeds with the following four steps: (i) an affine transformation is determined, which maximizes the overlap of the brain masks (coarse linear alignment). (ii) An inhomogeneity correction to eliminate low frequency bias is performed by comparison of the coarsely aligned images [87]. (iii) A cross-correlation-based technique is then applied to the bias-corrected images for fine linear alignment. (iv) Finally, the applied high-dimensional multiresolution full multigrid method determines the nonlinear deformations, thereby achieving a complete exploitation of information and effective processing [82]. Typical computation times on a recent CPU are 4 minutes for step (i)-(iii) and 7 minutes for step (iv).

The method determines a grey-value-guided movement of each voxel from source to target. In the final step volume alterations for each voxel are extracted from the high-dimensional deformation field. The final output is a map with a quantified value for each voxel, which indicates how much this area increased or decreased in volume. An example case consisting of baseline image, follow-up image and VGM map is given in Figure 5.1. While VGM has initially been applied to stroke data [82, 88], its benefit to MS has recently been shown [78, 89]. Its clinical application is currently hindered by the high computation time of 11 minutes per case.

5.2.3 Preprocessing

VGM maps were computed for all 71 patients. We truncated the VGM maps at $[-5, 5]$ and set values in $[-0.01, 0.01]$ to zero. For Deep VGM the input images were skull-stripped, bias-corrected, and rigidly registered as they were for VGM. The intensities were normalized to the interval $[-1, 1]$. We performed 5-fold-cross-validation. Each fold uses 55 training cases, 2 validation cases and 14 test cases.

5.2.4 Deep VGM: Architecture

We developed a residual architecture based on a 3D U-Net [24, 26] for Deep VGM. The network has four en- and decoding blocks and an additional convolutional block in between. Each convolutional block is extended by residual connections. Encoding is performed via max-pooling, while decoding is performed using a deconvolution. The encoder and the decoder are connected via skip connections, which pass intermediate features from the encoder to be concatenated with the decoding features.

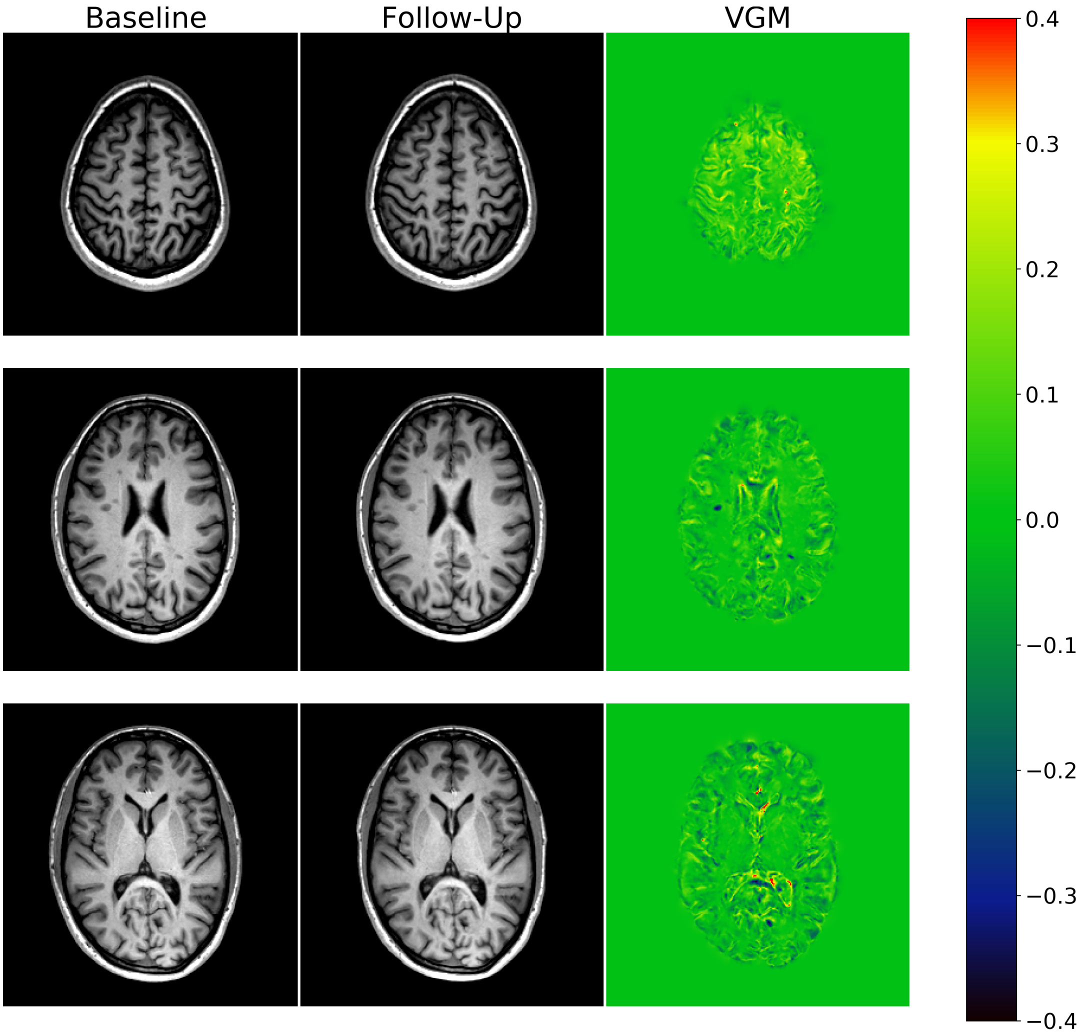


Figure 5.1: Example Data - Three slices from a patient scan. Left column: baseline image; middle column: follow-up image; right column: according VGM map. Two MS lesions which decreased in volume are visible in the middle slice.

The two upper blocks of both sides consist of two convolutions, while the lower blocks consist of three. The network receives the baseline and the follow-up image as a two channel input. The top level blocks have eight channels, which are doubled with every encoding block, until the lowest block has 128 channels. The final output is produced by a $1 \times 1 \times 1$ convolution. All other convolutions use $3 \times 3 \times 3$ kernels. We apply zero-padding, therefore, the size of the network output is equal to the input size. The Deep VGM architecture is depicted in Figure 5.2.

5.2.5 Deep VGM: Training and Loss Functions

We compare different loss functions $\mathcal{L}$ between the predictions $\hat{Y}$ and the labels Y. The number of voxels in a batch is denoted by N, while the predictions and labels of the individual voxels are denoted by $\hat{y}_i$ and y_i respectively. The voxel index in the image is denoted as i. We initially train the networks using the averaged voxel-wise

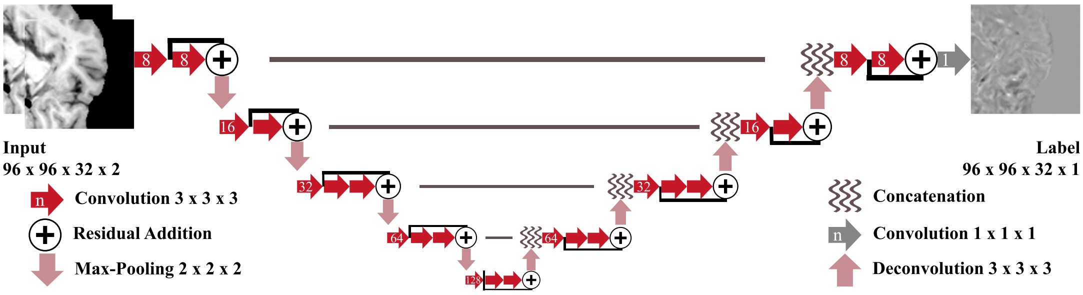


Figure 5.2: Scheme of the Deep VGM architecture. The network is based on a 3D U-Net. All operations are given in the legend. The number of resulting channels is marked by the number on each operation symbol.

Mean Squared Error (MSE) (Equation 5.1) and the Mean Absolute Error (MAE) (Equation 5.2).

$$\mathcal{L}_{MSE}\left(Y,\hat{Y}\right)=\frac{1}{N}\sum_{i=1}^{N}(\hat{y}-y)^2 \tag{5.1}$$

$$\mathcal{L}_{MAE}\left(Y,\hat{Y}\right)=\frac{1}{N}\sum_{i=1}^{N}|\hat{y}-y| \tag{5.2}$$

Recent studies in image regression have shown that more sophisticated loss function can significantly improve the predictions results [90]. Therefore, we additionally test weighted error loss functions (Equation 5.3), which put more emphasis on the high value regions. High change areas (VGM > 0.5) are weighted with one while low change areas are weighted with a step function in the interval $[0.4, 0.1]$.

$$\mathcal{L}_{\text{WME}}\left(Y,\hat{Y}\right)=\frac{1}{N}\sum_{i=1}^{N}w_i\cdot E\left(\hat{y}_i,y_i\right) \tag{5.3}$$

We also combine the error losses with a 3D gradient loss (see Equation 5.4).

$$\begin{aligned}\mathcal{L}_{ME+Grad}\left(Y,\hat{Y}\right)=&\frac{1}{N}\sum_{i=1}^{N}E\left(\hat{y}_i,y_i\right)\\&+\lambda\cdot\sum_{i,j,k}\left(|y_{i,j,k}-y_{i-1,j,k}|-|\hat{y}_{i,j,k}-\hat{y}_{i-1,j,k}|\right)^2\\&\qquad+\left(|y_{i,j,k}-y_{i,j-1,k}|-|\hat{y}_{i,j,k}-\hat{y}_{i,j-1,k}|\right)^2\\&\qquad+\left(|y_{i,j,k}-y_{i,j,k-1}|-|\hat{y}_{i,j,k}-\hat{y}_{i,j,k-1}|\right)^2\end{aligned} \tag{5.4}$$

For training we apply L2 regularization on the network weights. The regularization term is weighted with 10^{-7}. Training is performed using the Adam optimizer with a learning rate of $\text{lr} = 10^{-3}$. We use $96 \times 96 \times 32$ voxel patches for training with a batch size of eight. Training patch selection is distributed slice-wise along the cranio-caudal axis. Patches are sampled randomly with the constraint that patch centers have to be inside the brain mask. Each network is trained for 200 epochs. The number of epochs is based on an initial set of experiments.

5.2.6 Evaluation

To evaluate the similarity between the original VGM maps and the Deep VGM maps we employ three metrics. Firstly, the Structural Similarity Index Measure (SSIM) [91] which expresses a combined similarity value for luminance, contrast, and structure between two images, is used. Secondly, we compute the MAE for all voxels inside the brain mask. Lastly, we test whether the non-change regions inside the brain mask are the same for VGM and Deep VGM. This is verified by computing the overlap of $|\text{VGM}| < 0.01$ and $|\text{Deep VGM}| < 0.01$ using the Dice Similarity Coefficient (DSC) [92, 93].

For 12 patients, the maps from the best performing network were additionally compared to the original VGM maps by an experienced neurologist. For the quantitative evaluation individual lesions were first identified on the T_1-weighted images. Based on the VGM maps each lesion was then classified by the neurologist as "chronic-active" (either chronic enlarging or chronic shrinking) or "chronic-stable". This process was repeated using the Deep VGM maps and the results were compared.

5.2.7 Implementation

All networks were implemented using TensorFlow 2.0 and Python 3.5. For the evaluation and image processing we used SimpleITK. Training and testing were performed on an NVIDIA GTX 1050 graphics card. The training time for each network was 26 hours.

5.3 Results

5.3.1 Quantitative Evaluation

The prediction of each Deep VGM map took 2.5 seconds. This equals a reduction of 99.62% in comparison to VGM. Combined with the computation time for the preprocessing, Deep VGM takes 4.04 minutes to compute one map. Evaluation results for the three metrics are listed in Table 5.1. Of the two simple error loss functions, the $\mathcal{L}_{MAE}$ loss performed better for all evaluation metrics (SSIM, MAE, and DSC) than the $\mathcal{L}_{MSE}$ loss.

Weighting the loss functions decreased the quality of the predicted maps. Adding the gradient loss function improved the predictions only in combination with $\mathcal{L}_{MAE}$. Here, SSIM and MAE showed a significant improvement, i.e. increase in the values for SSIM and decrease in MAE, respectively. The DSC did increase, but not significantly. Predictions from all networks and difference maps are shown in Figure 5.3.

5.3.2 Qualitative Evaluation

Based on the aforementioned results, we selected the network trained with the $\mathcal{L}_{MAE+Grad}$ loss function for Deep VGM. 94 chronic MS lesions (69 stable; 17 shrinking; 8 enlarging) in the test images were examined. Using the Deep VGM maps all 17 shrinking lesions were correctly identified. Only one enlarging lesion was not detected. Out of the 69 stable lesions Deep VGM falsely showed two lesions as "chronic-active" (one enlarging, one shrinking). Deep VGM therefore, has a 3% lesion error rate in reference to the original VGM maps. Figure 5.4 shows two correct cases and the two falsely classified stable lesions.

Table 5.1: Quantitative Evaluation - best value for each metric is marked bold. Significantly better values ($p < 0.01$, paired t-test) are additionally underlined. High values for SSIM and DSC represent better results (depicted by up arrow) and low values for MAE (down arrow), respectively.

	SSIM ↑	MAE ↓	DSC ↑
$\mathcal{L}_{MAE}$	0.9504 ± 0.0242	0.0385 ± 0.0120	0.9806 ± 0.0033
$\mathcal{L}_{MSE}$	0.9452 ± 0.0265	0.0417 ± 0.0131	0.9799 ± 0.0031
$\mathcal{L}_{WMAE}$	0.9425 ± 0.0273	0.0438 ± 0.0131	0.9802 ± 0.0038
$\mathcal{L}_{WMSE}$	0.9425 ± 0.0332	0.0500 ± 0.0142	0.9788 ± 0.0037
$\mathcal{L}_{MAE+Grad}$	**0.9521 ± 0.0236**	**0.0377 ± 0.0116**	**0.9807 ± 0.0034**
$\mathcal{L}_{MSE+Grad}$	0.9450 ± 0.0265	0.0412 ± 0.0123	0.9800 ± 0.0033

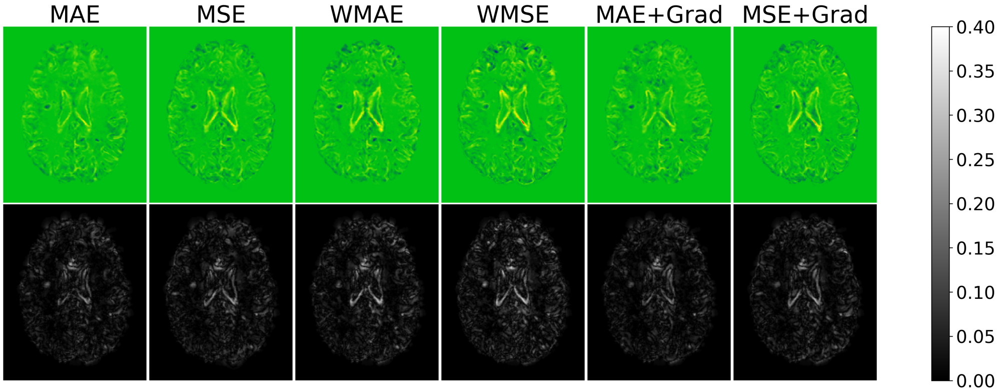


Figure 5.3: Results - Top row shows the predicted Deep VGM maps from the networks trained with different loss functions. Bottom row shows the respective absolute difference maps in comparison to the original VGM.

5.4 Discussion

Predicting VGM maps by our convolutional neural network combined with the $\mathcal{L}_{MAE+Grad}$ loss function (Deep VGM) is feasible. It produces VGM maps similar to the original approach. As depicted in Figure 5.3 high errors only occur in the brain periphery, close to the borders of the brain mask. In these regions registration errors have the highest influence. We also see high differences in some cases for the outer rim of the lateral ventricle. For white matter, Deep VGM maps are estimated with high accuracy.

However, there is a slight systematic under estimation of exceptionally high VGM values. Since the majority of cases in our study don't have VGM values > 3 and thus, these values are rare in the training data and therefore, the networks do not predict them in the test cases. Weighting the high change values stronger in the loss functions did reduce this problem, but lead to a systematic overestimation instead. Another weighting scheme could potentially achieve better results, those investigated here did not improve the result. However, the addition of the gradient loss significantly improved the overall performance and seems to be a better solution to this problem.

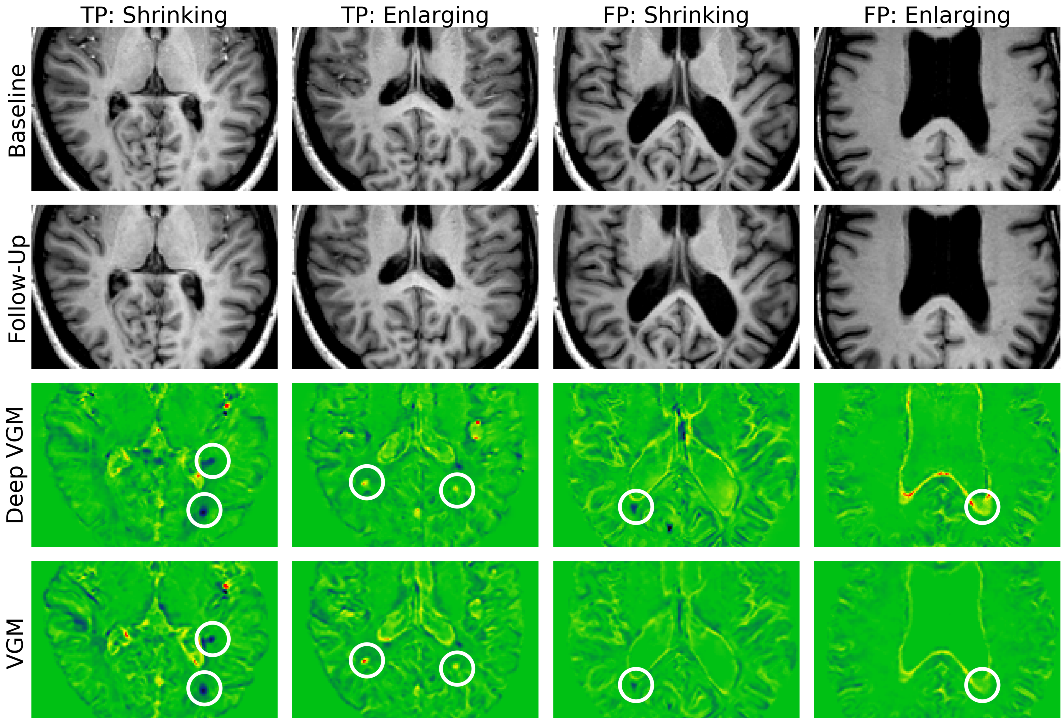


Figure 5.4: Results - Examples of examined lesions with respective T1w baseline MRI and follow-up, VGM and Deep VGM map. Each lesion is marked with a white circle. First column shows TP shrinking lesions which have been correctly identified by Deep VGM. The second column shows TP enlarging lesions. The third and fourth column show the two FP lesions.

Our initial experiments only included data from MS patients. In future, we plan to include more diverse training data to be able to apply Deep VGM to other brain diseases such as stroke.

5.5 Conclusion

Our novel Deep VGM can produce quantified VGM maps at high quality while saving computational time compared to the original VGM approach. This opens the possibility to further translate this approach towards clinical routine. To facilitate the use of VGM analysis for clinical users, Deep VGM was already integrated into the clinical image analysis software mTRIAL [94], an integrated solution for use in clinical trials and diagnostics. Eventually, automated, objective and therefore, personalized diagnostics of disease evolution in patients with MS might become possible.

6. "Convolutional Neural Network Ensemble Segmentation with Ratio-based Sampling for the Arteries and Veins in Abdominal CT Scans", *IEEE T-BME, doi: 10.1109/TBME.2020.3042640*

6.1 Introduction

The exact position and structure of veins and arteries is an essential piece of information for diagnostics and therapy planning. Information about blood vessels is especially relevant in patient-specific surgical planning, intra-operative navigation and minimal-invasive interventional approaches. Furthermore, blood vessel information is used to identify organ positions and to analyze and characterize cancer metastases to guide clinical decision-making. Abdominal blood vessels have large inter-patient variations in branching patterns, branch positions, and branch lengths. Incorrect or incomplete information about the vascular anatomy might lead to damage during surgery.

The viability of tissue can only be guaranteed if the supply of oxygen-rich blood via the arteries and the return of the oxygen-depleted blood via the veins is ensured. This makes the knowledge of arterial and venous vessel courses essential for the planning of a safe living donor organ transplantation. Different anatomical variants of the graft such as kidneys with a single artery and vein versus those with multiple arteries and/or veins require different handling [95]. Analysis of the hepatic vasculature is vital to see whether a donor liver is suitable for transplantation [96]. Furthermore, consideration of vascular structures and their anatomical relationships with the liver segments is crucial for a comprehensive liver resection surgery plan [97]. The risk of hepatic arterial injury during Transjugular Intrahepatic Portosystemic Shunt (TIPS) placement can also be reduced by careful analysis of the arterial and venous tree prior to the procedure [98].

6.1.1 Related Work

A myriad of dedicated blood vessel segmentation algorithms for different body regions and modalities have been developed [99]. With very few exceptions [100], most of them are tailored to a respective body region. Segmentation is often realized via vessel enhancement, which improves vessel perception by Wavelet filtering [101] or by the computation of the local vesselness measure based on the eigenvalues of the Hessian matrix [102]. Other prominent segmentation approaches employ deformable models [103] or vessel tracking [104]. As in many image processing fields, substantial effort has been made to apply CNNs [97, 105]. One major application is the segmentation of retinal blood vessels in two-dimensional (2D) color images acquired with Three Charge Coupled Devices (3-CCD) cameras [106]. Here, the combination of inception models and a FCNN yields state-of-the-art results regarding performance and speed. Tetteh *et al.* proposed a FCNN-based blood vessel segmentation method from 3D Magnetic Resonance Angiography (MRA) volumes [107]. Here, 2D cross-hair filters were utilized to incorporate 3D information whilst reducing the associated computational burden. A similar approach was employed by Kitrungrotsakul *et al.* for the segmentation of hepatic arteries [108]. Recent results also underlined the feasibility of synthetic training data to improve overall segmentation performance of CNNs in the case of brain and abdominal arteries [107, 109]. The inherent class imbalance issue of artery segmentation in CT images has recently been addressed by Oda *et al.* by variable density sampling of training patches, which resulted in higher prediction accuracy for small vessels [110]. The separation of arteries and veins is usually performed in an additional processing step after vessel segmentation [111][112]. To the best of our knowledge, no one has performed simultaneous segmentation and separation in Contrast Enhanced Computed Tomography (CE-CT) images using a single processing step.

As shown in Table 6.1, 2D as well as 3D CNNs are used in current state of the art vessel segmentation methods. Because CT images are 3D data, the usage of 3D CNNs intuitively seems to be the appropriate choice. However, their performance is restricted due to the finite amount of available GPU memory which causes a limited field of view during training [30]. While 2D networks are not as memory expensive and can therefore be trained on large patches or entire image slices, they fail to fully explore inter-slice information [113]. CNN performance can further be enhanced by the combination of individual networks into ensembles. The basic principle behind ensemble learning is that, by combining a series of weaker base learners a stronger ensemble can be constructed. To achieve high ensemble performance the individual components should be as diverse and simultaneously as accurate as possible [114]. Predictions of individual networks can either be averaged or more complex schemes like bagging or boosting can be applied. While boosting weights the base learners based on their accuracy, bagging reduces variance by training the same base learner on several subsets of the available data. A common approach is the averaging of multiple networks instances trained with different initializations. In acute ischemic lesion segmentation, averaging has been shown to improve performance by 4.18% [115]. This solution is able to correct irregular errors, but suffers from an architecture-bias. Ensembles of heterogeneous networks are more robust and generalize better [116].

6.1.2 Contribution

We present a novel approach to simultaneously segment the abdominal vessel trees (arteries, veins) combining well recognised CNN architectures into a powerful ensemble that allows for high segmentation accuracy and structural integrity, equipped with an optimized sampling scheme and loss function. We identify which CNN architectures achieve the best results for segmentation and separation of the abdominal

Table 6.1: Related work for vessel segmentation using CNNs

Reference	Modality	Anatomy	Architecture	DSC
Tetteh 2017 [106]	3-CCD	Retinal Vessels	Inception + FCNN (2D)	0.85
Chen 2017 [117]	MRA	Cerebral Arteries	Y-Net (3D)	0.83
Tetteh 2018 [107]	MRA	Cerebral Arteries	FCNN (3D)	0.86
Kitrungrotsakul 2018 [108]	CT	Hepatic Arteries	FCNN (3D)	0.88
Russ 2019 [109]	CT	Abdominal Arteries	U-Net (2.5D)	0.83
Oda 2019 [110]	CT	Abdominal Arteries	FCNN (2.5D)	0.87
Decathlon Challenge 2018 Task08 [30, 118]	CT	Hepatic Veins	U-Net (-)	0.63
Yu 2019 [119]	CT	Hepatic Veins	U-Net (3D)	0.72

vascular systems in CE-CT. Simultaneous segmentation is compared to individual segmentation. We present a structured sampling approach for segmentation tasks that suffer from high class imbalance. Well performing architectures are combined into a heterogeneous ensemble. We additionally compare different weighting schemes. Lastly, we investigate how performance translates from one data set to another.

6.2 Materials and Methods

6.2.1 Data

For this work we use two publicly available data sets. Firstly, we use the 3D-IRCADb 01 (IRCAD) [120]. It is composed of abdominal CE-CT scans of ten women and ten men. The annotations for several structures such as liver, kidneys, lungs, etc. are provided in the data set. The relevant classes for this work are *portalvein*, *venacava*, *venoussystem* and *artery*. We opted for a single venous class as we aim at a a holistic representation of the venous system and therefore combined the three venous classes into one *vein* class. Not all annotations are available for each IRCAD case. For eight cases the *artery* label is not provided. The image volumes of the IRCAD extend from the diaphragm to the beginning of the pelvic bone. Volumes of two cases (no. 19 and 20) extend further and were thus cropped accordingly. Three cases are especially challenging as they show a large lesion, strong metal artefacts or an aortic aneurysm, respectively.

Secondly, we use the data from the MICCAI Multi-Atlas Labeling Beyond the Cranial Vault Workshop and Challenge (BTCV) [121] in combination with the annotations and cropping coordinates provided by Gibson *et al.* [122]. The relevant classes are *aorta*, *inferior vena cava*, *portal vein* and *splenic vein*. Again, we combined the three venous classes into one *vein* class. We exclude cases 9, 23, 25 and 36 from our study as they show insufficient contrast enhancement.

Both data sets were revised by a medical professional and extended to completion for abdominal arteries and veins.

Table 6.2: Data Set Statistics

Data Set	% Artery Voxels	% Vein Voxels	# Cases	Median Spacing [X/Y, Z]	# Slices [min, max]
IRCAD	0.275	0.631	20	[0.74, 1.60]	[74, 260]
BTCV	0.239	0.515	26	[0.76, 3.00]	[33, 87]

6.2.2 Preprocessing and Sample Mining

As shown in Table 6.2, the vessel voxels account for less than 1% of the volume in our abdominal CE-CT data set which is in line with prior studies [110]. It is possible to counteract this under-representation by weighting the classes differently in the loss function. However, finding the correct weighting scheme is a complicated task, especially for a non-binary segmentation. Alternatively, the sampling pattern can be adapted accordingly [110]. We propose a ratio-based combination of oversampling and undersampling [123]. This ensures a more equal representation of the classes during training and thus combats the strong class imbalance. The patches for the training are sampled dynamically from the images in each epoch. We differentiate between two classes of patches:

1. Background: patch is centered on a background voxel
2. Vessel: patch is centered on a vessel voxel

Patch selection is distributed slice-wise along the cranio-caudal axis. Slices with more annotations are sampled more densely than slices with fewer annotations. As the number of slices per case differs highly (see Table 6.2), we use a fixed number of samples per volume to ensure balance between the training cases. We ensure a 50:50-ratio between background and vessel patches during the sampling process. The same ratio is used in the training batches, ensuring that the network is shown vessel and background samples in each iteration.

The images are resampled to a voxel spacing of $0.75\,\text{mm} \times 0.75\,\text{mm} \times 1.5\,\text{mm}$. The densities are windowed to range between $[-150\,\text{HU}, 275\,\text{HU}]$ and then mapped to the interval $[-1, 1]$. Data augmentation is performed in the form of a rotation with angle α around the cranio-caudal axis with $\alpha \in [-12.6\,^\circ, 12.6\,^\circ]$ to mimic possible patient positions. Additionally, the in-plane resolution is varied by up to 2%. For 2D, each patch is extended by one slice in each direction along the cranio-caudal axis to cover more spatial information. Each sample therefore has three channels. The practice of using additional image slices as color channels, thus giving some 3D information while still using 2D operations is also referred to as 2.5D in literature.

6.2.3 Networks

We selected two auto-encoder architectures and one fully convolutional architecture, that does not perform encoding and decoding, from literature, each implemented in 2D and 3D. The 3D networks have the same basic structure, but all operations are swapped for their 3D counterpart. For all networks we use spatial drop out with a rate of 0.01. All convolutional layers use the ELU activation function [124], except for the final 1×1 ($\times$ 1) convolution, which employs *softmax*.

The **U-Net** was implemented according to Ronneberger *et al.* [24]. The network has four en- and decoding blocks as well as an additional convolutional block in between. Encoding is performed via max-pooling, while decoding is performed using a deconvolution. The encoder and the decoder are connected via skip connections, which pass intermediate features from the encoder to be concatenated with the decoding features. The two upper blocks of both sides consist of two convolutions, while the lower blocks consist of three. The top level blocks have eight channels, which are doubled with every encoding block, until the lowest block has 128 channels. All convolutions use 3×3 ($\times$ 3) kernels. In contrast to the original architecture we apply zero-padding, therefore the size of the network output is equal to the input size. Details of this architecture are shown in Figure 6.1a.

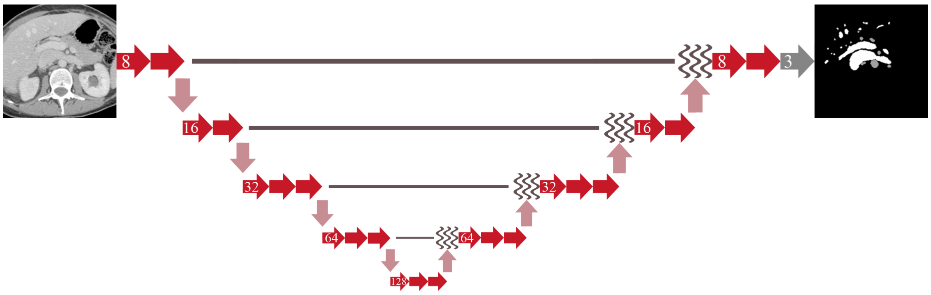
As a second architecture we implemented the **V-Net** by Milletari *et al.* [25]. Similar to the U-Net, the network has four en- and decoding blocks as well as an additional convolutional block in between. Encoder and decoder are also connected via skip connections, but encoding is performed with a strided convolution and decoding with a strided deconvolution. The top-level blocks consist of a single convolutional layer, the second level blocks of two and all remaining blocks have three. Each block contains a residual addition, where the initial feature map is added to the final feature map. The number of channels for each block is equal to those of the U-Net configuration, but all convolutions use 5×5 ($\times$ 5) kernels. Details of this architecture are shown in Figure 6.1b.

The third implemented architecture is the DeepVesselNet (**D-Net**) by Tetteh *et al.*[107]. This architecture employs 3×3 ($\times$ 3) and 5×5 ($\times$ 5) kernels. We use less channels than in the original publication, because the network uses a high amount of GPU memory due to the size of the feature maps. We use full convolutions instead of cross-hair convolutions for the same reason. The network architecture is depicted in Figure 6.1c.

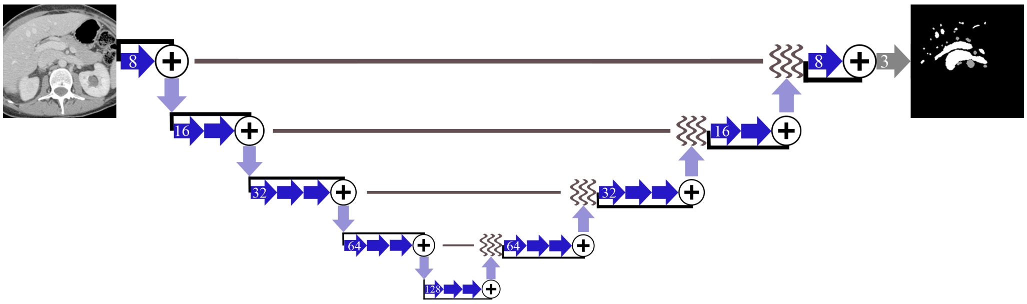
The number of trainable Parameters for the 2D and 3D versions of each network is given in Table 6.3.

Table 6.3: Number of Trainable Parameters

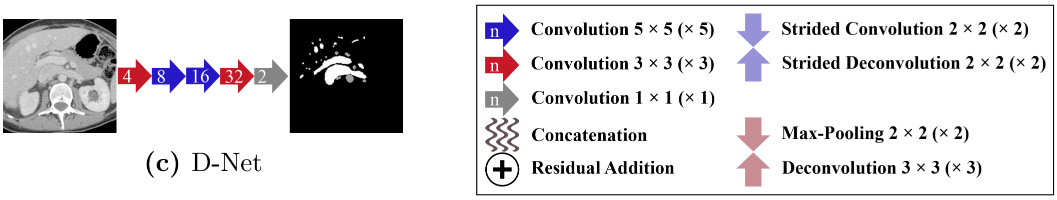
Network	# 2D	# 3D
U-Net	780 168	2 337 672
V-Net	2 250 120	10 983 536
D-Net	8 872	34 088



(a) U-Net



(b) V-Net



(c) D-Net

Figure 6.1: Details of the three architectures used for vessel segmentation. All operations are given in the legend. Kernel sizes are given for 2D with the extra parameter for 3D listed in brackets. The number of channels resulting from operations is marked by the number on the operation symbol.

6.2.4 Training and Loss Function

Training is performed using the Adam optimizer with a learning rate of 10^{-3}. We combine the Dice loss $\mathcal{L}_{DSC}$ (Equation 6.1) and the cross entropy loss $\mathcal{L}_{CE}$ (Equation 6.2) between the predicted probabilities P and the labels Y. The number of voxels in a batch is denoted by N, while the probabilities and labels of the individual voxels are denoted by $p_{i,c}$ and $y_{i,c}$. The voxel index in the image is denoted as i and c is the class index.

$$\mathcal{L}_{DSC}\left(Y,P\right) = 1 - \frac{2 \cdot \sum\limits_{c=1}^{3} \sum\limits_{i=1}^{N} p_{i,c} \cdot y_{i,c}}{\sum\limits_{c=1}^{3} \sum\limits_{i=1}^{N} p_{i,c} + \sum\limits_{c=1}^{3} \sum\limits_{i=1}^{N} y_{i,c}} \tag{6.1}$$

$$\mathcal{L}_{CE}\left(Y,P\right) = - \frac{\sum\limits_{c=1}^{3} \sum\limits_{i=1}^{N} y_{i,c}\, log(p_{i,c})}{3 \cdot N} \tag{6.2}$$

$\mathcal{L}_{CE}$ is weighted with $\lambda_1 = 10$. For the training we apply L2 regularization on the network weights W. The regularization term is weighted with $\lambda_2 = 10^{-7}$. Thus, the complete objective $\mathcal{L}$ is:

$$\begin{aligned} \mathcal{L}\left(Y,P,W\right) = & \mathcal{L}_{DSC}\left(Y,P\right) \\ & + \lambda_1 \, \mathcal{L}_{CE}\left(Y,P\right) \\ & + \lambda_2 \sum_{w \in W} w^2. \end{aligned} \tag{6.3}$$

Each network is trained for at least 30 epochs. After that training stops as soon as the difference of validation accuracy of the vein and artery class over the last twenty epochs each fall under 10^{-4}. We then select from the last ten epochs the weights with the highest average accuracy on the validation data. The samples per volume, patch and batch sizes for 2D and 3D are listed in Table 6.4. The number of training iterations in an epoch is equal for the 2D and the 3D training.

Table 6.4: Sampling Parameters

	Samples per Volume	Batch Size	Patch Dimensions
2D	160	16	256 × 256 × 3
3D	80	8	128 × 128 × 32 × 1

6.2.5 Prediction and Ensembles

For the 2D inference, we mirror padded the image volume by a single slice in each direction of the cranio-caudal axes to be able to predict segmentations for the entire volume. The usage of 3D CNNs requires large memory on the GPU. We were not able to predict the segmentation of one entire CT volume using our current hardware. Therefore, we split the CT volume into sections that would fit on the GPU. Merging the regional probabilities back into one mask, we noticed that segmentation quality decreases further from the center along the cranio-caudal axis of each region. We therefore revised the process and instead perform the inference for overlapping 512 × 512 × 64 voxel regions. When merging the partial probabilities, we apply a weighting scheme based on the distance to the center slice. For the 3D inference the volumes were also mirror padded so that the border voxels would be covered by two overlapping regions. To combine the predictions of networks into an ensemble E we average the resulting probability maps for each class. The probability maps of all networks and ensembles are resampled back to the original data resolution and the *argmax* is then applied to extract the predicted segmentation. We apply no further post-processing.

6.2.6 Evaluation Metrics

We employ four metrics to assess the quality of the predicted segmentation. Two metrics compare the predictions $\hat{Y}$ of the networks to the labels Y, one describes the quality of vessel separation and the last one judges the structural integrity of $\hat{Y}$.

Firstly, we use the DSC to assess the overlap:

$$DSC\left(\hat{Y}, Y\right) = \frac{2\,|\hat{Y} \cap Y|}{|\hat{Y}| + |Y|} \tag{6.4}$$

Secondly, we employ the Average Symmetric Surface Distance (ASSD) which is more sensitive to shape and alignment:

$$ASSD\left(\hat{Y}, Y\right) = \frac{\sum\limits_{\hat{y} \in \hat{Y}} \min d(\hat{y}, Y) + \sum\limits_{y \in Y} \min d(y, \hat{Y})}{|\hat{Y}| + |Y|} \tag{6.5}$$

Thirdly, we use the Vessel Confusion Rate (VCR), to analyze which fraction of voxels of one vessel class c is falsely identified as the other $\bar{c}$:

$$VCR\left(\hat{Y}, Y\right) = \frac{\sum\limits_{i=1}^{N} y_{i,\bar{c}} \cdot \hat{y}_{i,c}}{|Y_c|} \tag{6.6}$$

Lastly, we compute the Connectivity (C), which describes how much of the predicted segmentation is covered by the Largest Connected Component (LCC):

$$C\left(\hat{Y}\right) = \frac{|LCC(\hat{Y})|}{|\hat{Y}|} \tag{6.7}$$

6.3 Experiments and Results

To assess the performance of the CNN segmentation we performed three experiments:

1. We trained and tested six network configurations on the IRCAD data set using 5-fold-cross-validation. For each fold we used 14 cases for training, 2 for validation and 4 for testing. Each configuration is trained once individually for each vessel class and once simultaneously.
2. Well performing networks from the first experiment were combined into an ensemble. We investigated different weighting schemes.
3. Well performing networks were trained once on the entire IRCAD. The trained networks and the ensembles were then applied to the BTCV data set.

The described networks were implemented using TensorFlow 2.0 and Python 3.5. For the evaluation and image processing we used SimpleITK. Training and testing were performed on a Windows Server 2016 with an Intel Core i7-7700K CPU, 64GB RAM and a NVIDIA Titan Xp graphics card.

6.3.1 Experiment 1

The results of the evaluation metrics for the venous and the arterial vessel tree are listed in Table 6.5. The D-Nets consistently resulted in the worst scores for all metrics. The simultaneously trained 2D U-Net and 2D V-Net achieved the highest DSC

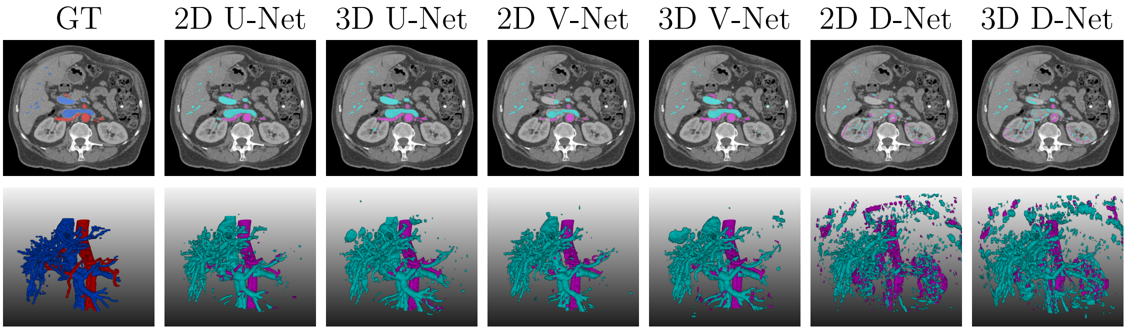


Figure 6.2: Experiment 1 - Visual comparison of segmentation results of six different simultaneously trained networks for case 17. The manual annotations, groundtruth (GT), are shown in the first column. Arteries are shown in red and veins in blue. Top row: one slice from the CE-CT volume with the superimposed masks. Bottom row: 3D visualization of the vessel trees. Predictions are shown in magenta (arteries) and cyan (veins).

and ASSD for both structures. The simultaneously trained 3D V-Net also achieved comparable results for the Arteries. Lowest confusion rates resulted from individual training, but the simultaneously trained 2D U-Net and 3D V-Net performed similarly. Highest connectivity for the veins was achieved by the simultaneously trained V-Net, while the highest values for the arteries resulted from the individually trained 3D V-Net. However, for both structures several auto-encoders from individual and simultaneous training performed similarly. The results of all six networks for one exemplary case are shown in Figure 6.2. Visual inspection of the segmentation showed that the larger abdominal vessels such as the aorta, the inferior vena cava, the portal vein and the hepatic veins were segmented with high precision by the U- and V-Nets. Errors mostly arose for smaller abdominal vessels such as the renal and mesenteric vessels. The D-Nets falsely segmented the kidneys and were barely able to detect the aorta, but they were able to detect the hepatic veins.

6.3.2 Experiment 2

Based on the results from Experiment 1 we selected the 2D U-Net, the 2D V-Net, the 3D U-Net and the 3D V-Net for the ensembles. We compared seven configurations:

1. All networks were weighted equally.
2. Only 2D networks.
3. Only 3D networks.
4. Only U-Nets.
5. Only V-Nets.
6. 2D networks were weighted slightly higher than 3D networks.
7. 2D U-Net, 2D V-Net and 3D V-Net.

The weighting parameters for all ensembles are listed in Table 6.6. The results of the evaluation are listed in Table 6.7. The ensembles performed better than the individual networks for the all metrics, except the venous ASSD, for which they performed on par with the 2D U-Net. The DSC for veins and arteries as well as the

Table 6.5: Experiment 1 - Quantitative comparison of the segmentation quality of six different networks. The best result for each metric and class is marked grey. Results which did not significantly (paired t-test, $p < 0.05$) differ from the best result are marked in light grey.

			DSC ↑	ASSD ↓	VCR ↓	C ↑
Veins Only		2D U-Net	0.728±0.060	2.250±0.913	0.006±0.005	0.838±0.140
		2D V-Net	0.743±0.049	2.096±0.768	0.006±0.006	0.863±0.138
		2D D-Net	0.279±0.108	8.745±4.400	0.026±0.016	0.160±0.084
		3D U-Net	0.719±0.052	2.565±0.943	0.006±0.006	0.800±0.210
		3D V-Net	0.716±0.083	2.456±1.016	0.009±0.007	0.841±0.158
		3D D-Net	0.340±0.094	8.907±3.353	0.027±0.019	0.310±0.134
Simultaneous	Veins	2D U-Net	0.741±0.050	1.966±0.713	0.007±0.006	0.868±0.139
		2D V-Net	0.743±0.049	2.122±0.815	0.008±0.007	0.899±0.077
		2D D-Net	0.254±0.092	8.398±4.634	0.017±0.008	0.133±0.056
		3D U-Net	0.702±0.055	2.519±0.825	0.010±0.009	0.833±0.167
		3D V-Net	0.708±0.083	2.464±0.956	0.008±0.012	0.875±0.126
		3D D-Net	0.329±0.094	9.250±3.455	0.035±0.043	0.019±0.156
	Arteries	2D U-Net	0.827±0.067	3.342±1.578	0.016±0.020	0.932±0.056
		2D V-Net	0.824±0.069	3.094±1.112	0.017±0.011	0.916±0.054
		2D D-Net	0.373±0.135	16.851±7.687	0.101±0.068	0.451±0.163
		3D U-Net	0.803±0.075	4.058±2.044	0.021±0.018	0.937±0.043
		3D V-Net	0.815±0.076	3.367±1.744	0.013±0.009	0.946±0.023
		3D D-Net	0.413±0.146	21.184±11.102	0.099±0.064	0.473±0.138
Arteries Only		2D U-Net	0.822±0.068	3.572±1.294	0.010±0.011	0.936±0.059
		2D V-Net	0.812±0.075	3.666±1.564	0.014±0.019	0.932±0.078
		2D D-Net	0.345±0.133	17.259±8.619	0.080±0.069	0.482±0.175
		3D U-Net	0.793±0.083	5.469±3.705	0.020±0.024	0.929±0.044
		3D V-Net	0.813±0.071	3.557±1.442	0.011±0.009	0.952±0.026
		3D D-Net	0.418±0.160	23.514±11.837	0.112±0.084	0.507±0.166

the arterial VCR and the arterial C of the respective ensembles were significantly better than every single individual network (paired t-test, $p < 0.05$). The best DSC for both classes was achieved by E_6, which weights the 2D networks slightly higher. E_1 performed similar for both structures. The lowest VCR for veins and arteries was achieved by E_1, while E_6 performed similarly.

Table 6.6: Experiment 2 - Ensemble weights.

	2D U-Net	2D V-Net	3D U-Net	3D V-Net
E_1	0.25	0.25	0.25	0.25
E_2	0.50	0.50	0.00	0.00
E_3	0.00	0.00	0.50	0.50
E_4	0.50	0.00	0.50	0.00
E_5	0.00	0.50	0.00	0.50
E_6	0.30	0.30	0.20	0.20
E_7	0.34	0.34	0.00	0.32

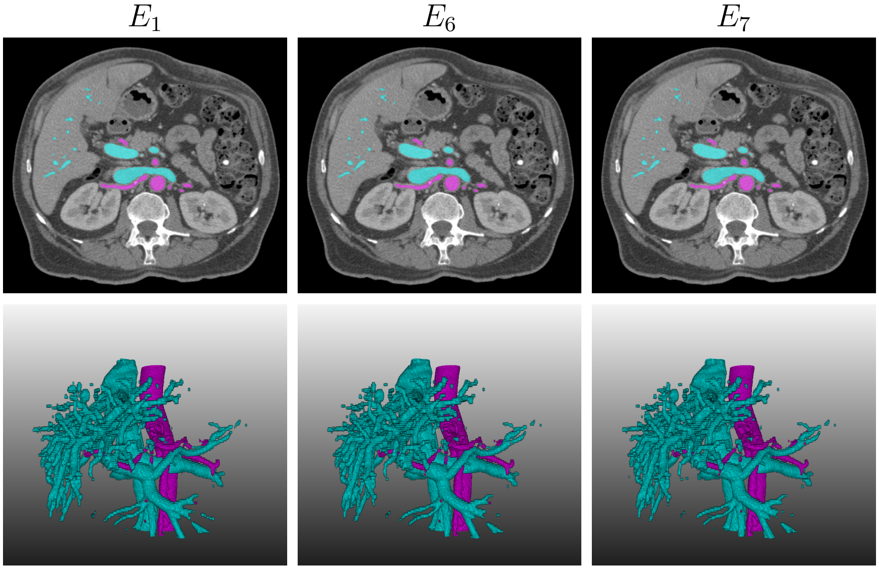


Figure 6.3: Experiment 2 - Visual comparison of segmentation results of 3 different ensembles for case 17.

Comparing Figure 6.2 to Figure 6.3 shows the strong reduction in small false positive fragments from the individual networks to the ensembles. Segmentation of the smaller abdominal vessels was also greatly improved. Figure 6.4 shows the segmentation results of the three more challenging IRCAD cases from the 2D U-Net and E_1. Due to the strong metal artefacts in case 3, none of the networks were able to segment the complete aorta. Case 7 suffers from a large tumor that distorted especially the inferior vena cava. The networks were nonetheless able to segment the deformed vessels. Case 9 suffers from an aortic aneurysm. All networks segmented the inner region of the aneurysm, but were not able to detect that the surrounding region also belongs to the aorta. The networks were also not able to segment the inferior vena cava in the slices of the aneurysm.

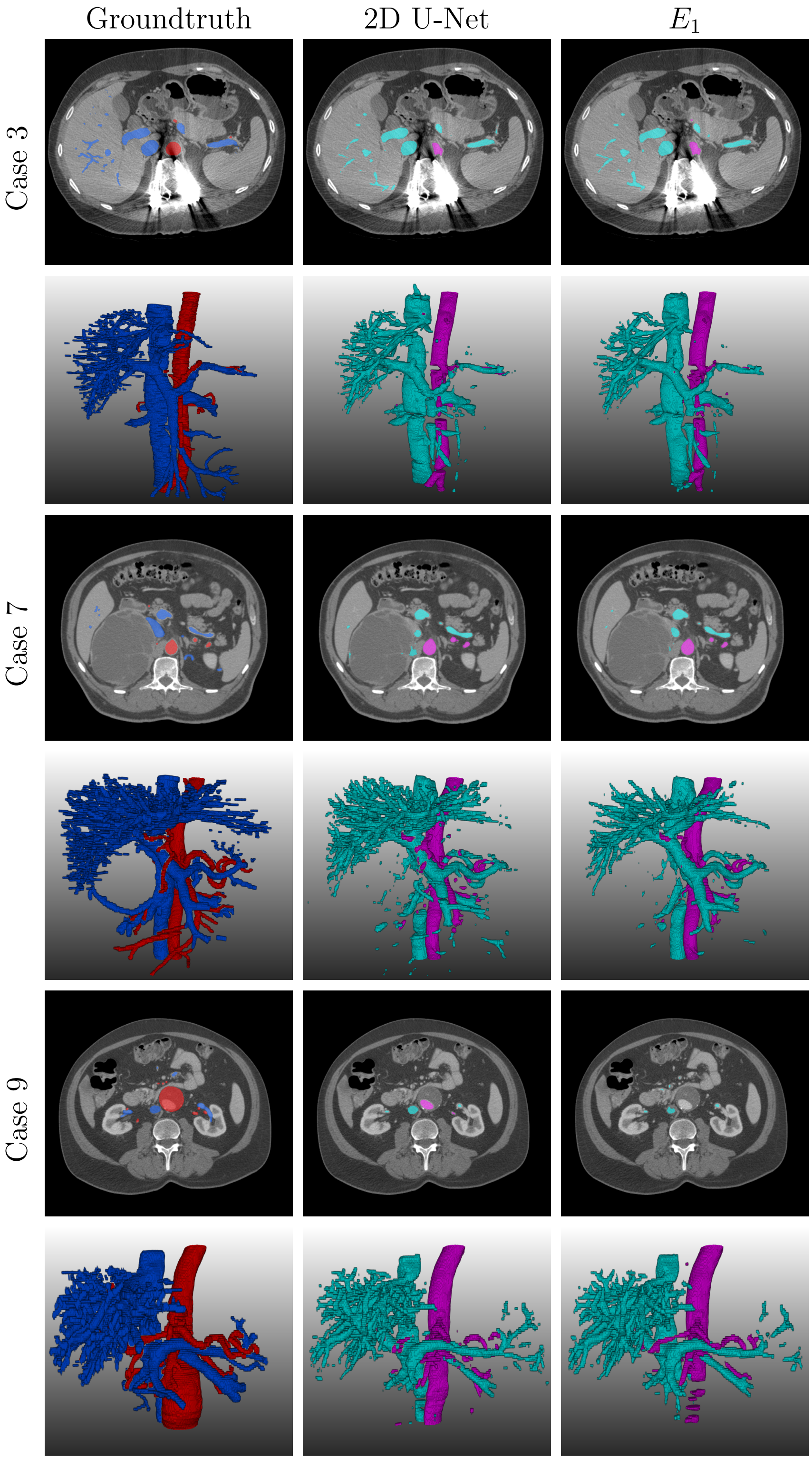


Figure 6.4: Experiment 2 - Visual comparison of segmentation results for three challenging IRCAD cases. The strong metal artifacts in case 3 lead to holes in the aorta segmentation. The networks were able to correctly segment the distorted vessels in case 7, but failed to segment the outer layer of the aneurysm in case 9.

Table 6.7: Experiment 2 - Quantitative comparison of the segmentation quality of seven different ensembles.

		DSC ↑	ASSD ↓	VCR ↓	C ↑
Veins	E_2	0.756±0.050	2.050±0.772	0.005±0.005	0.891±0.145
	E_2	0.752±0.051	2.041±0.808	0.006±0.006	0.889±0.127
	E_3	0.728±0.056	2.264±0.803	0.007±0.006	0.844±0.189
	E_4	0.741±0.054	2.125±0.769	0.007±0.006	0.879±0.153
	E_5	0.751±0.051	2.145±0.826	0.006±0.006	0.881±0.115
	E_6	0.758±0.050	2.038±0.774	0.005±0.005	0.889±0.143
	E_7	0.755±0.051	2.055±0.788	0.006±0.006	0.892±0.150
Arteries	E_1	0.838±0.075	3.081±1.841	0.008±0.009	0.960±0.049
	E_2	0.835±0.068	2.870±1.335	0.012±0.016	0.941±0.052
	E_3	0.823±0.077	3.177±1.766	0.011±0.009	0.965±0.020
	E_4	0.832±0.071	3.113±1.608	0.012±0.016	0.954±0.051
	E_5	0.833±0.075	2.793±1.282	0.010±0.009	0.946±0.047
	E_6	0.838±0.074	3.089±1.883	0.008±0.011	0.957±0.050
	E_7	0.836±0.074	3.044±1.728	0.009±0.013	0.953±0.051

6.3.3 Experiment 3

We selected the ensembles from Experiment 2 as well as their constituent networks for retraining on the IRCAD data set and application to the BTCV data set. Evaluation results are listed in Table 6.8. The highest DSC for both classes was achieved by the ensembles. However, the best vein DSC was 6.33% lower than the best result from Experiment 2, while the arterial DSC was nearly the same. The lowest VCRs and ASSDs were also produced by the ensembles. All networks and ensembles achieved similar results for the venous C, but the ensembles achieved the best results for the arteries. The differences between ensembles were less prominent than in Experiment 2. An example for one case is shown in Figure 6.5.

6.4 Discussion

In the first experiment the auto-encoder architectures outperformed the D-Net significantly. The D-Net was originally developed for the segmentation of cerebral vessels in MRA data. Such data has a much more uniform background than abdominal images. The differentiation between a complex background and vascular structures seems to be too complicated for the network to learn. Due to the far lower number of feature maps and operations the D-Net cannot learn features as complex as the U- and V-Net. Discrimination between a more homogeneous background like liver parenchyma and veins, however, is in fact possible.

For the majority of the networks and metrics the differences between individual and simultaneous training were not significant. Exceptions were the 3D U-Net, which

Table 6.8: Experiment 3 - Quantitative comparison of the segmentation transfer quality from IRCAD to BTCV.

		DSC ↑	ASSD ↓	VCR ↓	C ↑
Veins	2D U-Net	0.689±0.125	2.401±1.911	0.007±0.007	0.686±0.221
	2D V-Net	0.680±0.142	2.186±1.941	0.006±0.008	0.677±0.235
	3D U-Net	0.666±0.113	2.883±2.925	0.009±0.015	0.762±0.181
	3D V-Net	0.667±0.116	2.917±2.710	0.011±0.017	0.702±0.204
	E_1	0.709±0.135	2.086±2.078	0.005±0.007	0.762±0.208
	E_2	0.710±0.133	2.060±2.0210	0.005±0.007	0.736±0.224
	E_3	0.706±0.133	2.041±1.966	0.005±0.007	0.725±0.230
	E_4	0.708±0.135	2.135±2.145	0.005±0.007	0.757±0.203
	E_5	0.706±0.133	2.246±2.416	0.005±0.008	0.760±0.201
	E_6	0.696±0.128	2.538±3.105	0.005±0.007	0.752±0.190
	E_7	0.698±0.140	1.994±1.741	0.005±0.008	0.741±0.208
Arteries	2D U-Net	0.830±0.064	2.610±1.343	0.006±0.008	0.942±0.065
	2D V-Net	0.811±0.080	2.732±1.502	0.011±0.007	0.912±0.087
	3D U-Net	0.821±0.103	2.409±1.871	0.006±0.006	0.964±0.022
	3D V-Net	0.784±0.161	2.833±2.975	0.008±0.007	0.946±0.074
	E_1	0.833±0.087	1.988±1.589	0.003±0.004	0.976±0.017
	E_2	0.835±0.081	1.994±1.488	0.003±0.005	0.965±0.045
	E_3	0.833±0.077	2.094±1.429	0.004±0.005	0.960±0.049
	E_4	0.834±0.084	2.060±1.547	0.003±0.004	0.977±0.018
	E_5	0.836±0.080	2.120±1.495	0.003±0.004	0.977±0.019
	E_6	0.837±0.075	2.210±1.340	0.004±0.005	0.971±0.024
	E_7	0.811±0.115	2.122±1.767	0.006±0.005	0.941±0.107

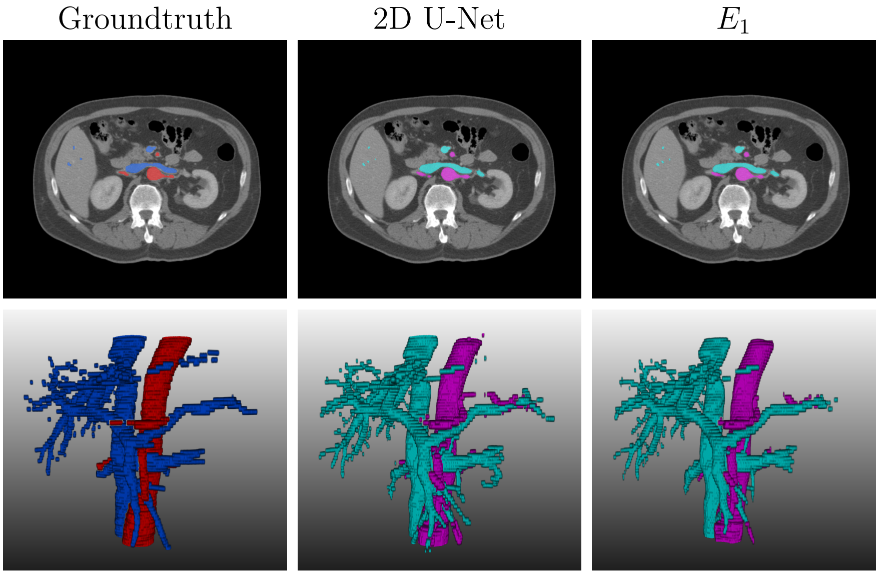


Figure 6.5: Experiment 3 - Visual comparison of segmentations for BTCV case 31.

performed significantly better with individual training for the venous VCR, as well as the 2D U-Net which achieved a significantly lower ASSD with simultaneous training. From this follows that the simultaneous training does not lead to a loss of accuracy compared to individual segmentation, however there was also no prevalent benefit compared to individual segmentation.

U-Net and V-Net both performed similarly in 2D. While the 3D U-Net was only able to achieve comparable results for VCR and C, the 3D V-Net performed on par with the 2D networks for the artery class. The venous vessel trees in our data contain many small vessels close to the resolution limit. This seems to be difficult for the 3D networks to segment. The artery class on the other hand consists mostly of the aorta. This large structure can be well segmented by the 3D networks. Both 3D auto-encoders did however produce segmentations with high structural integrity according to C.

The auto-encoders had VCRs of 1% and lower, we can therefore conclude that our system reliably differentiates between arteries and veins. This high accuracy is comparable to the performance of graph-based separation approaches [125]. However, our direct dual class segmentation does not require an additional algorithm to segment the vessels in the first place. Our CNN approach also does not require the selection of hand-crafted features for the separation.

Combining different architectures can boost performance by up to 9.7%. This is a higher improvement than has been previously reported by Winzeck et al. for averaging the networks with different initializations [115]. Heterogeneous ensembles can unify the advantages of the networks and therefore predictions become more reliable. The accuracy is especially increased for small visceral vessels with diameters close to the image resolution. Weighting well performing architectures slightly higher can enhance the results as shown in Experiment 2. However this does not necessarily translate to other data as seen in Experiment 3. Equally weighting well performing architectures results in a robust ensemble and can therefore be recommended as the approach of choice.

Performance loss when applying the networks to another data set is common, however we saw a substantially stronger performance loss for the vein class. A major reason for this was the low quality contrast enhancement in the BTCV data set. The artery class was not affected by this as all images were taken in the portal-venous phase. The automatic processing of CE-CT highly depends on standardized protocols and the reliable presence of contrast agent in the target structures. Insufficient contrast in the images is detrimental to the segmentation quality. Similarly to the IRCAD data the networks were not able to correctly segment the vessels in the presence of metallic objects such as stents. With regards to this and the three more complex IRCAD cases, we saw that anomalies, which were not present in the training data, are challenging for the individual networks and the ensembles. Especially metal artifacts, e.g. from vena cava filters or pedicle screws, currently limit the CNN ensemble. Training CNNs for robust segmentation requires diverse training data sets showing a large variety of pathologies and artefacts.

Compared to other state-of-the-art methods (see Table 6.1) we achieve comparable results for the arteries and superior results for the veins. Except for Russ *et al.* none of the other methods worked on the IRCAD data set and we used further refined annotations. Others only target a subsection of the vascular trees e.g. the hepatic veins. They also only address a two class problem, while our approach solves a three class problem.

6.5 Conclusion

In conclusion, we have proposed a novel CNN ensemble which extracts and seperates the venous and arterial vessel tree from abdominal CE-CT images. Our experiments showed that auto-encoder architectures such as the U- and V-Net perform better for this complex task than the D-Net. Combining 2D and 3D versions of these networks creates a high performance segmentation algorithm. While 2D networks performed best regarding the DSC, we saw that 3D networks enhance the structural integrity of the segmentations. The effectiveness of the proposed method was shown on the IRCAD data set achieving a DSC of 0.758 ± 0.050 for the veins and 0.838 ± 0.074 for the arteries. Confusion rates between the two vascular systems were lower than 1.5%. We further showed that the trained networks can be directly applied to other data sets. Our state-of-the-art results are comparable to systems which only segment one of the target structures.

We plan to extend the proposed method to CE-CT scans in the arterial phase so that the extracted information can be used for planning and intra-procedural navigation of minimally invasive catheter-based therapies such as Transarterial Chemoembolization (TACE) for oncologic treatment of liver tumors [126]. Another possible application would be the segmentation and separation of pulmonary vasculature, however we expect this to be more challenging due to the non-contrast imaging [127]. Moreover, we will further investigate how robustness in regards to metal artifacts can be enhanced.

The proposed method, which combines an optimized training pipeline with an ensemble of established segmentation architectures, is widely applicable to segmentation of vessels and other small structures that suffer from a high class imbalance. Given that the current limitations can be overcome by fine-tuning on an extended data set, our method has high potential to improve patient care.

7. "Automated Screening for Abdominal Aortic Aneurysm in CT scans under clinical conditions using Deep Learning", *submitted European Radiology, 23.03.2021*

7.1 Introduction

AAA is a potentially life-threatening condition. [128–130]. Possible rupture is associated with a high mortality exceeding 50 % [131–133]. In clinical routine, small AAA presents itself as a co-finding on abdominal CT images performed for various reasons. The focus on other clinical questions and the time-consuming nature of AAA analysis might lead to underreporting and delayed diagnosis [134]. Therefore, patients might get discharged without detection of early AAA and therefore might not receive beneficial integration into surveillance strategies or commencement of treatment.

However, early initiation of surveillance and risk factor modification are possible if small AAA is correctly diagnosed [135]. Ultimately, the risk of spontaneous rupture can be significantly lowered if diagnosed larger AAA are treated surgically or interventionally [136]. Artificial intelligence seems highly suited to contribute to management of AAA.

7.1.1 Related Work

With DL being introduced for various use-cases in medicine, a new and promising era of technical support and guidance for physicians is emerging [57]. Research effort has recently been made towards utilizing deep learning for AAA detection, segmentation and prognostic evaluation.

Mohammadi *et al.* [137] and López-Linares *et al.* [138] both described a 2D convolutional neural network (CNN)-based cascading pipeline for automated detection and segmentation of AAA in abdominal CT scans. Habijan *et al.* [139] have applied a 3D U-Net with deep supervision to segment AAAs in CT. Such advances in medical image segmentation allow for exact measurement of diameters and vessel lengths and are very useful tools for intervention planning and optimized graft selection [129]. Zhang *et al.* [140], Do *et al.* [141] and Garcia-Garcia *et al.* [142] developed algorithms to predict AAA growth from abdominal Computed Tomography Angiography (CTA) scans. The works of Harris *et al.* [143] and Cao *et al.* [144] both exemplarily show the potential of artificial intelligence for familiar questions like detection (and partly segmentation/classification) of Stanford type B aortic dissection.

Hahn *et al.* [145] furthermore developed an algorithm that shows promising results for detection of vascular endoleaks after Endovascular Aneurysm Repair (EVAR) implantation in AAA. However, these tools have in the majority of cases not been developed for routine clinical application.

A robust and automated algorithm that can be included in routine clinical workflow remains a great challenge [146]. The lack of algorithm generalizability is a central obstacle, which usually is caused by development on highly pre-selected data sets containing scans from a limited amount of scanners and mostly exclusively CTA contrast phases.

7.1.2 Contribution

The aim of this study is to develop and describe a fully automated 3D AAA screening algorithm, which can run as a background process in the clinic workflow. The main requirements are robustness, reliability and precision, whereas algorithm training and clinical application should be feasible with minimal effort. In a validation step, correct focus of the algorithm onto the aortic lumen should be confirmed.

7.2 Materials and Methods

7.2.1 Patients and Data Set

Ethical approval for this study was obtained from the ethics committee II of the Medical Faculty Mannheim, Heidelberg University (2016-863R-MA, November 17th 2016). We acquired the described data set from our Radiology Information System (RIS). The scans were extracted from our Picture Archiving and Communication System (PACS). The presence of an AAA was confirmed by a radiology resident with 1.5 years of experience in the interpretation of abdominal CT scans. The abdominal aorta was considered aneurysmatic when the aorta exceeded a 50 % increase of the regular diameter [147]. The scans derived from a total of seven different SIEMENS CT scanners located at two different medical sites. All scans that had been performed with contrast agent have been included. Some scans were acquired 70 seconds post-contrast media application (venous), some in CTA contrast phase (arterial) and some in an mixed phase of 40-50 seconds post-application, which were classified as venous studies within this analysis. The data set contains scans of AAA of various size and shape with multiple co-findings and various devices like intra-arterial stents and metallic interferences being present.

Annotation of the data set was performed in two ways. Firstly, each CT scan was assigned to one of two classes (0: no AAA, 1: AAA). Secondly, the axial position of the origin of the renal arteries for the aorta was noted as an anchor point in every scan, which was subsequently used to automatically extract a sub volume of standardized size. Four example cases are shown in Figure 7.1. Distribution of classes and further statistics of our data set are listed in Table 7.1.

7.2.2 Networks, Preprocessing and Training

We extended three established architectures from literature to be applied to 3D image classification: AlexNet [19], VGG-16 [20] and ResNet [21]. We altered the

Table 7.1: Data set statistics - Spacing is given as median ± standard deviation.

Cases (total)	187
Cases in arterial phase	85
Cases in venous phase	102
Cases with metal artifacts	44
Cases with stents	15
Cases with AAA	100
Voxel spacing X/Y	0.9 ± 0.1 mm
Voxel spacing Z	1.5 ± 0.5 mm
Slices [min, max]	[101, 2687]
Scanners (Siemens)	SOMATOM Force, SOMATOM Definition Flash, SOMATOM Definition AS 128, SOMATOM Sensation 64, SOMATOM Emotion 16, SOMATOM Emotion 16, Biograph mCT

described architectures to economize memory consumption. All networks use the ReLU activation function, except for the last layer, where the softmax is used. The architectures are shown in Figure 7.2. Details of the implementation are provided in the supplementary material.

We use stratified 5-fold cross-validation. The data is split into five disjoint test sets, each consisting of 36-39 cases. For each fold, one of these test sets is used and 6 cases are selected for validation from the non-test data. The remaining non-test cases are used for training. The densities of the CT images are windowed to range between $[-200\,\mathrm{HU}, 400\,\mathrm{HU}]$ and then mapped to the interval $[-1, 1]$. All images are resampled to a spacing of 0.9 × 0.9 × 1.5 mm. During training this resolution is changed by up to 10 %. Furthermore, data augmentation is performed via rotation around the cranio-caudal axis with angle $\alpha \in [-12.6°, 12.6°]$. Additionally, density jittering by up to ±3 HU is applied. 3D patches of 320 × 384 × 224 voxels are extracted from the CT volumes. This reduces memory requirements and removes the parts of the image depicting air and the patient table. For prediction, the patch is centered on the center of the anchor point slice. During training the center is shifted by up to ten voxels along the cranio-caudal axis and is randomly positioned on the slice.

Training is performed using the Adam optimizer with a learning rate of 10^{-3}. We use the binary cross entropy loss and apply L2 regularization on the network weights. The regularization term is weighted with 10^{-5}.We apply early stopping to the training process. After the initial 20 epochs, we test for the convergence criterion at the end of each epoch. The criterion is defined as the change of validation accuracy over the last twenty epochs falling below 10^{-4}. For application we then select the weights with the highest accuracy on the validation data from the last 20 epochs.

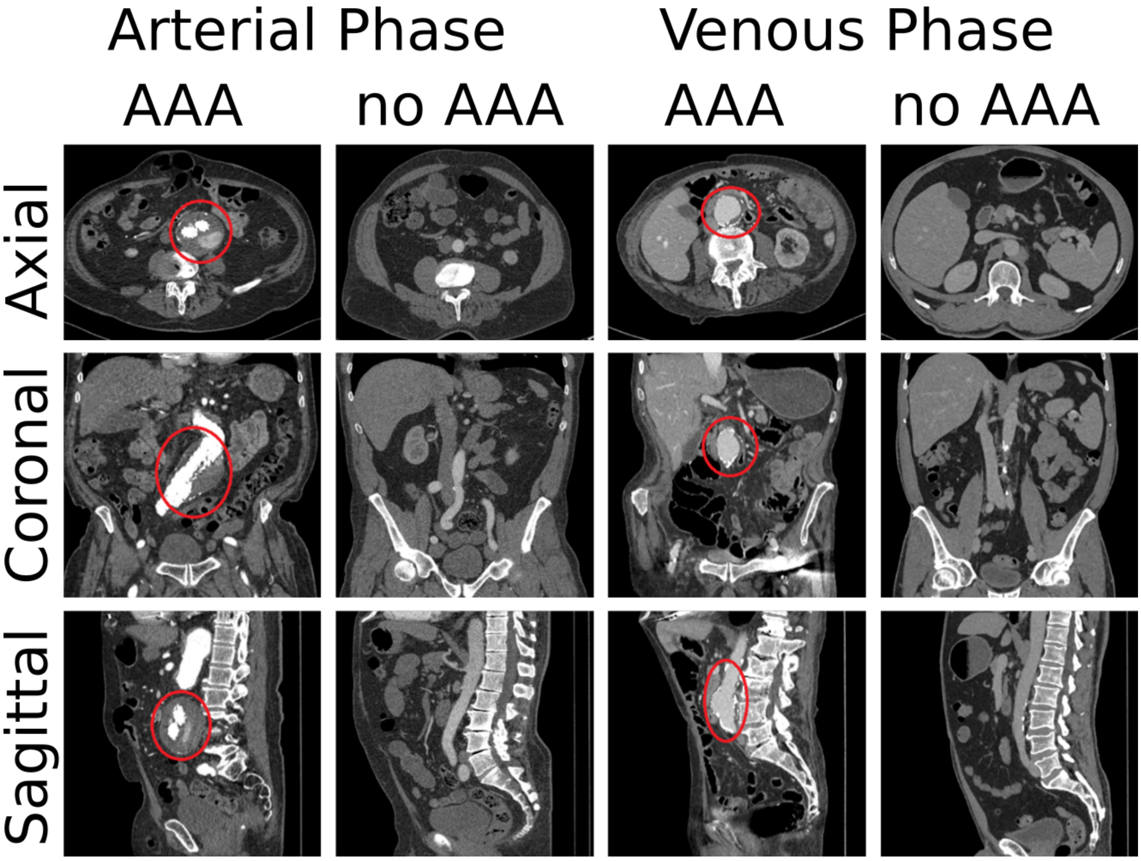


Figure 7.1: Four example cases from our data set. For each case one slice of each principal anatomical plane is shown. The location of the AAAs is marked with a red circle.

7.2.3 Layer-wise Relevance Propagation

LRP allows the calculation of voxel-wise decomposition of the decision of a CNN and can thus be used to provide interpretability for CNNs [148]. The relevance is propagated layer-wise from the network output back through the network, until the input layer is reached. The final result equals a relevance map, which provides a relevance value on the output class for each single input value.

We apply LRP to analyze the relevance for the network decision on the AAA class. Positive relevance values therefore indicate relevance for the AAA class, while negative values indicate relevance against the AAA class. The relevance values are dimensionless quantities and not comparable across samples or networks. We therefore, normalize the maps by their sum to standardize them [149, 150].

7.2.4 Evaluation

We employ five metrics to assess the quality of the predicted classification. We use a discrimination threshold of 0.5 for all four binary metrics. They are derived from four outcomes: TP represents a sample correctly identified as AAA, TN denotes a non AAA sample correctly classified as such, FP is an AAA sample falsely identified as a non AAA case and FN denotes a non AAA sample being misclassified as AAA. We employ accuracy (A), precision (P), recall (R) and the F1 score (F_1) [9]. These

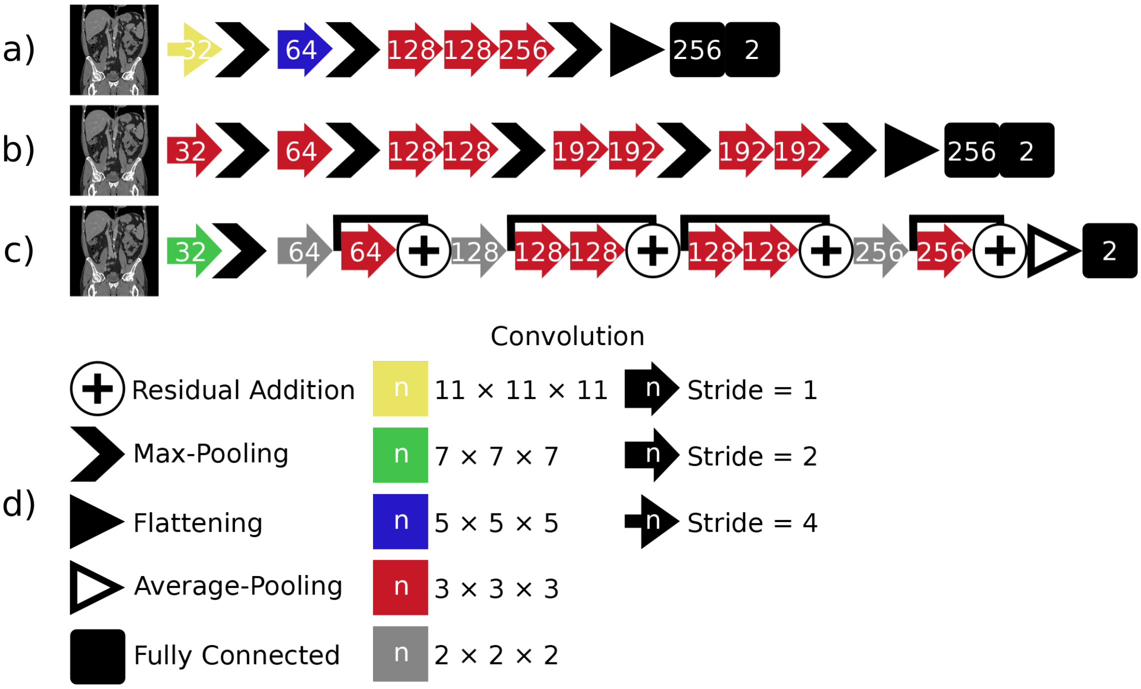


Figure 7.2: The three architectures used for classification are shown on the left: a) 3D AlexNet, b) 3D VGG and c) 3D ResNet. All operations are given in the legend on the right d). The number of channels resulting from operations is marked by the number on the operation symbol.

measures compare the predictions of the networks to the labels. The Receiver Operating Characteristic (ROC) curve plots the ratio between TP and FP decisions, when the discrimination threshold of a binary classifier is varied. It evaluates the performance based on the predicted probabilities of the networks and the labels. The Area Under the Curve (AUC) provides a metric for the classifier performance.

To asses the correspondence of the relevance maps with the aorta we use a 5-point Likert scale. Positive and negative relevance values were considered equally for the assessment. The influence of the aorta on the network decision is scored as: 1 (no relevance in the aorta), 2 (low relevance in the aorta), 3 (medium relevance in the aorta), 4 (high relevance in the aorta) and 5 (very high relevance in the aorta).

7.3 Experiments and Results

To assess the performance of the CNN classification we performed three experiments:

1. We trained and tested three architectures. This was only done for one fold.
2. LRP is applied to the best network from experiment 1 to validate that the network decision is based on the correct Region of Interest (ROI). Relevance maps were scored by an experienced radiologist.
3. The training of the best network is repeated for the four remaining folds to verify repeatability.

The results of the evaluation metrics for the first experiment are listed in Table 7.2. The ROC curves for the three networks are shown in Figure 7.3. 3D ResNet achieved the best results according to all five metrics. Out of the 3D ResNet's five false positive cases, two cases showed a pre-aneurysmatic enlargment.

Table 7.2: Experiment 1 - Quantitative comparison of the classification quality of three different networks. The best result for each metric is marked **bold**. ↑ marks metrics for which higher values are better.

Network	A ↑	P ↑	R ↑	F_1 ↑	AUC ↑
3D AlexNet	0.769	0.773	0.810	0.791	0.899
3D VGG	0.769	0.800	0.762	0.780	0.860
3D ResNet	**0.872**	**0.808**	**1.000**	**0.894**	**0.931**

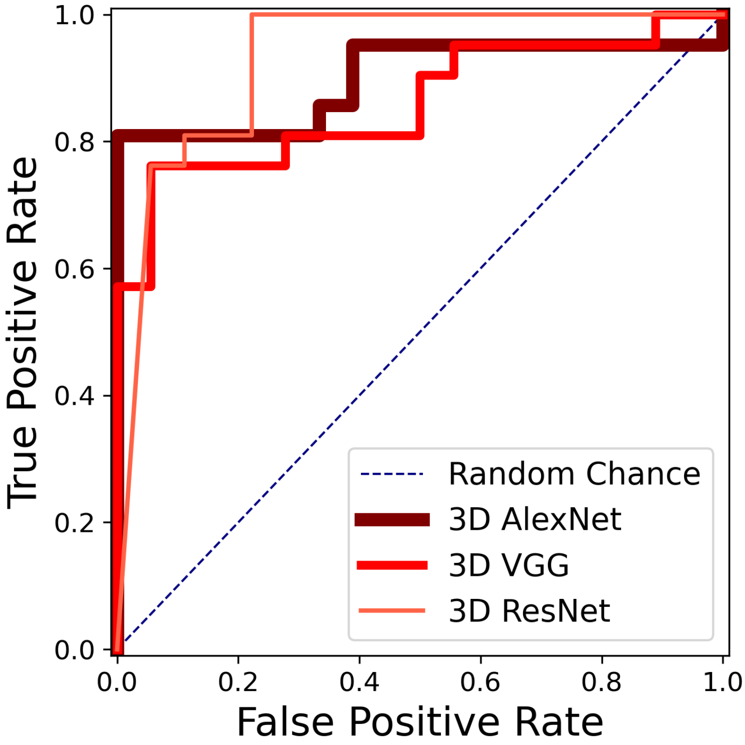


Figure 7.3: Experiment 1: Receiver operating characteristic (ROC) curves for 3D VGG, 3D AlexNet and 3D ResNet.

In the second experiment, the rating of the relevance maps resulted in an average score of 4.56 for correctly classified cases. The exact distribution can be seen in Figure 7.4 c). Relevance maps for a case with and a case without AAA are shown exemplarily in Figure 7.4 a) and Figure 7.4 b), respectively. For the AAA class, highest relevance values were present on the inner lumen of the aneurysm. There was one case where no relevance was present in the aorta. This patient had an Extracorporeal Membrane Oxygenation (ECMO) tube placed in the vena cava. The relevance analysis showed that the network focused on the vena cava instead of the aorta.

The results of the evaluation metrics for the five folds and across all data (aggregated from the five folds) from the third experiment are listed in Table 7.3.

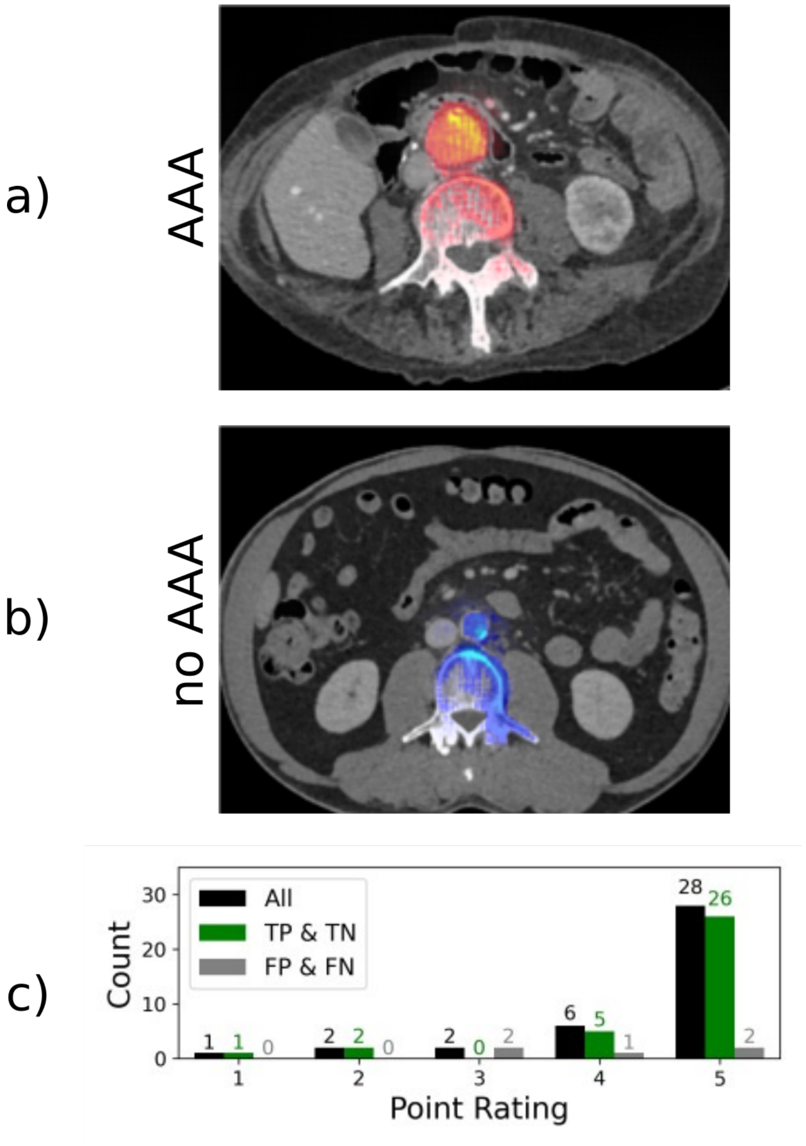


Figure 7.4: Experiment 2 - Results of the analysis using LRP. a) and b) show relevance maps for predicting the AAA super-imposed on the CT images for two examples cases. (a) shows a case with AAA, while (b) shows a case without AAA. Positive values are shown in red with high values in yellow, while negative values are shown in blue with high values as light blue. The high relevance around the aorta confirms that the networks correctly learned to make the classification decision based on the ROI. c) shows the score distribution of the assessment by an experienced radiologist.

Table 7.3: Experiment 3 - Quantitative comparison of the classification quality of 3D ResNet across 5 folds.

Fold	A ↑	P ↑	R ↑	F_1 ↑	AUC ↑
1	0.872	0.808	1.000	0.894	0.931
2	0.821	0.850	0.810	0.829	0.929
3	0.838	0.850	0.850	0.850	0.919
4	0.944	0.947	0.947	0.947	0.961
5	0.806	0.773	0.895	0.829	0.885
All	0.856	0.841	0.900	0.870	0.926

7.4 Discussion

The findings of this study show, that our 3D ResNet demonstrated a high performance and robustness in detecting AAA in abdominal CT scans with a resulting AUC of 0.926 and accuracy of 0.856 while only requiring only minimal annotations. Based on these findings, the architecture seems suitable for clinical screening purposes.

Ultrasound-based AAA screening has been described to decrease AAA mortality and increase effectiveness of treatment and has partly been introduced into clinical practice [151, 152]. Automated AAA screening on abdominal CT scans that have been acquired for various other reasons would add another chance of early detection of AAA, possibly supporting radiologist and clinicians in reporting, monitoring and treating AAA [134].

Time-consuming manual annotation remains a major bottleneck of training and validation of newly developed algorithms on large clinical data sets [153, 154]. The screening algorithm presented here in contrast was designed to be trainable on minimally annotated data. Setting of the anchor point could in the future be automated via anatomical landmark detection [155] and would provide the opportunity for training and validation on even larger data sets. A critical factor for the successful clinical implementation of artificial intelligence lies in the transfer of results back into the clinical systems. The LRP analysis presented here for verification is able to generate a graphical analysis of the decision-relevant areas in the imaging material. PACS export of these results might significantly increase clinical acceptance. Detailed effects on patient treatment represent an exciting field of further research.

Different groups have already described pipelines and tools for detection and classification of AAA in CT scans. A general shift of focus towards compatibility with routine clinical implementation can be observed, however this process remains challenging [137, 146, 156–158]. Different levels of performance have been reported for AAA related image processing tasks. Mohammadi *et al.* reported a high accuracy of successfully classifying 2D patches to show the aorta (0.986) [137]. For the prediction of aortic dissection and aortic rupture using a 2D CNN a recall of 0.900 and 0.889 as well as AUC of 0.979 and 0.990 were reported by Harris *et al.*, respectively [143]. Our 3D ResNet achieves a similar recall, but slightly lower AUC values than these works. However, the task of classifying an entire 3D volume is also more complex than classifying selected 2D patches and we targeted a different problem. To the best of our knowledge and based on our extensive literature research, the application of 3D CNNs to CT AAA screening has not been presented previously. Solutions which apply 3D CNNs for whole CT binary classification of other pathologies in CT data report accuracies of 0.918 (lung cancer [159]) and 0.93 (Covid-19, [160]).

It is important to note that the results achieved with the algorithm proposed in this study have been validated on an heterogeneous data set of 187 images, which includes different contrast phases, scanners and artifacts. Mohammadi *et al.* trained and tested their algorithm on a data set of solely ten patients with two patients having an “obvious” AAA [137]. Whereas in one of our previous studies [161], metallic interferences have shown to negatively impact CNN performance, in this study accuracy remained high even with presence of artifacts. The algorithm described here proofed to be robust enough to achieve reliable results under these conditions which represent the reality of clinical work. The authors believe that validation on this unfiltered clinical data set marks a major step towards integration of the newly developed technology into clinical practice.

3D ResNet produced some FP results. Two of the FP cases, however, did show enlargements of the aorta. These were not strong enough to be classified as aneurysms using our criteria yet, but significant enough to be confirmed as enlargements by a radiologist. These detections of early enlargements are adding up to the potential of the network for screening purposes. Extension of the criteria to small aneurysms in the annotation, could subsequently yield detection of even earlier enlargements. The application of LRP confirmed that the network based it’s decision making process on the aorta region. For one patient with an ECMO tube placed in the vena cava the network focused on the vena cava instead of the aorta. The classification result for the case was, however, correct. We assume that the ECMO tube was interpreted as a stent by the network and therefore caused the mislocated area of interest. For further extension of the data set inclusion of more patients with such inserts seems important.

Limitations of the technology on the one hand are caused by the design of the algorithm as a binary classification task instead of object detection or segmentation. Our solution does not allow for automated segmentation and volumetry (which has already been achieved by other authors). Screening could potentially also be solved with an object detection network, this would however require far more extensive annotations. Future directions lie in the practical use of the technology developed within this study to notify clinicians in real-time. Also, in cases of relevant AAA a second pipeline could be triggered which includes fully automated AAA segmentation and analysis, like the one introduced by Lareyre *et al.* [157]. On the other hand, limitation is caused by the validation that could not be performed on a large international cohort with different CT scanner manufacturers. Our current data set of 187 cases does cover many variations of clinical scans (especially since they derive from a large University Medical Center with a high variety of different findings present), but not all. We employed a range of data augmentation methods to simulate further variations, but training and validation on a much larger international cohort seems paramount for integration into the workflow of different medical sites. These limitations have to be considered when interpreting the results. Training of the developed methodology on a larger cohort would be possible with reasonable extra effort and therefore should be done to ensure broad generalizability.

From a medical and clinical perspective, it will be exciting to investigate if earlier detection of small AAA contributes to effective patient treatment and improvement of prognosis and outcome. A possible observer study monitoring effects of the technology on patient care would be a logical and necessary next step.

This study demonstrated the feasibility of CNN-based fully automated detection of AAA in an unselected clinical data set of 187 abdominal CT images. Accuracy, Precision, Recall, F1 score and ROC AUC of 0.856, 0.841, 0.900, 0.870 and 0.926 was achieved. Our 3D ResNet seems to be suitable for screening purposes in routine clinical workflows. Possible generation of relevance maps contribute to explainability of the decision process and could be exported into clinical workflows. Integration of this deep learning screening for AAA into routine workflow might lead to improved patient monitoring, earlier diagnosis and improved patient treatment with possible reduction of rupture risk.

8. Summary

The field of digital image processing has been revolutionized by DL. Due to this technology, completely automated image analysis with high accuracy is starting to become available. In the medical domain, DL has been applied to a variety of application areas [1]. The technology is not limited to a singular task. In recent years, several CNN architectures for a range of image processing applications have been presented. Independent of the task and the application, the training procedure for these networks always consists of the same basic steps. For tomographic images, this includes specialized preprocessing with regard to the modality and the image spacing. As tomographic data such as CT and MRI is 3D, the usage of 3D CNNs is especially promising, however their training is more complex and requires large amounts of GPU memory in contrast to 2D networks.

In this work a pipeline for training and application of DL methods in medical image processing was developed. The implementation is based on the commonly used TensorFlow library [37] and has been adapted for three different release versions. The pipeline directly loads medical image data and performs resampling in accordance with the header information. This is realized using SimpleITK which provides a simplified interface to the Insight Toolkit (ITK) [38]. Basic functionalities, such as the data handling between SimpleITK and Tensorflow, are implemented in a basis module. More specialized task-specific modules add further functionalities. The pipeline is modular, so individual components, such as the network architecture, can easily be switched. All processing steps are accessible as 2D and 3D operations. A wide range of preprocessing methods can be individually chosen for the task. Data augmentation is implemented as an online process, that generates new samples in every epoch. Regularization methods such as weight regularization and early stopping are available. The training progress is automatically documented using TensorFlow's TensorBoard functionality.

In the current version three types of image processing tasks can be handled: image regression, semantic segmentation and image classification. The loading process is adapted for each of these. For segmentation and image regression the label is also altered during data augmentation, while this is not the case for classification. Several state-of-the-art architectures for the different tasks have been implemented within the framework. Additionally, state-of-the-art loss functions and evaluation metrics have been added.

In Chapter 4 to Chapter 7 different scientific studies are presented which show the successful application of the pipeline to the three image processing tasks: classification, segmentation and regression. A detailed summary for each study is provided in the following paragraphs.

Simulation-Based Deep Artifact Correction with Convolutional Neural Networks for Limited Angle Artifacts,
Z Med Phys, doi: 0.1016/j.zemedi.2019.01.002

In Chapter 4 the ability of CNNs to correct limited angle artifacts in cTS scans has been investigated. This work presents the application of the first version of the training pipeline to image regression. DAC was realized with three U-Net-based networks and a 3D-ResNet auto-encoder. The availability of ground truth data is often limited in image processing and commonly not available in medical imaging. This study explored the use of simulated training data from a digital phantom, which offers a promising solution to this problem.

This study was not targeted at a specific anatomical area, but rather at a specific imaging method. Non-conventional scan trajectories for interventional three-dimensional imaging promise low-dose interventions and a better radiation protection to the personnel. cTS scan trajectories yield an anisotropical image quality distribution. In contrast to conventional CTs, the reconstructions have a preferred focus plane. In the other two perpendicular planes, limited angle artifacts are introduced. A reduction of these artifacts leads to enhanced image quality while yielding lower dose exposure.

The study showed that limited angle artifacts can be mitigated using simulation-based DAC. The U-Net-corrected cTS achieved a RMSE of 124.24 HU on 60 simulated test scans in comparison to the digital phantoms. This equals an error reduction of 59.35% from the cTS. The achieved image quality is similar to a simulated CBCT. The presented network was also able to mitigate artifacts in scans of objects which strongly differ from the training data. Application to real cTS test scans showed an error reduction of 45.18% and 26.4% with the 3D-ResNet in reference to a high-dose CBCT.

Deep Voxel-Guided Morphometry (VGM): Learning regional brain changes in serial MRI,
MLCN-LNCS, doi: 10.1007/978-3-030-66843-3_16

A second application of the pipeline to image regression was presented in Chapter 5. This study showed the application to change detection serial MRI scans for the progression assessment of MS. In this second version of the pipeline, the sample mining has been refined to select patches from the non-air image regions. Additional normalization methods for MRI have been added. The whole pipeline was also updated to a new Tensorflow version. The memory intensive prediction of 3D volumes was restructured to predict overlapping regions and merging them subsequently.

The area of application for this study was neurology. Progression assessment is an essential, yet challenging task for the clinical management of MS. Analysis algorithms such as VGM enable detection and quantification of even minor changes of the brain at different time points. To shorten computation times and ameliorate clinical applicability, a CNN based VGM (Deep VGM) was developed. It provides a fast solution for intra-individual serial volume change analysis in MS. Deep VGM is a residual architecture based on the 3D U-Net. Several loss functions to predict VGM maps from a base line and a follow up brain MRI were investigated. The approach was trained and tested in 71 MS patients. The Deep VGM maps were compared to the respective VGM maps via several image metrics and rated by an experienced neurologist.

Deep VGM configured with the Mean Absolute Error and Gradient loss outperformed all other tested loss functions. Deep VGM maps showed high similarity to the original VGM maps (SSIM = 0.9521 ± 0.0236). This was additionally confirmed by a neurologist analyzing the MS lesions. Deep VGM resulted in a 3% lesion error rate compared to the original VGM approach. Computation time of Deep VGM was 99.62% shorter than VGM. The experiments demonstrate that Deep VGM can approximate the complex VGM mapping at high quality while saving computation time.

Convolutional Neural Network Ensemble Segmentation with Ratio-based Sampling for the Arteries and Veins in Abdominal CT Scans, *IEEE T-BME, doi: 10.1109/TBME.2020.3042640*

Chapter 6 investigated several CNN architectures for multi-class segmentation. Functionality to switch between 2D and 3D processing has been further refined for this study. Vessel segmentation is a task, that is heavily effected by class-ratio imbalance. Therefore, ratio-based sampling was proposed to counter this problem. The method selects training patches centered on object and background voxels in a 50 : 50 ratio. This is also the ratio between object and background patches and every single training patch. The training process has also been extended to include early stopping.

The area of application for this study was abdominal vasculature. 3D blood vessel structure information is important for diagnosis and treatment in various clinical scenarios such as living donor organ transplantations. A fully automatic method for the extraction and differentiation of the arterial and venous vessel trees from abdominal CE-CT volumes using CNNs was presented. Using the pipeline 2D and 3D versions of the U-Net, the V-Net and the DeepVesselNet were trained. All networks were trained with a combination of the Dice and cross entropy loss.

Performance was evaluated on 20 IRCAD subjects. Best performing networks were combined into an ensemble. Seven different weighting schemes were investigated. Trained networks were additionally applied to 26 BTCV cases to validate the generalizability. Based on the experiments, the optimal configuration is an equally weighted ensemble of 2D and 3D U- and V-Nets. The method achieved DSCs of 0.758 ± 0.050 (veins) and 0.838 ± 0.074 (arteries) on the IRCAD data set. Application to the BTCV data set showed a high transfer ability. Abdominal vascular structures can be segmented more accurately using ensembles than individual CNNs. 2D and 3D networks have complementary strengths and weaknesses. The ensemble of 2D and 3D U-Nets and V-Nets in combination with ratio-based sampling achieves a high agreement with manual annotations for both artery and vein segmentation. The achieved results surpass other state-of-the-art methods.

Automated Screening for Abdominal Aortic Aneurysm in CT scans under clinical conditions using Deep Learning, *submitted Eur Rad, 15.01.2021*

Finally, in Chapter 7 the application to binary classification was demonstrated. Therefore, the data handling was extended to handle classification labels. The pipeline has been adapted to a third Tensorflow version for this project to enable the use of the iNNvestigate neural networks! toolbox [162] for LRP.

The area of application for this study was abdominal vasculature. AAA is a critical condition that is often diagnosed as a secondary finding. A dataset of 187 CT scans with various contrast enhancements and artifacts consisting of patients with and without AAA was aggregated. ResNet, VGG-16 and AlexNet were adapted for 3D classification and applied to the entire CT scans. To verify the decision process of the network, LRP was applied.

The 3D ResNet outperformed both the 3D VGG an the 3D AlexNet. The network proved robust performance against metal artifacts and stents. The LRP showed that the network correctly focused on the aorta for the decision process. Across the whole data set our algorithm achieved an accuracy of 0.856 and AUC of 0.926. We presented a deep learning based solution for AAA screening in CE-CT. The algorithm proved to be robust and showed high performance even on a heterogeneous data set. Using LRP, relevance maps can be generated to make the decision process interpretable for the physician. Including automatic AAA screening into the clinical routine could enable earlier detection and therefore better patient care.

9. Outlook

A major hurdle in the development of DL methods remains the necessity of large annotated data sets. The proposed pipeline is able to overcome some limitation and train even on small data sets. A promising solution to further bypass this problem is the training on simulated data. This has been shown for artifact correction in Chapter 4. In a follow-up project, the generation of synthetic data using cycle Generative Adversarial Networks (GANs) for segmentation has proven successful [90, 109]. This approach has been extended to provide multimodal ground truth data for registration and segmentation [163]. Future research should further investigate which characteristics of simulated data are essential and how generalization to real data can be enhanced.

The inherent class imbalance in segmentation task can significantly decrease the performance of a DL solutions. In Chapter 6, ratio-based sample mining was proposed to solve this issue. This method does not require any adjustments for other datasets, as it only requires default segmentation markers and setting the ratio parameter. The application to liver [123, 164] and kidney segmentation [165] has already been investigated. It is especially promising for highly imbalanced problems such as lesion segmentation. Initial results on the application to MS lesion segmentation have already been presented [166] and are being further investigated. The next step, which is especially important to facilitate the M^{2}OLIE workflow, will be the multimodal segmentation of liver lesions as soon as the first patient cohort has undergone imaging.

To enable clinical usage of the proposed methods, integration of the algorithms into the clinical infrastructure is required. Deep VGM, which was presented in Chapter 5, was already integrated into the clinical image analysis software mTRIAL [94]. Therefore a working prototype is available to use in clinical trials and diagnostics. The screening method proposed in Chapter 4 was designed to run as a background process. To enable clinical usage, the next step will be to link the algorithm to the PACS and automate the anchor point annotation. The classification results could then be used to directly notify the physician when an AAA is detected in a CT scan. The vessel segmentation method from Chapter 6 as well as the artifact correction method from Chapter 4 have not yet been integrated into a clinical systems. Their integration is required to include these approaches into the M^{2}OLIE workflow and should therefore be a priority in the subsequent development.

A commonly voiced criticism to DL is that the decisions cannot be explained. This is also referred to as the black box problem. The lack of interpretability could be a critical factor for the acceptance of AI in clinical practice. In Chapter 7, the use of LRP to verify CNN decisions was explored. This technology computes relevance maps, that can be intuitively interpreted using simple visualization techniques. LRP is not limited to classification. It could also be used to optimize imaging protocols. Differences in relevance between MRI contrasts for MS lesion segmentation have already been shown [150]. Further exploration of this method and related approaches for interpretable AI is key to the clinical application of DL methods.

The training of CNNs requires an immense amount of computational effort and can take several days. However, a major advantage of these networks methods is the fast computation time for inference. This is not only an advantage of newly developed solutions, but can also be used to optimize computation time for existing methods as has been shown for Deep VGM in Chapter 5. Application time for the algorithms presented in the different studies are all in the range of a few seconds. While inference is already fast, it could potentially be shortened further by optimization of the application pipeline.

The training pipeline proposed in this thesis has already been adapted to binary classification, semantic segmentation and image regression. For image processing tasks falling into these categories, prototypes can be developed in a short time provided that a sufficiently annotated data set is available. While application for classification was only shown for a binary problem, multi-class segmentation is already implemented. The classification interface could also be further extended to multi-label classification. As most segmentation problems in the medical domain fall into the category of semantic segmentation, the segmentation interface was so far only implemented to solve tasks of this category. Extension to instance segmentation would be possible, but requires the implementation of a multi-step process, as DL solutions for instance segmentation require the use of several networks. The current implementation for image regression can be applied to a wide range of tasks. For some, the implementation of more specialized architectures may be required, but this can be done quickly due to the modular structure. Adaptation of the pipeline to single value regression is also possible with minimal effort.

10. Bibliography

[1] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. van der Laak, B. Van Ginneken, and C. I. Sánchez, "A survey on deep learning in medical image analysis," *Medical Image Analysis*, vol. 42, pp. 60–88, 2017. (cited on Page 1, 27, and 81)

[2] OECD, *Health at a Glance 2019.* 2019. (cited on Page 1)

[3] M. R. Oliva and S. Saini, "Liver cancer imaging: role of ct, mri, us and pet," *Cancer Imaging*, vol. 4, pp. S42–S46, 2004. (cited on Page 1)

[4] J. Yoon, L. Drumright, and M. van der Schaar, "Anonymization through data synthesis using generative adversarial networks (ads-gan)," *IEEE Journal of Biomedical and Health Informatics*, vol. 24, pp. 2378–2388, 2020. (cited on Page 1)

[5] T. Buzug, *Computed Tomography: From Photon Statistics to Modern Cone-Beam CT.* Springer Berlin Heidelberg, 2008. (cited on Page 6)

[6] A. Katsevich, "An improved exact filtered backprojection algorithm for spiral computed tomography," *Advances in Applied Mathematics*, vol. 32, no. 4, pp. 681–697, 2004. (cited on Page 6)

[7] G. N. Hounsfield, "Computerized transverse axial scanning (tomography): Part 1. description of system," *The British Journal of Radiology*, vol. 46, no. 552, pp. 1016–1022, 1973. (cited on Page 7)

[8] P. Lauterbur, "Image formation by induced local interactions," *Nature*, vol. 242, no. 5394, pp. 190–191, 1973. (cited on Page 7)

[9] M. Hossin and M. Sulaiman, "A review on evaluation metrics for data classification evaluations," *International Journal of Data Mining & Knowledge Management Process*, vol. 5, no. 2, pp. 1–11, 2015. (cited on Page 9 and 74)

[10] A. A. Taha and A. Hanbury, "Metrics for evaluating 3d medical image segmentation: analysis, selection, and tool," *BMC Medical Imaging*, vol. 15, pp. 1–28, 2015. (cited on Page 10)

[11] L. Zhang, L. Zhang, X. Mou, and D. Zhang, "Fsim: A feature similarity index for image quality assessment," *IEEE Transactions on Image Processing*, vol. 20, no. 8, pp. 2378–2386, 2011. (cited on Page 11)

[12] E. Kumar, *Artificial Intelligence.* I.K. International Publishing House Pvt. Limited, 2008. (cited on Page 11)

[13] T. Mitchell, *Machine Learning.* McGraw-Hill International Editions, McGraw-Hill, 1997. (cited on Page 12)

[14] L. Deng and D. Yu, "Deep learning: Methods and applications," Tech. Rep. MSR-TR-2014-21, Microsoft, May 2014. (cited on Page 12 and 15)

[15] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, “Backpropagation applied to handwritten zip code recognition,” *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989. (cited on Page 13)

[16] D. H. Hubel and T. N. Wiesel, “Receptive fields and functional architecture of monkey striate cortex,” *The Journal of Physiology*, vol. 195, no. 1, pp. 215–243, 1968. (cited on Page 13)

[17] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. (cited on Page 13)

[18] K. Yamaguchi, K. Sakamoto, T. Akabane, and Y. Fujimoto, “A neural network for speaker-independent isolated word recognition,” in *First International Conference on Spoken Language Processing (ICSLP 90)*, pp. 1077–1080, 1990. (cited on Page 15)

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *NIPS 2012* (F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds.), vol. 25, pp. 1097–1105, Curran Associates, Inc., 2012. (cited on Page 15, 72, and 104)

[20] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *ICLR 2015*, 2015. (cited on Page 15, 72, and 104)

[21] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, IEEE, 2016. (cited on Page 15, 32, 72, and 104)

[22] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3431–3440, IEEE, 2015. (cited on Page 15)

[23] V. Badrinarayanan, A. Kendall, and R. Cipolla, “Segnet: A deep convolutional encoder-decoder architecture for image segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017. (cited on Page 15)

[24] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *Proceedings of the International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*, pp. 234–241, Springer, 2015. (cited on Page 16, 28, 31, 46, and 57)

[25] F. Milletari, N. Navab, and S.-A. Ahmadi, “V-net: Fully convolutional neural networks for volumetric medical image segmentation,” in *Proceedings of the International Conference on 3D Vision (3DV)*, pp. 565–571, 2016. (cited on Page 16, 18, and 57)

[26] Ö. Çiçek, A. Abdulkadir, S. Lienkamp, T. Brox, and O. Ronneberger, "3d u-net: Learning dense volumetric segmentation from sparse annotation," in *Proceedings of the International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*, vol. 9901 of *LNCS*, pp. 424–432, Springer, 2016. (cited on Page 16 and 46)

[27] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proceedings of the International Conference on Neural Information Processing Systems*, vol. 27, pp. 2672–2680, Curran Associates, Inc., 2014. (cited on Page 16)

[28] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the International Conference on Computer Vision (ICCV)*, pp. 2223–2232, IEEE, 2017. (cited on Page 16)

[29] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, pp. 1–48, 2019. (cited on Page 17)

[30] F. Isensee, P. F. Jaeger, S. A. A. Kohl, J. Petersen, and K. H. Maier-Hein, "nnu-net: a self-configuring method for deep learning-based biomedical image segmentation," *Nature Methods*, vol. 18, no. 2, pp. 203–211, 2020. (cited on Page 18, 54, and 55)

[31] N. Qian, "On the momentum term in gradient descent learning algorithms," *Neural networks*, vol. 121, pp. 145–151, 1999. (cited on Page 19)

[32] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, no. 7, pp. 2121–2159, 2011. (cited on Page 19)

[33] M. D. Zeiler, "Adadelta: An adaptive learning rate method," *CoRR*, vol. abs/1212.5701, 2012. (cited on Page 20)

[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. (cited on Page 20)

[35] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014. (cited on Page 21)

[36] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the International Conference on Machine Learning*, vol. 37 of *PMLR*, pp. 448–456, 07–09 Jul 2015. (cited on Page 21 and 105)

[37] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg,

D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. Software available from tensorflow.org. (cited on Page 23, 81, and 105)

[38] B. C. Lowekamp, D. T. Chen, L. Ibáñez, and D. Blezek, "The design of simpleitk," *Frontiers in Neuroinformatics*, vol. 7, p. 45, 2013. (cited on Page 23, 81, and 105)

[39] P. Engel-Hills, "Radiation protection in medical imaging," *Radiography*, vol. 12, pp. 153–160, 2006. (cited on Page 27)

[40] A. G. Farman, "ALARA still applies," *Oral Surgery, Oral Medicine, Oral Pathology, Oral Radiology and Endodontology*, vol. 100, pp. 395–397, 2005. Editorial. (cited on Page)

[41] M. L. D. Gunn and J. R. Kohr, "State of the art: Technologies for computed tomography dose reduction," *Emergency Radiology*, vol. 17, pp. 209–218, 2010. (cited on Page)

[42] S. Vaegler, D. Stsepankou, J. Hesser, and O. Sauer, "Incorporation of local dependent reliability information into the Prior Image Constrained Compressed Sensing (PICCS) reconstruction algorithm," *Zeitschrift für medizinische Physik*, vol. 25, pp. 375–390, 2015. (cited on Page 27)

[43] J. D. Pack, F. Noo, and H. Kudo, "Investigation of saddle trajectories for cardiac ct imaging in cone-beam geometry," *Physics in Medicine and Biology*, vol. 49, pp. 2317–2336, 2004. (cited on Page 27)

[44] G. J. Gang, J. W. Stayman, T. Ehtiati, and J. H. Siewerdsen, "Task-driven image acquisition and reconstruction in cone-beam CT," *Physics in Medicine and Biology*, vol. 60, pp. 3129–3150, 2015. (cited on Page 27)

[45] K. Chung, L. R. Schad, and F. G. Zöllner, "Tomosynthesis implementation with adaptive online calibration on clinical c-arm systems," *International Journal of Computer Assisted Radiology and Surgery*, vol. 13, no. 10, pp. 1481–1495, 2018. (cited on Page 27 and 29)

[46] B. E. Claus, D. A. Langan, O. Al Assad, and X. Wang, "Circular tomosynthesis for neuro perfusion imaging on an interventional C-arm," in *Medical Imaging: Physics of Medical Imaging*, vol. 9412, p. 94122A, International Society for Optics and Photonics, 2015. (cited on Page 27)

[47] D. A. Langan, B. E. H. Claus, O. Al Assad, Y. Trousset, C. Riddell, G. Avignon, S. B. Solomon, H. Lai, and X. Wang, "Interventional C-arm tomosynthesis for vascular imaging: initial results," in *Medical Imaging: Physics of Medical Imaging*, p. 94125N, International Society for Optics and Photonics, 2015. (cited on Page)

[48] F. Xu, L. Helfen, T. Baumbach, and H. Suhonen, “Comparison of image quality in computed laminography and tomography,” *Optics Express*, pp. 794–806, 2012. (cited on Page 27)

[49] L. Borg, J. Frikel, J. S. Jørgensen, and E. T. Quinto, “Analyzing reconstruction artifacts from arbitrary incomplete x-ray ct data,” *SIAM Journal on Imaging Sciences*, vol. 11, no. 4, pp. 2786–2814, 2018. (cited on Page 27)

[50] C. R. Vogel and M. E. Oman, “Iterative Methods for Total Variation Denoising,” *SIAM Journal on Scientific Computing*, vol. 17, pp. 227–238, 1996. (cited on Page 27)

[51] Z. Tian, X. Jia, K. Yuan, T. Pan, and S. B. Jiang, “Low-dose CT reconstruction via edge-preserving total variation regularization,” *Physics in Medicine and Biology*, vol. 56, pp. 5949–5967, 2011. (cited on Page)

[52] E. Y. Sidky and X. Pan, “Image reconstruction in circular cone-beam computed tomography by constrained, total-variation minimization,” *Physics in Medicine and Biology*, vol. 53, pp. 4777–4807, 2008. (cited on Page 27)

[53] L. Ritschl, F. Bergner, C. Fleischmann, and M. Kachelrieß, “Improved total variation-based CT image reconstruction applied to clinical data,” *Physics in Medicine and Biology*, vol. 56, no. 6, p. 1545, 2011. (cited on Page 27)

[54] Y. Huang, T. Würfl, K. Breininger, L. Liu, G. Lauritsch, and A. Maier, “Some investigations on robustness of deep learning in limited angle tomography,” in *Proceedings of the International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*, pp. 145–153, Springer, 2018. (cited on Page 27 and 28)

[55] Z. Chen, X. Jin, L. Li, and G. Wang, “A limited-angle CT reconstruction method based on anisotropic TV minimization,” *Physics in Medicine and Biology*, vol. 58, no. 7, p. 2119, 2013. (cited on Page 27)

[56] G.-H. Chen, J. Tang, and S. Leng, “Prior image constrained compressed sensing (PICCS): a method to accurately reconstruct dynamic CT images from highly undersampled projection data sets,” *Medical Physics*, vol. 35, no. 2, pp. 660–663, 2008. (cited on Page 27)

[57] A. S. Lundervold and A. Lundervold, “An overview of deep learning in medical imaging focusing on mri,” *Zeitschrift für Medizinische Physik*, vol. 29, no. 2, pp. 102–127, 2019. (cited on Page 27, 45, and 71)

[58] C. H. McCollough, A. C. Bartley, R. E. Carter, B. Chen, T. A. Drees, P. Edwards, D. R. Holmes III, A. E. Huang, F. Khan, S. Leng, K. L. McMillan, G. J. Michalak, K. M. Nunez, L. Yu, and J. G. Fletcher, “Low-dose CT for the detection and classification of metastatic liver lesions: Results of the 2016 Low Dose CT Grand Challenge,” *Medical Physics*, vol. 44, no. 10, pp. e339–e352, 2017. (cited on Page 28)

[59] H. Zhang, L. Li, K. Qiao, L. Wang, B. Yan, L. Li, and G. Hu, "Image prediction for limited-angle tomography via deep learning with convolutional neural network," *arXiv preprint arXiv:1607.08707*, 2016. (cited on Page 28)

[60] Y. Han, J. J. Yoo, and J. C. Ye, "Deep Residual Learning for Compressed Sensing CT Reconstruction via Persistent Homology Analysis," *CoRR*, vol. abs/1611.06391, 2016. (cited on Page 28)

[61] Y. Han and J. C. Ye, "Deep residual learning approach for sparse-view CT reconstruction," in *Proceedings of the International Conference on Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine (Fully3D)*, 2017. (cited on Page 28)

[62] Y. Han and J. C. Ye, "Framing U-Net via Deep Convolutional Framelets: Application to Sparse-View CT," *IEEE Transactions on Medical Imaging*, vol. 37, pp. 1418–1429, 2018. (cited on Page 28)

[63] D. Lee, J. Yoo, and J. C. Ye, "Deep residual learning for compressed sensing MRI," in *International Symposium on Biomedical Imaging (ISBI)*, pp. 15–18, IEEE, 2017. (cited on Page 28)

[64] A.-K. Schnurr, K. Chung, L. R. Schad, and F. G. Zöllner, "Abstract: Deep residual learning for limited angle artefact correction," in *Proceedings of Bildverarbeitung für die Medizin (BVM)*, pp. 280–280, Springer, 2018. (cited on Page 28)

[65] S. Xu, P. Prinsen, J. Wiegert, and R. Manjeshwar, "Deep residual learning in ct physics: scatter correction for spectral ct," *CoRR*, vol. abs/1708.04151, 2017. (cited on Page 28)

[66] J. Maier, Y. Berker, S. Sawall, and M. Kachelrieß, "Deep scatter estimation (DSE): feasibility of using a deep convolutional neural network for real-time x-ray scatter prediction in cone-beam CT," in *Medical Imaging: Physics of Medical Imaging* (J. Y. Lo, T. G. Schmidt, and G.-H. Chen, eds.), vol. 10573, pp. 393–398, International Society for Optics and Photonics, 2018. (cited on Page 28)

[67] S. J. Wirkert, H. Kenngott, B. Mayer, P. Mietkowski, M. Wagner, P. Sauer, N. T. Clancy, D. S. Elson, and L. Maier-Hein, "Robust near real-time estimation of physiological parameters from megapixel multispectral images with inverse Monte Carlo and random forest regression," *International Journal of Computer Assisted Radiology and Surgery*, vol. 11, no. 6, pp. 909–917, 2016. (cited on Page 28)

[68] Y. Zhang and H. Yu, "Convolutional Neural Network Based Metal Artifact Reduction in X-Ray Computed Tomography," *IEEE Transactions on Medical Imaging*, vol. 37, no. 6, pp. 1370–1381, 2018. (cited on Page 28)

[69] W. P. Segars, G. Sturgeon, S. Mendonca, J. Grimes, and B. M. W. Tsui, "4D XCAT phantom for multimodality imaging research," *Medical Physics*, vol. 37, no. 9, pp. 4902–4915, 2010. (cited on Page 30)

[70] W. van Aarle, W. J. Palenstijn, J. de Beenhouwer, T. Altantzis, S. Bals, K. J. Batenburg, and J. Sijbers, "The ASTRA Toolbox: A platform for advanced algorithm development in electron tomography," *Ultramicroscopy*, vol. 157, pp. 35–47, 2015. (cited on Page 31)

[71] W. van Aarle, W. J. Palenstijn, J. Cant, E. Janssens, F. Bleichrodt, A. Dabravolski, J. D. Beenhouwer, K. J. Batenburg, and J. Sijbers, "Fast and flexible X-ray tomography using the ASTRA toolbox," *Opt. Express*, vol. 24, no. 22, pp. 25129–25147, 2016. (cited on Page 31)

[72] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 807–814, 2010. (cited on Page 31)

[73] S. Wen, W. Liu, Y. Yang, T. Huang, and Z. Zeng, "Generating realistic videos from keyframes with concatenated gans," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 8, pp. 2337–2348, 2019. (cited on Page 33)

[74] P. T. Lauzier, J. Tang, and G.-H. Chen, "Prior image constrained compressed sensing: Implementation and performance evaluation," *Medical Physics*, vol. 39, no. 1, pp. 66–80, 2012. (cited on Page 34)

[75] J. Hatton, B. McCurdy, and P. B. Greer, "Cone beam computerized tomography," *Physics in Medicine and Biology*, vol. 54, pp. N329–46, 2009. (cited on Page 42)

[76] A. Richter, Q. Hu, D. Steglich, K. Baier, J. Wilbert, M. Guckenberger, and M. Flentje, "Investigation of the usability of conebeam CT data sets for dose calculation," *Radiation Oncology*, vol. 3, no. 1, pp. 1–13, 2008. (cited on Page 42)

[77] U. W. Kaunzner and S. A. Gauthier, "Mri in the assessment and monitoring of multiple sclerosis: an update on best practice," *Therapeutic Advances in Neurological Disorders*, vol. 10, no. 6, pp. 247–261, 2017. (cited on Page 45)

[78] J. Fox, M. Kraemer, T. Schormann, A. Dabringhaus, J. Hirsch, P. Eisele, K. Szabo, C. Weiss, M. Amann, K. Weier, *et al.*, "Individual assessment of brain tissue changes in ms and the effect of focal lesions on short-term focal atrophy development in ms: a voxel-guided morphometry study," *Int. Journal of Molecular Sciences*, vol. 17, no. 4, p. 489, 2016. (cited on Page 45 and 46)

[79] X. Lladó, O. Ganiler, A. Oliver, R. Martí, J. Freixenet, L. Valls, J. C. Vilanova, L. Ramió-Torrentà, and A. Rovira, "Automated detection of multiple sclerosis lesions in serial brain mri," *Neuroradiology*, vol. 54, no. 8, pp. 787——807, 2012. (cited on Page 45)

[80] N. Patel, M. Horsfield, C. Banahan, A. Thomas, M. Nath, J. Nath, P. Ambrosi, and E. Chung, "Detection of focal longitudinal changes in the brain by subtraction of mr images," *American Journal of Neuroradiology*, vol. 38, no. 5, pp. 923–927, 2017. (cited on Page 45)

[81] H. J. Seo and P. Milanfar, “A non-parametric approach to automatic change detection in mri images of the brain,” in *International Symposium on Biomedical Imaging (ISBI)*, pp. 245–248, IEEE, 2009. (cited on Page 45)

[82] T. Schormann and M. Kraemer, “Voxel-guided morphometry (”vgm”) and application to stroke,” *IEEE Transactions on Medical Imaging*, vol. 22, no. 1, pp. 62–74, 2003. (cited on Page 45 and 46)

[83] A. Tousignant, P. Lemaître, D. Precup, D. L. Arnold, and T. Arbel, “Prediction of disease progression in multiple sclerosis patients using deep learning analysis of mri data,” in *Proceedings of the International Conference on Medical Imaging with Deep Learning (MIDL)*, vol. 102 of *PMLR*, pp. 483–492, 2019. (cited on Page 45)

[84] N. M. Sepahvand, D. L. Arnold, and T. Arbel, “Cnn detection of new and enlarging multiple sclerosis lesions from longitudinal mri using subtraction images,” in *International Symposium on Biomedical Imaging (ISBI)*, pp. 127–130, IEEE, 2020. (cited on Page 45)

[85] C. Polman, S. Reingold, B. Banwell, M. Clanet, J. Cohen, M. Filippi, K. Fujihara, E. Havrdova, M. Hutchinson, L. Kappos, F. Lublin, X. Montalban, P. O’Connor, M. Sandberg-Wollheim, A. Thompson, E. Waubant, B. Weinshenker, and J. Wolinsky, “Diagnostic criteria for multiple sclerosis: 2010 revisions to the mcdonald criteria,” *Annals of Neurology*, vol. 69, no. 2, pp. 292–302, 2011. (cited on Page 45)

[86] F. Segonne, A. Dale, E. Busa, M. Glessner, D. Salat, H. Hahn, and B. Fischl, “A hybrid approach to the skull stripping problem in mri,” *Neuroimage*, vol. 22, pp. 1060–1075, 2004. (cited on Page 46)

[87] E. B. Lewis and N. C. Fox, “Correction of differential intensity inhomogeneity in longitudinal mr images,” *Neuroimage*, vol. 23, no. 1, pp. 75–83, 2004. (cited on Page 46)

[88] M. Kraemer, T. Schormann, G. Hagemann, B. Qi, O. W. Witte, and R. J. Seitz, “Delayed shrinkage of the brain after ischemic stroke: preliminary observations with voxel-guided morphometry,” *Journal of Neuroimaging*, vol. 14, no. 3, pp. 265–272, 2004. (cited on Page 46)

[89] M. Kraemer, T. Schormann, A. Dabringhaus, J. Hirsch, K. Stephan, V. Hömberg, L. Kappos, and A. Gass, “Individual assessment of chronic brain tissue changes in mri–the role of focal lesions for brain atrophy development. a voxel-guided morphometry study,” *Klinische Neurophysiologie*, vol. 39, no. 01, p. A178, 2008. (cited on Page 46)

[90] D. F. Bauer, A.-K. Schnurr, T. Russ, S. Goerttler, L. R. Schad, F. G. Zoellner, and K. Chung, “Synthesis of ct images using cyclegans: Enhancement of anatomical accuracy,” in *Proceedings of the International Conference on Medical Imaging with Deep Learning (MIDL)*, (London, United Kingdom), 2019. (cited on Page 48 and 85)

[91] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004. (cited on Page 49)

[92] T. J. Sørensen, *A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on Danish commons.* I kommission hos E. Munksgaard, 1948. (cited on Page 49)

[93] L. R. Dice, “Measures of the amount of ecologic association between species,” *Ecology*, vol. 26, no. 3, pp. 297–302, 1945. (cited on Page 49)

[94] J. Gregori, C. Cornelissen, S. Hoffmann, M. Treiber, S. Randoll, S. Heldmann, J. Klein, R. Opfer, L. Spies, A. Gass, T. Ziemsen, H. Kitzler, and F. Weiler, “Feasibility of fully automated atrophy measurement of the upper cervical spinal cord for group analyses and patient-individual diagnosis support in ms,” in *Proceedings of the Congress of the European Committee for Treatment and Research in Multiple Sclerosis*, p. P1120, 2018. (cited on Page 51 and 85)

[95] J. Hernández-Rivera, R. Espinoza-Pérez, J. Cancino-López, R. Silva-Rueda, M. Salazar-Mendoza, and R. Paniagua-Sierra, “Anatomical variants in renal transplantation, surgical management, and impact on graft functionality,” *Transplantation Proceedings*, vol. 50, no. 10, pp. 3216–3221, 2018. (cited on Page 53)

[96] E. Goceri, “Automatic labeling of portal and hepatic veins from MR images prior to liver transplantation,” *International Journal of Computer Assisted Radiology and Surgery*, vol. 11, no. 12, pp. 2153–2161, 2016. (cited on Page 53)

[97] H. Fu, Y. Xu, S. Lin, D. W. Kee Wong, and J. Liu, “DeepVessel: Retinal vessel segmentation via deep learning and conditional random field,” in *Proceedings of the International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*, pp. 132–139, Springer, 2016. (cited on Page 53 and 54)

[98] T. M. Downing, S. N. Khan, R. C. Zvavanjanja, Z. Bhatti, A. K. Pillai, and S. T. Kee, “Portal venous interventions: How to recognize, avoid, or get out of trouble in transjugular intrahepatic portosystemic shunt (TIPS), balloon occlusion sclerosis (ie, BRTO), and portal vein embolization (PVE),” *Techniques in Vascular and Interventional Radiology*, vol. 21, no. 4, pp. 267–287, 2018. (cited on Page 53)

[99] S. Moccia, E. D. Momi, S. E. Hadji, and L. S. Mattos, “Blood vessel segmentation algorithms - review of methods, datasets and evaluation metrics,” *Comput. Methods Programs Biomed.*, vol. 158, pp. 71–91, 2018. (cited on Page 53)

[100] A. A. Novikov, D. Major, M. Wimmer, G. Sluiter, and K. Bühler, “Automated Anatomy-Based Tracking of Systemic Arteries in Arbitrary Field-of-View CTA

Scans," *IEEE Transactions on Medical Imaging*, vol. 36, no. 6, pp. 1359–1371, 2017. (cited on Page 53)

[101] J. V. B. Soares, J. J. G. Leandro, R. M. Cesar, H. F. Jelinek, and M. J. Cree, "Retinal vessel segmentation using the 2-D gabor wavelet and supervised classification," *IEEE Transactions on Medical Imaging*, vol. 25, no. 9, pp. 1214–1222, 2006. (cited on Page 53)

[102] L. Wang, A.-K. Schnurr, S. Zidowitz, J. Georgii, Y. Zhao, M. Razavi, M. Schwier, H. K. Hahn, and C. Hansen, "Segmentation of hepatic artery in multi-phase liver CT using directional dilation and connectivity analysis," in *Medical Imaging: Computer-Aided Diagnosis*, vol. 9785, pp. 436–443, International Society for Optics and Photonics, 2016. (cited on Page 53)

[103] C. Xu, D. Pham, and J. Prince, "Image segmentation using deformable models," in *Handbook of Medical Imaging, Volume 2: Medical Image Processing and Analysis*, pp. 129–174, International Society for Optics and Photonics, 2010. (cited on Page 54)

[104] O. Friman, M. Hindennach, C. Kühnel, and H.-O. Peitgen, "Multiple hypothesis template tracking of small 3D vessel structures," *Medical Image Analysis*, vol. 14, no. 2, pp. 160–171, 2010. (cited on Page 54)

[105] A.-K. Schnurr, K. Chung, T. Russ, L. R. Schad, and F. G. Zöllner, "Simulation-based deep artifact correction with convolutional neural networks for limited angle artifacts," *Zeitschrift für medizinische Physik*, vol. 29, no. 2, pp. 150–161, 2019. (cited on Page 54)

[106] G. Tetteh, M. Rempfler, C. Zimmer, and B. H. Menze, "Deep-fext: Deep feature extraction for vessel segmentation and centerline prediction," in *Proceedings of the International Workshop on Machine Learning in Medical Imaging (MLMI)*, pp. 344–352, 2017. (cited on Page 54 and 55)

[107] G. Tetteh, V. Efremov, N. D. Forkert, M. Schneider, J. Kirschke, B. Weber, C. Zimmer, M. Piraud, and B. H. Menze, "DeepVesselNet: Vessel segmentation, centerline prediction, and bifurcation detection in 3-D angiographic volumes," *arXiv:1803.09340v3*, 2018. (cited on Page 54, 55, and 57)

[108] T. Kitrungrotsakul, X.-H. Han, X. Wei, and Y.-W. Chen, "Multi-pathways cnn for robust vascular segmentation," in *Medical Imaging: Biomedical Applications in Molecular, Structural, and Functional Imaging*, vol. 10578, pp. 426–431, International Society for Optics and Photonics, 2018. (cited on Page 54 and 55)

[109] T. Russ, S. Goerttler, A.-K. Schnurr, D. F. Bauer, S. Hatamikia, L. R. Schad, F. G. Zöllner, and K. Chung, "Synthesis of CT images from digital body phantoms using CycleGAN," *International Journal of Computer Assisted Radiology and Surgery*, vol. 14, no. 10, pp. 1741–1750, 2019. (cited on Page 54, 55, and 85)

[110] M. Oda, H. R. Roth, T. Kitasaka, K. Misawa, M. Fujiwara, and K. Mori, "Abdominal artery segmentation method from CT volumes using fully convolutional neural network," *International Journal of Computer Assisted Radiology and Surgery*, vol. 14, no. 12, pp. 2069–2081, 2019. (cited on Page 54, 55, and 56)

[111] P. Nardelli, D. Jimenez-Carretero, D. Bermejo-Pelaez, G. R. Washko, F. N. Rahaghi, M. J. Ledesma-Carbayo, and R. San José Estépar, "Pulmonary artery–vein classification in CT images using deep learning," *IEEE Transactions on Medical Imaging*, vol. 37, no. 11, pp. 2428–2440, 2018. (cited on Page 54)

[112] N. Li, S. Zhou, Z. Wu, B. Zhang, and G. Zhao, "Statistical modeling and knowledge-based segmentation of cerebral artery based on tof-mra and mr-t1," *Computer Methods and Programs in Biomedicine*, vol. 186, p. 105110, 2020. (cited on Page 54)

[113] H. Zheng, Y. Zhang, L. Yang, P. Liang, Z. Zhao, C. Wang, and D. Z. Chen, "A new ensemble learning framework for 3d biomedical image segmentation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 5909–5916, 2019. (cited on Page 54)

[114] Z.-H. Zhou, *Ensemble methods: foundations and algorithms.* CRC press, 2012. (cited on Page 54)

[115] S. Winzeck, S. Mocking, R. Bezerra, M. Bouts, E. McIntosh, I. Diwan, P. Garg, A. Chutinet, W. Kimberly, W. Copen, P. Schaefer, H. Ay, A. Singhal, K. Kamnitsas, B. Glocker, A. Sorensen, and O. Wu, "Ensemble of convolutional neural networks improves automated segmentation of acute ischemic lesions using multiparametric diffusion-weighted mri," *American Journal of Neuroradiology*, vol. 40, no. 6, pp. 938–945, 2019. (cited on Page 54 and 67)

[116] K. Kamnitsas, W. Bai, E. Ferrante, S. McDonagh, M. Sinclair, N. Pawlowski, M. Rajchl, M. Lee, B. Kainz, D. Rueckert, and B. Glocker, "Ensembles of multiple models and architectures for robust brain tumour segmentation," in *Proceedings of the International Workshop Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, pp. 450–462, 2018. (cited on Page 54)

[117] L. Chen, Y. Xie, J. Sun, N. Balu, M. Mossa-Basha, K. Pimentel, T. S. Hatsukami, J.-N. Hwang, and C. Yuan, "Y-Net: 3D intracranial artery segmentation using a convolutional autoencoder," *arXiv:1712.07194*, 2017. (cited on Page 55)

[118] A. L. Simpson, M. Antonelli, S. Bakas, M. Bilello, K. Farahani, B. van Ginneken, A. Kopp-Schneider, B. A. Landman, G. Litjens, B. Menze, *et al.*, "A large annotated medical image dataset for the development and evaluation of segmentation algorithms," *arXiv:1902.09063*, 2019. (cited on Page 55)

[119] W. Yu, B. Fang, Y. Liu, M. Gao, S. Zheng, and Y. Wang, "Liver vessels segmentation based on 3D residual U-Net," in *Proceedings of the International Conference on Image Processing*, pp. 250–254, IEEE, 2019. (cited on Page 55)

[120] L. Soler, A. Hostettler, V. Agnus, A. Charnoz, J. Fasquel, J. Moreau, A. Osswald, M. Bouhadjar, and J. Marescaux, "3D image reconstruction for comparison of algorithm database: A patient specific anatomical and medical image database," 2010. https://www.ircad.fr/fr/recherche/3d-ircadb-01-fr/. (cited on Page 55)

[121] B. Landman, Z. Xu, J. Igelsias, M. Styner, T. Langerak, and A. Klein, "MICCAI multi-atlas labeling beyond the cranial vault - Workshop and challenge," 2015. 10.7303/syn3193805. (cited on Page 55)

[122] E. Gibson, F. Giganti, Y. Hu, E. Bonmati, S. Bandula, K. Gurusamy, B. Davidson, S. P. Pereira, M. J. Clarkson, and D. C. Barratt, "Multi-organ abdominal ct reference standard segmentations," 2018. 10.5281/zenodo.1169361. (cited on Page 55)

[123] A.-K. Schnurr, L. R. Schad, and F. G. Zöllner, "Kann ein festes Klassenverhältnis im Training die CNN-Lebersegmentierung im CT verbessern?," in *Proceedings of the Conference on Image-Guided Interventions (IGIC)*, 2019. (cited on Page 56 and 85)

[124] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (ELUs)," *arXiv:1511.07289*, 2015. (cited on Page 57)

[125] B. Dashtbozorg, A. M. Mendonça, and A. Campilho, "An automatic graph-based approach for artery/vein classification in retinal images," *IEEE Transactions on Image Processing*, vol. 23, no. 3, pp. 1073–1083, 2014. (cited on Page 67)

[126] N. Rathmann, K. Kara, J. Budjan, T. Henzler, A. Smakic, S. O. Schönberg, and S. J. Diehl, "Parenchymal liver blood volume and dynamic volume perfusion CT measurements of hepatocellular carcinoma in patients undergoing transarterial chemoembolization," *Anticancer research*, vol. 37, no. 10, pp. 5681–5685, 2017. (cited on Page 68)

[127] P. K. Saha, Z. Gao, S. K. Alford, M. Sonka, and E. A. Hoffman, "Topomorphologic separation of fused isointensity objects via multiscale opening: Separating arteries and veins in 3-d pulmonary ct," *IEEE Transactions on Medical Imaging*, vol. 29, no. 3, pp. 840–851, 2010. (cited on Page 68)

[128] I. M. Nordon, R. J. Hinchliffe, I. M. Loftus, and M. M. Thompson, "Pathophysiology and epidemiology of abdominal aortic aneurysms," *Nat. Rev. Cardiol*, vol. 8, no. 2, p. 92, 2011. (cited on Page 71)

[129] E. L. Chaikof, R. L. Dalman, M. K. Eskandari, B. M. Jackson, W. A. Lee, M. A. Mansour, T. M. Mastracci, M. Mell, M. H. Murad, L. L. Nguyen, G. S. Oderich, M. S. Patel, M. L. Schermerhorn, and B. W. Starnes, “The society for vascular surgery practice guidelines on the care of patients with an abdominal aortic aneurysm,” *Journal of Vascular Surgery*, vol. 67, no. 1, pp. 2–77.e2, 2018. (cited on Page 71)

[130] E. Turton, D. Scott, M. Delbridge, S. Snowden, and R. Kester, “Ruptured abdominal aortic aneurysm: a novel method of outcome prediction using neural network technology,” *Eur. J. Vasc. Endovasc. Surg.*, vol. 19, no. 2, pp. 184–189, 2000. (cited on Page 71)

[131] T. Schmitz-Rixen, M. Keese, M. Hakimi, A. Peters, D. Böckler, K. Nelson, and R. Grundmann, “Ruptured abdominal aortic aneurysm—epidemiology, predisposing factors, and biology,” *Langenbeck’s Archives of Surgery*, vol. 401, no. 3, pp. 275–288, 2016. (cited on Page 71)

[132] A. S. Peters, M. Hakimi, P. Erhart, M. Keese, T. Schmitz-Rixen, M. Wortmann, M. S. Bischoff, and D. Böckler, “Current treatment strategies for ruptured abdominal aortic aneurysm,” *Langenbeck’s Archives of Surgery*, vol. 401, no. 3, pp. 289–298, 2016. (cited on Page)

[133] J. S. Lindholt, R. Søgaard, and J. Laustsen, “Prognosis of ruptured abdominal aortic aneurysms in denmark from 1994–2008,” *Clinical Epidemiology*, vol. 4, p. 111, 2012. (cited on Page 71)

[134] R. Claridge, S. Arnold, N. Morrison, and A. M. van Rij, “Measuring abdominal aortic diameters in routine abdominal computed tomography scans and implications for abdominal aortic aneurysm screening,” *Journal of Vascular Surgery*, vol. 65, no. 6, pp. 1637–1642, 2017. (cited on Page 71 and 78)

[135] C. Oliver-Williams, M. J. Sweeting, J. Jacomelli, L. Summers, A. Stevenson, T. Lees, and J. J. Earnshaw, “Safety of men with small and medium abdominal aortic aneurysms under surveillance in the naaasp,” *Circulation*, vol. 139, no. 11, pp. 1371–1380, 2019. (cited on Page 71)

[136] K. Salata, M. A. Hussain, C. de Mestral, E. Greco, B. A. Aljabri, M. Mamdani, T. L. Forbes, D. L. Bhatt, S. Verma, and M. Al-Omran, “Comparison of outcomes in elective endovascular aortic repair vs open surgical repair of abdominal aortic aneurysms,” *JAMA Network Open*, vol. 7, no. 2, pp. e196578–e196578, 2019. (cited on Page 71)

[137] S. Mohammadi, M. Mohammadi, V. Dehlaghi, and A. Ahmadi, “Automatic segmentation, detection, and diagnosis of abdominal aortic aneurysm (aaa) using convolutional neural networks and hough circles algorithm,” *Cardiovasc. Eng. Technol.*, vol. 10, no. 3, pp. 490–499, 2019. (cited on Page 71, 78, and 79)

[138] K. López-Linares, N. Aranjuelo, L. Kabongo, G. Maclair, N. Lete, M. Ceresa, A. García-Familiar, I. Macía, and M. A. González Ballester, “Fully automatic detection and segmentation of abdominal aortic thrombus in post-operative

cta images using deep convolutional neural networks," *Medical Image Analysis*, vol. 46, pp. 202–214, 2018. (cited on Page 71)

[139] M. Habijan, I. Galić, H. Leventić, K. Romić, and D. Babin, "Abdominal aortic aneurysm segmentation from ct images using modified 3d u-net with deep supervision," in *2020 International Symposium ELMAR*, pp. 123–128, 2020. (cited on Page 71)

[140] L. Zhang, Z. Jiang, J. Choi, C. Y. Lim, T. Maiti, and S. Baek, "Patient-specific prediction of abdominal aortic aneurysm expansion using bayesian calibration," *IEEE J Biomed Health Inform*, vol. 23, no. 6, pp. 2537–2550, 2019. (cited on Page 71)

[141] H. N. Do, A. Ijaz, H. Gharahi, B. Zambrano, J. Choi, W. Lee, and S. Baek, "Prediction of abdominal aortic aneurysm growth using dynamical gaussian process implicit surface," *IEEE Transactions on Biomedical Engineering*, vol. 66, no. 3, pp. 609–622, 2019. (cited on Page 71)

[142] G. García, J. Maiora, A. Tapia, and M. de Blas, "Evaluation of texture for classification of abdominal aortic aneurysm after endovascular repair," *Journal of Digital Imaging*, vol. 25, no. 3, pp. 369–376, 2012. (cited on Page 71)

[143] R. J. Harris, S. Kim, J. Lohr, S. Towey, Z. Velichkovich, T. Kabachenko, I. Driscoll, and B. Baker, "Classification of aortic dissection and rupture on post-contrast ct images using a convolutional neural network," *Journal of Digital Imaging*, vol. 32, no. 6, pp. 939–946, 2019. (cited on Page 71 and 78)

[144] L. Cao, R. Shi, Y. Ge, L. Xing, P. Zuo, Y. Jia, J. Liu, Y. He, X. Wang, S. Luan, X. Chai, and W. Guo, "Fully automatic segmentation of type b aortic dissection from cta images enabled by deep learning," *European Journal of Radiology*, vol. 121, p. 108713, 2019. (cited on Page 71)

[145] S. Hahn, M. Perry, S. Wshah, C. S. Morris, and D. J. Bertges, "Machine deep learning accurately detects endoleak after endovascular abdominal aortic aneurysm repair," *Journal of Vascular Surgery*, vol. 69, no. 6, pp. e202–e203, 2019. (cited on Page 72)

[146] H. A. Hong and U. U. Sheikh, "Automatic detection, segmentation and classification of abdominal aortic aneurysm using deep learning," in *Proceedings of the International Colloquium on Signal Processing and its Application*, pp. 242–246, IEEE, 2016. (cited on Page 72 and 78)

[147] J. Maiora and M. Graña, "A hybrid segmentation of abdominal ct images," in *Proceedings of the International Conference on Hybrid Artificial Intelligence Systems (HAIS)*, pp. 416–423, Springer, 2012. (cited on Page 72)

[148] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PLOS ONE*, vol. 10, no. 7, pp. 1–46, 2015. (cited on Page 74)

[149] G. Chlebus, N. Abolmaali, A. Schenk, and H. Meine, “Relevance analysis of mri sequences for automatic liver tumor segmentation,” in *Proceedings of the International Conference on Medical Imaging with Deep Learning (MIDL)*, 2019. (cited on Page 74)

[150] A.-K. Schnurr, M. Schöben, I. Hermann, R. Schmidt, G. Chlebus, L. R. Schad, and F. G. Zöllner, “Relevance analysis of mri sequences for ms lesion detection,” in *Proceedings of the ESMRMB Congress*, pp. 77–78, 2020. (cited on Page 74 and 85)

[151] P. Bains, J. L. Oliffe, M. H. Mackay, and M. T. Kelly, “Screening older adult men for abdominal aortic aneurysm: A scoping review,” *American Journal of Men’s Health*, vol. 15, no. 2, 2021. (cited on Page 78)

[152] M. Sweeting, J. Marshall, M. Glover, A. Nasim, and M. J. Bown, “Evaluating the cost-effectiveness of changes to the surveillance intervals in the uk abdominal aortic aneurysm screening programme.,” *Value Health*, vol. 24, no. 3, pp. 369–376, 2021. (cited on Page 78)

[153] K. Yasaka and O. Abe, “Deep learning and artificial intelligence in radiology: Current applications and future directions,” *PLOS Medicine*, vol. 15, no. 11, p. e1002707, 2018. (cited on Page 78)

[154] L. Saba, M. Biswas, V. Kuppili, E. Cuadrado Godia, H. S. Suri, D. R. Edla, T. Omerzu, J. R. Laird, N. N. Khanna, S. Mavrogeni, A. Protogerou, P. P. Sfikakis, V. Viswanathan, G. D. Kitas, A. Nicolaides, A. Gupta, and J. S. Suri, “The present and future of deep learning in radiology,” *European Journal of Radiology*, vol. 114, pp. 14–24, 2019. (cited on Page 78)

[155] F. Ghesu, B. Georgescu, Y. Zheng, S. Grbic, A. Maier, J. Hornegger, and D. Comaniciu, “Multi-scale deep reinforcement learning for real-time 3d-landmark detection in ct scans,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 1, pp. 176–189, 2019. (cited on Page 78)

[156] J. Raffort, C. Adam, M. Carrier, A. Ballaith, R. Coscas, E. Jean-Baptiste, R. Hassen-Khodja, N. Chakfé, and F. Lareyre, “Artificial intelligence in abdominal aortic aneurysm,” *Journal of Vascular Surgery*, vol. 72, no. 1, pp. 321–333.e1, 2020. (cited on Page 78)

[157] F. Lareyre, C. Adam, M. Carrier, C. Dommerc, C. Mialhe, and J. Raffort, “A fully automated pipeline for mining abdominal aortic aneurysm using image segmentation,” *Scientific Reports*, vol. 9, no. 1, p. 13750, 2019. (cited on Page 79)

[158] K. Hirata, T. Nakaura, M. Nakagawa, M. Kidoh, S. Oda, D. Utsunomiya, and Y. Yamashita, “Machine learning to predict the rapid growth of small abdominal aortic aneurysm,” *Journal of Computer Assisted Tomography*, vol. 44, no. 1, pp. 37–42, 2020. (cited on Page 78)

[159] H. Polat and H. Danaei Mehr, “Classification of pulmonary ct images by using hybrid 3d-deep convolutional neural network architecture,” *Applied Sciences*, vol. 9, no. 5, p. 940, 2019. (cited on Page 79)

[160] D. Singh, V. Kumar, and M. Kaur, “Classification of covid-19 patients from chest ct images using multi-objective differential evolution–based convolutional neural networks,” *European Journal of Clinical Microbiology and Infectious Diseases*, pp. 1–11, 2020. (cited on Page 79)

[161] A. K. Golla, D. F. Bauer, R. Schmidt, T. Russ, D. Nörenberg, K. Chung, C. Tönnes, L. R. Schad, and F. G. Zöllner, “Convolutional neural network ensemble segmentation with ratio-based sampling for the arteries and veins in abdominal ct scans,” *IEEE Transactions on Biomedical Engineering*, 2020. in press. (cited on Page 79)

[162] M. Alber, S. Lapuschkin, P. Seegerer, M. Hägele, K. T. Schütt, G. Montavon, W. Samek, K.-R. Müller, S. Dähne, and P.-J. Kindermans, “innvestigate neural networks!,” *Journal of Machine Learning Research*, vol. 20, no. 93, pp. 1–8, 2019. (cited on Page 83 and 105)

[163] D. F. Bauer, T. Russ, B. I. Waldkirch, W. P. Segars, L. R. Schad, F. G. Zöllner, and A.-K. Golla, “Generation of multimodal ground truth datasets for abdominal medical image registration using cyclegan,” *International Journal of Computer Assisted Radiology and Surgery*, 2021. in press. (cited on Page 85)

[164] A.-K. Schnurr, L. R. Schad, and F. G. Zöllner, “Sparsely connected convolutional layers in cnns for liver segmentation in ct,” in *Proceedings of Bildverarbeitung für die Medizin (BVM)*, pp. 80–85, Springer, 2019. (cited on Page 85)

[165] A.-K. Schnurr, C. Drees, L. R. Schad, and F. G. Zöllner, “Comparing sample mining schemes for cnn kidney segmentation in t1w mri,” in *Proceedings of the International Conference on Functional Renal Imaging*, 2019. (cited on Page 85)

[166] A.-K. Schnurr, I. Hermann, R. Schmidt, A. Gass, F. G. Zöllner, and L. R.Schad, “Fully convolutional neural network segmentation of multiple sclerosis lesions using t1 and t2* maps,” in *Proceedings of the ESMRMB Congress*, vol. 36, p. 312, 2019. (cited on Page 85)

11. Appendix

11.1 Supplementary Material Chapter 6

Table 11.1: Vascular Structures - The listed vessels were included in the artery and vein class annotations, if they were sufficiently visible.

Atery	Vein
Aorta ascendens	Vena cava inferior
Arcus aortae	Vena hepatica sinistra
Aorta descendens	Vena hepatica media
Aorta thoracica	Vena hepatica dextra
Aorta abdominalis	Vena portae
Truncus coeliacus	Vena splenica
Arteria mesenterica superior	Vena renalis sinistra
Arteria renalis sinistra	Vena renalis dextra
Arteria renalis dextra	Vena iliaca communis
Arteria splenica	
Arteria iliaca communis	

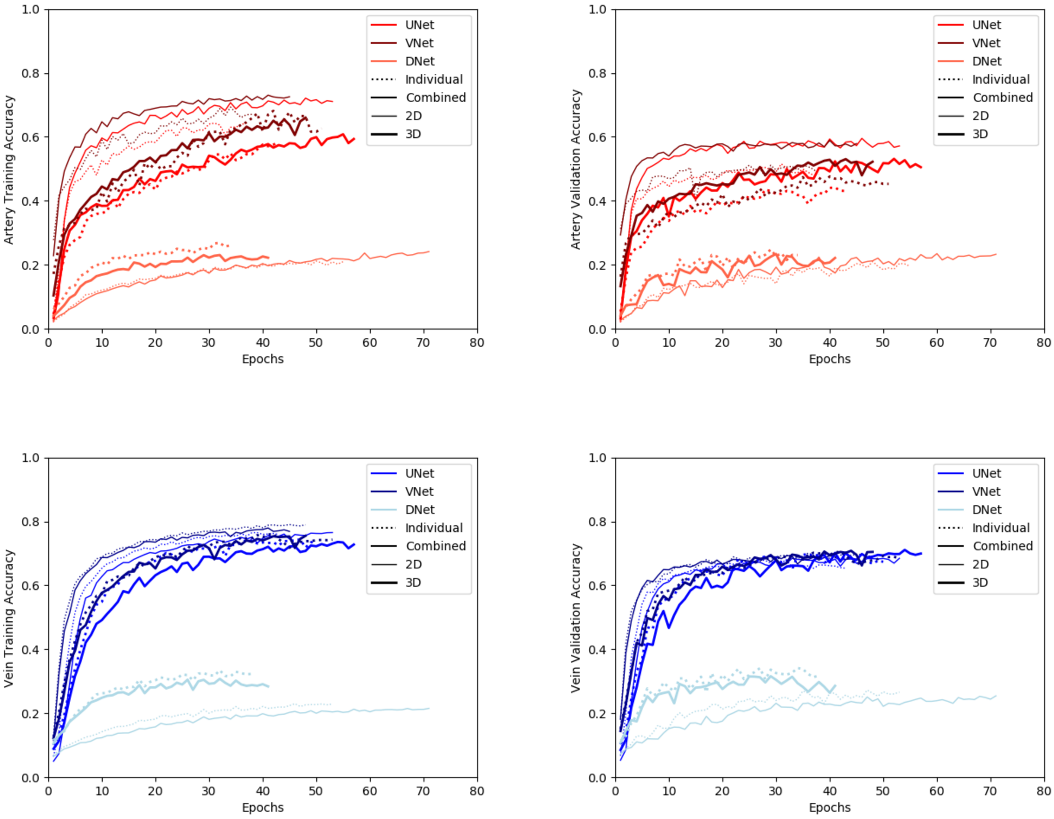


Figure 11.1: Average training and validation accuracy for all networks from Experiment 1.

11.2 Supplementary Material Chapter 7

Data Aggregation

Keywords used to find suitable cases in the the PACS were (translated): ”aortic aneurysm”, ”aneurysm”, ”abdominal aneurysm”, ”AAA”, ”no aneurysm”, ”no AAA” within the radiology report. This search strategy yielded a substantial number of reports confirming the presence of AAA and other reports denying presence of AAA.

Networks

AlexNet was proprosed by Krizhevsky *et al.* for image classification [19]. It’s first five layers are convolutional, with three max-pooling layers in between. The final three layers are fully connected layers. We reduced the number of filters and employ one fully connected layer less to make it suitable for 3D classification.

The **VGG-16** architecture was introduced by Simonyan *et al.* [20]. In contrast to AlexNet it uses multiple 3×3 convolutions replacing large kernel-sized filters. Our 3D implementation has a lower number of filters and layers. Additionally, the first convolutional layer uses a stride of two in every spatial direction.

He *et al.* developed a CNN with residual connections called **ResNet** [21]. Residual connections are formed by adding feature maps from the beginning of a block of convolutions to the final feature maps. Other than these connections the ResNet also uses average pooling instead of flattening the feature values before the first fully connected layer. The ResNet only uses max-pooling for downsizing in the first layer, subsequently downsizing is performed via strided convolutions. It also employs batch-normalization [36]. Similarly to the other networks, we adapted the number of channels and layers. We also increased the stride in the first layer to two.

The convergence of the networks is shown in Figure 11.2. 3D ResNet converged after the highest number of epochs. 3D AlexNet showed low generalization capability during the training. Training time for each network was 34 to 57 hours. Average prediction time was 3 seconds.

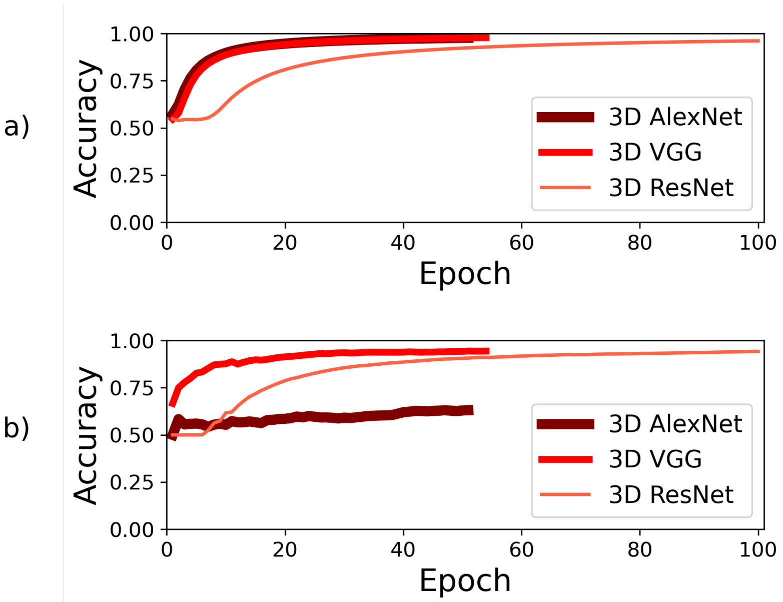


Figure 11.2: Average training a) and validation b) accuracy for all three networks across the training epochs. Each training is plotted up to the selected best epoch. The selected 3D ResNet trained for 100 epochs, while 3D VGG and 3D AlexNet trained 54 and 51 epochs, respectively. 3D VGG and 3D ResNet both reached a high validation accuracy at convergence.

Implementation

The described networks were implemented using TensorFlow 1.12 [37] and Python 3.6. For the evaluation and image processing we used SimpleITK 1.2.4, which provides a simplified interface to the Insight Toolkit (ITK) [38]. LRP was realized using the iNNvestigate neural networks! toolbox 1.0.9 [162]. Training and testing were performed on a Windows Server 2016 with an Intel Core i7-7700K CPU, 64GB RAM and a NVIDIA Quadro P5000 graphics card with 16 GB VRAM.

12. Publications

Journal Papers

- **A.-K. Golla**, C. Tönnes, T. Russ, D. F. Bauer, M. F. Frölich,S. J. Diehl, S. O. Schönberg, M. Keese, L. R. Schad, F. G. Zöllner and J. S. Rink. Automated Screening for Abdominal Aortic Aneurysm in CT scans under clinical conditions using Deep Learning. submitted to European Radiology, 23.03.2021.
- F. G. Zöllner, M. Kocinski, L. Hansen, **A.-K. Golla**, A. Serifovic-Trbalic, A. Lundervold, A. Materka and P. Rogelj. Kidney segmentation in renal magnetic resonance imaging - current status and prospects. submitted to IEEE Access, 09.03.2021.
- I. Hermann, **A.-K. Golla**, E. Martínez-Heras, B. Rieger, R. Schmidt, J.-S. Hong, W.-K. Lee, W. Yu-Te, E. Solana, S. Llufriu, A. Gass, L. R. Schad, S. Weingärtner and F. G. Zöllner. Lesion probability mapping in MS patients using a regression network on MR Fingerprinting. submitted to BMC Medical Imaging, 24.02.2021.
- D. F. Bauer, T. Russ, B. I. Waldkirch, W. P. Segars, L. R. Schad, F. G. Zöllner and **A.-K. Golla**. Generation of Multimodal Ground Truth Datasets for Abdominal Medical Image Registration using CycleGAN. in press at International Journal of Computer Assisted Radiology and Surgery, 2021.
- I. Hermann, E. Martínez-Heras, B. Rieger, R. Schmidt, **A.-K. Golla**, J.-S. Hong, W.-K. Lee, M. Nagetegaal, E. Solana, S. Llufriu, A. Gass, L. R. Schad, S. Weingärtner, F. G. Zöllner. Accelerated white matter lesion analysis based on simultaneous T_1 and T_2* quantification using Magnetic Resonance Fingerprinting and Deep Learning. Magnetic Resonance in Medicine, 86(1), pp.471-486, doi: 10.1002/mrm.28688, 2021.
- A. Adlung, N. K. Paschke, **A.-K. Golla**, D. Bauer, S. A. Mohamed, M. Samartzi, M. Fatar, E. Neumaier-Probst, F. G. Zöllner and L. R. Schad. ^{23}Na MRI in ischemic stroke: acquisition time reduction using CNN postprocessing. NMR in Biomedicine, 34 (4), p. e4474, doi: 10.1002/nbm.4474, 2021.
- **A.-K. Golla**, D. F. Bauer, R. Schmidt, T. Russ, D. Nörenberg, K. Chung, C. Tönnes, L. R. Schad, and F. G. Zöllner. Convolutional Neural Network Ensemble Segmentation with Ratio-based Sampling for the Arteries and Veins in Abdominal CT Scans. in press at IEEE Transactions on Biomedical Engineering, doi: 10.1109/TBME.2020.3042640, 2021.
- C. Tönnes, S. Janssen, **A.-K. Schnurr**, T. Uhrig, K. Chung, L. Schad and F. G. Zöllner. Deterministic Arterial Input Function selection in DCE-MRI for automation of quantitative perfusion calculation of colorectal cancer. Magnetic Resonance Imaging, 75, pp.116-123, doi: 10.1016/j.mri.2020.09.009, 2021.
- T. Russ, S. Goerttler, **A.-K. Schnurr**, D. Bauer, S. Hatamikia, L. R. Schad, F. G. Zöllner and K. Chung. Synthesis of CT images from digital body phantoms using CycleGAN. International Journal of Computer Assisted Radiology and Surgery, 14 (10), pp.1741-1750, doi: 10.1007/s11548-019-02042-9, 2019.
- **A.-K. Schnurr**, K. Chung, T. Russ, L. R. Schad and F. G. Zöllner. Simulation-Based Deep Artifact Correction with Convolutional Neural Networks for Lim-

ited Angle Artifacts. Zeitschrift für Medizinische Physik, 29 (2), pp.150-161, doi: 0.1016/j.zemedi.2019.01.002, 2019.

Peer-Reviewed Conference Proceedings

- **A.-K. Schnurr**, P. Eisele, C. Rossmanith, S. Hoffmann, J. Gregori, A. Dabringhaus, M. Kraemer, R. Kern, A. Gass and F. G. Zöllner. Deep Voxel-Guided Morphometry (VGM): Learning regional brain changes in serial MRI. Proc. International Workshop on Machine Learning in Clinical Neuroimaging, Held in Conjunction with MICCAI, Lima, Peru, LNCS 12449, pp.159-168, doi: 10.1007/978-3-030-66843-3_16, 2020.
- **A.-K. Schnurr**, L. R. Schad and F. G. Zöllner. Sparsely Connected Convolutional Layers in CNNs for Liver Segmentation in CT. Proc. Bildverarbeitung für die Medizin 2019, Lübeck, Germany, pp.80-85, doi: 10.1007/978-3-658-25326-4_20, 2019.

Conference Abstracts

- D. Bauer, C. Ulrich, **A.-K. Golla**, T. Russ, C. Tönnes, J. Leuschner, M. Schmidt, L. R. Schadand F. G. Zöllner. Image reconstruction using end-to-end deep learning for low-dose CT. Proc. Computer Assisted Radiology and Surgery, Munich, Germany, 2021.
- T. Russ, Tom, W. Wang, **A.-K. Golla**, D. F. Bauer, M. Tivnan, C. Tönnes, Y. W. Ma, T. Reynolds, S. Hatamikia, L. R. Schad, F. G. Zöllner, G. J. Gang, J. W. Stayman. Fast Reconstruction of non-circular CBCT orbits using CNNs. Proc. Fully3D Congress, Online, 2021.
- D. F. Bauer, E. Oelschlegel, **A.-K. Golla**, A. Adlung, T. Russ, I. Hermann, I. Brumer, J. Rosenkranz, F. Tollens, S. Clausen, P. Aumüller, L. R. Schad, D. Nörenberg and F. G. Zöllner. An Anthropomorphic Pelvis Phantom for Prostate Brachytherapy and Biopsy. Proc. ISMRM Congress, Online, 2021.
- I. Hermann, **A.-K. Golla**, Eloy Martinez-Heras, R. Schmidt, E. Solana, S. Llufriu, A. Gass, L. R. Schad, S. Weingärtner and F. G. Zöllner. Deep Learning Reconstruction of MR Fingerprinting for simultaneous T1, T2 mapping and generation of WM, GM and WM lesion probability maps. Proc. ISMRM Congress, Online, 2021.
- **A.-K. Schnurr**, M. Schöben, I. Hermann, R. Schmidt, G. Chlebus, L. R. Schad and F. G. Zöllner. Relevance Analysis of MRI Sequences for MS Lesion Detection. Proc. ESMRMB Congress, Online, 37, pp.77-78, 2020.
- D. F. Bauer, T. Russ, W. P. Segars, **A.-K. Schnurr**, L. R. Schad and F. G. Zöllner. Generation of Fully Annotated Abdominal T1-weighted MR Ground Truth Data for Image Segmentation and Registration using CycleGANs and the XCAT Phantom. Proc. ESMRMB Congress, Online, 37, pp.26-27, 2020.
- C. Tönnes, S. Janssen, **A.-K. Schnurr**, F. G. Zöllner. Exploration of deep learning approaches for the segmentation of colorectal cancer in DCE-MRI images. Proc. ESMRMB Congress, Online, 37, pp.173-174, 2020.
- C. Tönnes, S. Janssen, **A.-K. Schnurr**, T. Uhrig, K. Chung, L. R. Schad and F. G. Zöllner. Filter-pipeline based algorithm to find the AIF in DCE-MRI images for perfusion calculation of rectal cancer. Proc. ISMRM Congress, Paris, France, 28, p.3381, 2020.

- A. Adlung, N. Paschke, **A.-K. Schnurr**, S. Mohamed, V. Saase, M. Samartzi, M. Fatar, E. Neumaier-Probst and L. Schad. CNNs improve tissue sodium concentration accuracy in white and grey matter from stroke patients at 3T 23Na MRI. Proc. ISMRM Congress, Paris, France, 28, p.3809, 2020.
- **A.-K. Schnurr**, I. Hermann, R. Schmidt, A. Gass, F. G. Zöllner and L. R. Schad. Fully Convolutional Neural Network Segmentation of Multiple Sclerosis Lesions using T1 and T2* Maps. Proc. ESMRMB Congress, Rotterdam, Netherlands, 36, p.312, 2019.
- A. Adlung, N. Paschke, **A.-K. Schnurr**, E. Probst, S. Mohamed, M. Samartzi, M. Fatar and L. R. Schad. Can a Convolutional Neural Network reduce the Measurement Time for 23Na Quantification?. Proc. ESMRMB Congress, Rotterdam, Netherlands, 36, p.121, 2019.
- **A.-K. Schnurr**, C. Drees, L. R. Schad and F. G. Zöllner. Comparing sample mining schemes for CNN kidney segmentation in T1w MRI. 3rd International Conference on Functional Renal Imaging, Nottingham, United Kingdom, 2019.
- D. Bauer, **A.-K. Schnurr**, T. Russ, S. Goerttler, L. R. Schad, F. G. Zöllner and K. Chung. Synthesis of CT Images Using CycleGANs: Enhancement of Anatomical Accuracy. Proc. International Conference on Medical Imaging with Deep Learning, London, United Kingdom, 2019.
- **A.-K. Schnurr**, L. R. Schad and F. G. Zöllner. Kann ein festes Klassenverhältnis im Training die CNN-Lebersegmentierung im CT verbessern?. Proc. 4th Conference on Image-Guided Interventions, Mannheim, Germany, 2019.
- D. Bauer, **A.-K. Schnurr**, T. Russ, B. Waldkirch, L. R. Schad, F. G. Zöllner and K. Chung. Synthesis of CBCT Images from Digital Phantoms Using CycleGANs. Proc. 4th Conference on Image-Guided Interventions, Mannheim, Germany, 2019.
- C. Tönnes, S. Janssen, **A.-K. Schnurr**, T. Uhrig, K. Chung, L. R. Schad and F. G. Zöllner. The impact of variations in annotation on the AIF and perfusion parameter. Proc. 4th Conference on Image-Guided Interventions, Mannheim, Germany, 2019.
- B. Waldkirch, D. Bauer, **A.-K. Schnurr**, S. Engelhardt, F. G. Zöllner, L. R. Schad and I. Wolf. Point-based Evaluation of Multimodal Non-Rigid Image Registration of Synthetic Abdominal Data Generated with a Digital Phantom and a Cycle-Consistent Network. Proc. 4th Conference on Image-Guided Interventions, Mannheim, Germany, 2019.
- G. Kabelitz, D. Bauer, **A.-K. Schnurr**, T. Russ, I. Hermann, F. G. Zöllner, L. R. Schad and K. Chung. Evaluation phantom for multimodal imaging and image-guided needle interventions. Proc. 4th Conference on Image-Guided Interventions, Mannheim, Germany, 2019.
- W. Neumann, T. Uhrig, N. Paschke, M. Siegfarth, A. Rothfuss, G. Kabelitz, K. Chung, **A.-K. Schnurr**, L. R. Schad, J. Stallkamp and F. G. Zöllner. A multiparametric (1H, 23Na, diffusion, flow) anthropomorphic abdominal phantom for multimodal MR and CT imaging. Proc. Intl. Soc. Mag. Reson. Med., Montreal, Canada, 27, p.1121, 2019.
- **A.-K. Schnurr**, K. Chung, L. R. Schad and F. G. Zöllner. Deep Residual Learning for Limited Angle Artefact Correction. Proc. Bildverarbeitung für die Medizin 2018, Erlangen, Germany, p.280, 2018.

Supervised Theses

- **Master's Thesis** ”Region Selection for Arterial Input Function determination and segmentation of colorectal tumors in DCE-MRI images for perfusion calculation”, Christian Tönnes, 2019.
- **Bachelor's Thesis** ”Einfluss des Sample Mining Schemes auf die Nierensegmentierung in MR-Bildern mit Convolutional Neural Networks”, Christian Drees, 2019.
- **Bachelor's Thesis** ”Faltungsnetzwerke für die automatische Segmentierung von Multiple Sklerose Läsionen in quantitativen MRT-Sequenzen und Analyse der einzelnen Sequenzrelevanzen”, Marco Schöben, 2020.
- **Master's Thesis** ”Implementation of Deep learning method for dominant polycystic kidney disease segmentation and total kidney volume estimation in T2w Magnetic Resonance Imaging”, Sun I, 2020.
- **Bachelor's Thesis** ”Segmentierung von zerebralen Gefäßen in Computertomographie-Perfusionsdaten mit CNNs”, Lara-Jasmin Behrend, 2021.
- **Bachelor's Thesis** ”Einfluss der Vorverarbeitung auf Lebersegmentierung mit Convolutional Neural Networks (CNNs)”, Jennifer Stehr, 2021.

13. Curriculum Vitae

Personal Data

Name	**Alena-Kathrin Golla née Schnurr**.
Date of Birth	**05. November 1991**.
Place of Birth	**Hamburg**.
Nationality	**German**.

Education

10.2017 – today	**Doctoral Candidate: Dr. sc. hum.**, *Topic: Optimized Training Pipeline for Deep Learning Applications in Medical Image Processing*, Ruprecht Karl University of Heidelberg, Supervisor: Prof. Dr. Ing. Frank Gerrit Zöllner.
04.2015 – 06.2017	**Master of Science: Computational Visualistics**, *Focus: Image Processing and Computational Intelligence*, Otto von Guericke University Magdeburg, Grade: 1.3.
10.2011 – 05.2015	**Bachelor of Science: Computational Visualistics**, *Specialization: Medicine*, Otto von Guericke University Magdeburg, Grade: 1.8.
08. – 12.2013	**Exchange Semester**, *Computer Information Systems, Web & Digital Media Development*, University of Wisconsin – Stevens Point, USA.
2002–2010	**Abitur**, Dietrich-Bonhoeffer-Gymnasium Quickborn, Grade: 2.0.

Experience

06.2017 – today	**Research Assistant**, *Computer Assisted Clinical Medicine, Medical Faculty Mannheim, Ruprecht Karl University of Heidelberg*, since 09.2019: Project Leader „Image Guided Interventions“, since 10.2019: Lecturer Biomedical Engineering.
10.2018 – 01.2019, 10.2019 – 01.2020	**Lecturer**, *MTA School at Ludwigshafen Municipal Hospital*, Subject: Statistics for Radiology and Laboratory Medicine.

10.2016 – 05.2017 **Internship and Master’s Thesis**, *Philips Research Laboratories Hamburg*, Topic ”Pseudo-CT Generation using Dual-Compartment Gaussian Process Regression for MR-Only Radiotherapy Planning of Prostate Cancer”, Grade: 1.0.

06.2015 – 08.2016, 04.2014 – 09.2014 **Student Assistant**, *AG Computer Assisted Surgery, Faculty of Computer Science, Otto von Guericke University Magdeburg*, Development of medical software for segmentation and intervention planning using MeVisLab.

10.2015 – 02.2016, 10.2012 – 07.2013 **Tutor for Mexican and Colombian Students**, *International Office, Otto von Guericke University Magdeburg*, Assistance with administrative formalities and applications, excursion planning.

10.2014 – 02.2015 **Internship and Bachelor’s Thesis**, *Fraunhofer MEVIS, Bremen*, Topic ”Connectivity Segmentation of Vessel Structures”, Grade: 1.3.

Scholarships & Grants

2019 **Innovation competition „KI für KMU“**, *Development and integration of a new magnetic resonance analysis method for assessing disease activity in patients with multiple sclerosis*, Joint project of the Medical Faculty Mannheim with mediri GmbH, Heidelberg, and MedicalSyn GmbH, Stuttgart.

2019 **Erasmus+ Mobility Program**.

2018 **2nd Phase Research Campus M^{2}OLIE**, *Work package II.2 Cascading multi-structure segmentation.*

2017 **NVIDIA GPU Grant**, *Quadro P5000 graphics card.*

2013 **PROMOS Scholarship**.

14. Acknowledgements

First and foremost, I would like to thank Prof. Dr. Lothar Schad for the opportunity to start my PhD thesis in his research group.

I take this opportunity to express gratitude to Prof. Dr. Frank Zöllner for the supervision of this thesis and for his guidance through each stage of the process.

I would like to thank to the people at CKM. During my time there I enjoyed the opportunities to grow and learn.

Special thanks goes to Khanlian Chung, Barbara Waldkirch and Gordian Kabelitz for the warm welcome and to my colleagues Dominik Bauer, Tom Russ and Christian Tönnes. House 8 coffee breaks were always a comforting source of inspiration and community. I deeply enjoyed our collaborations as well as our discussions.

I would like to thank my friends for their constant support throughout my studies and personal life.

I am incredibly grateful for the opportunities and the encouragement my parents provided me with. It is due to them that I was able to pursue my career choice and write this thesis.

Last and most importantly, thank you to my husband Björn for being by my side through this adventurous time.