Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Natural Sciences
Department of Physics

Ingvild Askim Adde

Deep Learning For Automatic Segmentation Of Rectal Cancer On Magnetic Resonance Images From Two Independent Cohorts

Master's thesis in Applied Physics and Mathematics
Supervisor: Kathrine Røe Redalen
June 2021

NTNU
Norwegian University of
Science and Technology

Ingvild Askim Adde

Deep Learning For Automatic Segmentation Of Rectal Cancer On Magnetic Resonance Images From Two Independent Cohorts

Master's thesis in Applied Physics and Mathematics
Supervisor: Kathrine Røe Redalen
June 2021

Norwegian University of Science and Technology
Faculty of Natural Sciences
Department of Physics

NTNU
Norwegian University of
Science and Technology

ABSTRACT

PURPOSE

Manual tumor delineation is required for several purposes, such as calculation of quantitative image biomarkers and target delineation in radiotherapy. However, the delineation process is a time-consuming task that is subject to intra- and interobserver variations. It would therefore be beneficial to develop a method that automatically segments the tumor and reduces intra- and interobserver variations. In addition, the automatic segmentation would save time for the radiologists and oncologists.

The aim of this thesis was to explore a Deep Learning (DL) approach with Convolutional Neural Networks (CNNs) for automatic segmentation of rectal cancer, based on Magnetic Resonance (MR) images from two independent patient cohorts.

MATERIALS AND METHODS

Two datasets with MR images of rectal cancer were used for training and testing of the DL models. The first dataset consisted of 89 patients from the Locally Advanced Rectal Cancer - Radiation Response Prediction (LARC-RRP) study, and the second dataset of 110 patients from the Functional MRI of Hypoxia-mediated Rectal Cancer Aggressiveness (OxyTarget) study. Manual delineations of the tumor volumes were made by experienced radiologists and used as ground truth.

Several DL models with a U-Net architecture were developed and varied in terms of image input, standardization method, loss function, learning rate, and data augmentation. The LARC-RRP dataset, the OxyTarget dataset, and a combination of both datasets were used as input for the models. Each dataset was split into a training set, validation set and test set. The model performances were evaluated based on the mean Dice Similarity Coefficient per patient (DSC_P) of the validation set. The best DL model for each dataset were then compared to the results from a Shallow Machine Learning (SML) approach where classification was carried out based on voxel intensities. Finally, the best DL models were tested on new unseen data by using the hold-out test sets as input.

RESULTS

The best model performance was achieved with the OxyTarget dataset when solely using T2 Weighted (T2w) MR images which contained tumor as input. The model used a learning rate of $1e-04$, data augmentation, the z-score normalization combined with the matching of histograms (MH + Z-Score) as standardization method, and the Modified Dice as loss function. The model achieved a DSC_P of 0.691 on the test set and outperformed the SML approach. The DSC_P between two radiologists, which delineated 76 of the patients in the OxyTarget dataset, was equal to 0.805. Thus, the model performed inferior to the interobserver variation.

CONCLUSION

The thesis explored whether DL models with a U-Net architecture can be used to automatically segment rectal cancer based on MR images from two independent patient cohorts. The final model had a DSC_P below the interobserver DSC_P. Thus, the results indicate that the DL model needs further improvement before it can be fully implemented in a clinical setting. However, the model could be carefully implemented with a satisfying threshold value on a per image slice basis. This would still increase the efficiency in the tumor delineation process. To improve the model performance the effect of including multiple MR sequences, as well as the use of transfer learning between different cohorts, various standardization methods, data augmentation methods and model architectures should be further investigated.

SAMMENDRAG

HENSIKT

Inntegning av kreftsvulstvolumet er en viktig del av både kvantitativ bildeanalyse og strålebehandling. Dette er en tidkrevende oppgave som er forbundet med usikkerhet grunnet interobservatørvariabilitet. Det ville derfor vært fordelaktig å utvikle en metode som segmenterer kreftsvulsten automatisk. En slik metode kan potensielt spare tid for radiologene/onkologene, og bidra til en mer konsekvent inntegning.

Hensikten med denne masteroppgaven var derfor å undersøke om dyp læring (DL) med konvolusjonelle nevrale nettverk kan benyttes for automatisk segmentering av kreftsvulster, basert på Magnetisk Resonans (MR) bilder fra to forskjellige pasientkohorter med endetarmskreft.

MATERIALER OG METODER

To datasett med Magnetresonanstomografi (MRI) av endetarmskreft ble benyttet for trening og testing av ulike DL-modeller. Det første datasettet bestod av 89 pasienter fra Locally Advanced Rectal Cancer - Radiation Response Prediction (LARC-RRP)-studien. Det andre datasettet bestod av 110 pasienter fra Hypoxia-mediated Rectal Cancer Aggressiveness (OxyTarget)-studien. Inntegningen av kreftsvulstene på bildene ble utført av erfarne radiologer, og ble benyttet som fasit for modellene.

Flere ulike DL-modeller med U-Net arkitektur ble utviklet. Modellene ble kombinert med forskjellige bildetyper, tapsfunksjoner, læringsrater, standardiseringsmetoder og dataøkningsmetoder. LARC-RRP-datasettet, OxyTarget-datasettet, og en kombinasjon av de to datasettene, ble brukt som data for modellene. Hvert datasett ble splittet opp i et treningssett, valideringssett og testsett. Modellene ble evaluert basert på gjennomsnittlig Dice likhetskoeffisient per pasient (DSC_P) på valideringssettet. Den beste DL-modellen for hvert datasett ble deretter sammenlignet med resultatene fra en maskinlæringsmodell hvor klassifiseringen ble gjennomført basert på voxelintensitetene. Til slutt ble de beste DL-modellene testet på usett data ved å benytte testsettene.

RESULTATER

Den beste modellen benyttet T2-vektede bilder som inneholdt kreftsvulst fra OxyTarget-datasettet. I tillegg anvendte modellen en læringsrate på $1e-04$,

dataøkning, en kombinasjon av z-verdi normalisering og tilpasning av piksel histogram som standardiseringsmetode, og den Modifiserte Dice tapet som tapsfunksjonen. Modellen oppnådde en DSC_P lik 0.691 på testsettet, og utkonkurrerte dermed maskinlæringsmetoden. Ved sammenlikning av inntegningen av kreftsvulster fra to ulike radiologer, som segmenterte 76 av pasientene i OxyTarget-datasettet, ble DSC_P beregnet til å være lik 0.805. Dermed hadde den beste DL-modellen en lavere prestasjon sammenlignet med variasjonen mellom de to manuelle inntegningene.

KONKLUSJON

Denne masteroppgaven har utforsket om DL med en U-Net arkitektur kan benyttes for automatisk segmentering av endetarmskreft, basert på MR-bilder fra to forskjellige pasientkohorter. Den endelige modellen hadde en lavere DSC_P sammenlignet med variasjonen mellom to radiologer (DSC_P). Dette indikerer at modellen må forbedres før den kan anvendes klinisk på egenhånd. Man kan allikevel benytte modellen på hvert enkelt bildesnitt, kombinert med en egnet grenseverdi som avgjør om den automatiske segmenteringen burde godkjennes eller ikke. En slik implementering av modellen kan fremdeles spare tid og øke effektiviteten i inntegningsprosessen. Videre utvikling av modellen bør utforske effekten av ulike MR-sekvenser. I tillegg bør bruken av ulike dataøkningsmetoder, standardiseringsmetoder og muligheten for overføring av kunnskap mellom kohorter utforskes videre.

PREFACE

This master's thesis was submitted as the end result of the 10 semester integrated master program in Applied Physics and Mathematics at the Norwegian University of Science and Technology (NTNU). The work started in January 2021 and was completed in June 2021. The thesis was a continuation of the project thesis conducted during the fall semester 2020.

The results of my research were presented at the *Nordic Association of Clinical Physics (NACP) 2021 Symposium* as a poster presentation. In addition, an abstract for the *Biology-Guided Adaptive Radiotherapy (BiGART) Symposium 2021* was accepted for oral presentation and a manuscript for fast-track peer-review publication in *Acta Oncologica* will be submitted based on the work presented in this thesis. The abstracts can be found in Appendix D and E, respectively.

I would like to thank my supervisor, Professor Kathrine Røe Redalen, who has provided excellent guidance, insight and feedback throughout the entire process. I am truly grateful for the valuable follow-up she has given me, and for making the project possible.

Furthermore, I wish to thank PhD Candidate Franziska Knuth for all of the helpful discussions. She has kindly answered all of my questions, and has given me irreplaceable comments on my work.

I would also like to thank Professor Cecilia Marie Futsæther for making the collaboration with the Norwegian University of Life Sciences (NMBU) possible, and for including me in meetings with her research group. The meetings have offered valuable tips and input from experienced scientists. I would further like to express my gratitude to PhD Candidate Ngoc Huynh Bao at NMBU for answering my countless questions regarding the models. The project would have been difficult to finish without her knowledge and insight.

Finally, I would like to thank my friends and family for always supporting and believing in me. The thesis would have been hard to finish without your constant encouragement and love.

Ingvild Askim Adde
Trondheim, June 21, 2021

The design of this thesis is based on the typographical look-and-feel `classicthesis` template in LaTeX originally developed by André Miede.

CONTENTS

1 INTRODUCTION 1
1.1 Tumor Delineation . . . 1
1.2 Deep Learning . . . 2
1.3 Automatic Tumor Segmentation . . . 2
1.4 Related Work . . . 3
1.5 Aim . . . 3
1.6 Declaration . . . 4
2 THEORY 5
2.1 Magnetic Resonance Imaging . . . 5
2.1.1 T_2 Weighted Images . . . 8
2.1.2 Diffusion Weighted Images . . . 9
2.1.3 Artifacts . . . 10
2.1.4 Windowing . . . 11
2.2 Machine Learning . . . 14
2.2.1 Linear Discriminant Analysis . . . 14
2.2.2 Quadratic Discriminant Analysis . . . 15
2.2.3 Support Vector Machine . . . 16
2.3 Deep Learning . . . 18
2.3.1 Neural Networks . . . 18
2.3.2 Loss Functions . . . 20
2.3.3 Gradient Based Optimization . . . 21
2.3.4 Overfitting . . . 23
2.3.5 Standardization of Input Data . . . 25
2.3.6 Training, Validating and Testing . . . 26
2.3.7 Performance Metrics . . . 28
2.4 Deep Learning for Image Segmentation . . . 31
2.4.1 Convolutional Neural Networks . . . 31
2.4.2 The U-Net Architecture . . . 34
3 MATERIALS AND METHODS 37
3.1 The LARC-RRP Study . . . 37
3.2 The OxyTarget Study . . . 37
3.3 Datasets . . . 38
3.4 Pre-Processing . . . 40
3.4.1 Cropping of Images . . . 40
3.4.2 Splitting into Training, Validation and Test Sets . . . 42
3.4.3 Conversion to the Hierarchical Data Format Version 5 File Format . . . 44
3.4.4 Standardization of Input Data . . . 46
3.5 Deep Learning Model . . . 49
3.5.1 Hyperparameters . . . 50
3.5.2 Data Augmentation . . . 52

3.6 Code and Software . . . 55
3.7 Analysis of the Model . . . 55
3.7.1 Box Plots . . . 55
3.7.2 Violin Plots . . . 56
3.8 Shallow Machine Learning Model . . . 56
3.8.1 Post-Processing . . . 58
3.9 Experimental Setup . . . 59
4 RESULTS 61
4.1 5-Fold Cross Validation . . . 61
4.2 Model Tuning . . . 64
4.2.1 Learning Rates and Loss Functions . . . 64
4.2.2 Standardization of Input Images . . . 69
4.2.3 Data Augmentation . . . 71
4.2.4 Summary of Model Tuning . . . 73
4.3 Comparison of OxyTarget Model and Radiologist$^2_0$ . . . 74
4.4 Shallow Machine Learning vs. Deep Learning . . . 76
4.5 Model Performance When Only Using Tumor Slices . . . 79
4.6 Model Performance on Test Sets . . . 82
4.7 Including Diffusion Weighted Images . . . 87
5 DISCUSSION 89
5.1 Splitting of Datasets . . . 89
5.2 Finding the Optimal Model Configuration . . . 90
5.2.1 Standardization of Input Data . . . 92
5.2.2 Data Augmentation . . . 94
5.2.3 A Complex Task . . . 95
5.3 Model Performance . . . 96
5.3.1 Comparison With Shallow Machine Learning Models . . . 98
5.3.2 Impact of Tumor Slices . . . 100
5.3.3 Generalization Ability . . . 101
5.3.4 The Importance of Performance Metrics . . . 102
5.4 Different Magnetic Resonance Sequences . . . 103
5.5 The Datasets . . . 104
5.6 Clinical Impact . . . 106
5.7 Further Work . . . 107
5.7.1 Model Configuration . . . 107
5.7.2 Transfer Learning . . . 109
5.7.3 The Input Images . . . 110
5.7.4 Additional Performance Metrics . . . 110
5.7.5 The Black Box Phenomena . . . 110
6 CONCLUSION 113

BIBLIOGRAPHY 115

APPENDIX 125
A SPLITTING OF DATASETS 125
A.1 Traditional Split of OxyTarget Data . . . 125

A.2 Traditional Split of LARC-RRP Data 125
A.3 5-Fold Cross Validation Split of OxyTarget Data 125
A.4 5-fold Cross Validation Split of LARC-RRP Data 126
B HDF5 FILES 127
B.1 HDF5 Files 127
C CODE 129
C.1 Default Augmentation Configuration 129
C.2 Best Combination Augmentation Configuration 130
D NACP 2021 SYMPOSIUM 131
D.1 Abstract 131
E BIGART 2021 133
E.1 Abstract 133

LIST OF FIGURES

Figure 2.1 Flipping of spins . . . 5
Figure 2.2 Dephasing and rephasing of spins . . . 7
Figure 2.3 Spin Echo sequence . . . 7
Figure 2.4 Stejskal-Tanner sequence . . . 9
Figure 2.5 Diffusion weighted image example . . . 11
Figure 2.6 Zipper artifact . . . 12
Figure 2.7 Windowing . . . 13
Figure 2.8 Linear discriminant analysis . . . 15
Figure 2.9 Hard and soft support vector machine . . . 17
Figure 2.10 Neural network . . . 18
Figure 2.11 Activation of a neuron . . . 19
Figure 2.12 Gradient descent . . . 22
Figure 2.13 Traditional split . . . 26
Figure 2.14 5-fold cross validation . . . 27
Figure 2.15 Confusion matrix . . . 28
Figure 2.16 Convolutional layer . . . 33
Figure 2.17 Padding . . . 34
Figure 2.18 Max pooling . . . 35
Figure 2.19 U-Net architecture . . . 36
Figure 3.1 Example of a T2 weighted image . . . 39
Figure 3.2 Image dimensions . . . 41
Figure 3.3 Example of a T2 weighted image before and after cropping . . . 41
Figure 3.4 Stratification of the Combined dataset . . . 43
Figure 3.5 Hierarchical data format version 5 file structure . . . 46
Figure 3.6 Pixel distributions after standardization . . . 48
Figure 3.7 Matching of pixel histograms image example . . . 49
Figure 3.7 Image examples of data augmentation . . . 54
Figure 3.8 Box plot structure . . . 56
Figure 3.9 Violin plot structure . . . 57
Figure 3.10 Unfolding methods . . . 58
Figure 3.11 Experimental plan . . . 60
Figure 4.1 Violin plots of mean DSC_S for 5-fold cross validation . . . 63
Figure 4.2 Violin plots of DSC_P for 5-fold cross validation . . . 63
Figure 4.3 Violin plots of DSC_P for different learning rates and loss functions . . . 66
Figure 4.4 Median DSC_P for different learning rates . . . 66
Figure 4.5 Median DSC_P for different loss functions . . . 67
Figure 4.6 Image examples when changing the learning rates and loss functions . . . 68

Figure 4.7 Violin plots of DSC_P for different standardization methods . . . 70
Figure 4.8 Violin plots of DSC_P for different data augmentation methods . . . 72
Figure 4.9 Maximum DSC_S image slices . . . 73
Figure 4.10 DSC_P for two different radiologists . . . 75
Figure 4.11 Scatter plot of DSC_P for each patient . . . 75
Figure 4.12 Violin plots of DSC_P for shallow machine learning models vs. deep learning models . . . 77
Figure 4.13 Example of training performance for LARC-RRP data . 79
Figure 4.14 Violin plots of DSC_P when only using tumor slices . . . 80
Figure 4.15 Violin plots of DSC_P on the test sets . . . 82
Figure 4.16 Image examples from the test patient with highest DSC_P 86
Figure 4.17 Violin plots of DSC_P when including diffusion weighted images . . . 87
Figure 4.18 Image examples when including diffusion weighted images . . . 88
Figure 5.1 Example of different image content . . . 92
Figure 5.2 Examples of dissimilar manual delineations . . . 97

LIST OF TABLES

Table 3.1 Overview of datasets 39
Table 3.2 Overview of image slices 42
Table 3.3 Traditional split of Combined data 44
Table 3.4 5-fold cross validation split of Combined data 44
Table 3.5 Overview of hierarchical data format version 5 files . . 45
Table 3.6 Structure of hierarchical data format version 5 datasets 45
Table 3.7 Overview of the U-Net architecture 50
Table 3.8 Fixed hyperparameters 51
Table 3.9 Tunable hyperparameters 52
Table 4.1 5-fold cross validation on the OxyTarget dataset 62
Table 4.2 5-fold cross validation on the LARC-RRP dataset 62
Table 4.3 Median DSC_P for different learning rates and loss functions 65
Table 4.4 Median DSC_P for different standardization methods . . 69
Table 4.5 Median DSC_P for different data augmentation methods 71
Table 4.6 Model parameters for the final deep learning models . 73
Table 4.7 Median DSC_P for two different radiologists 74
Table 4.8 Median DSC_P for shallow machine learning models . . 76
Table 4.9 Median DSC_P for shallow machine learning models vs. deep learning models 78
Table 4.10 Median DSC_P when only using tumor slices 81
Table 4.11 Median DSC_P on test sets 82
Table 4.12 DSC_P for OxyTarget test set patients 83
Table 4.13 DSC_P for LARC-RRP test set patients 84
Table 4.14 DSC_P for Combined test set patients 85
Table 4.15 Median DSC_P when including diffusion weighted images 87

ACRONYMS

ACC Accuracy

AI Artificial Intelligence

ART Adaptive Radiation Therapy

CNN Convolutional Neural Network

CT Computed Tomography

DICOM Digital Imaging and Communications in Medicine

DL Deep Learning

DSC Dice Similarity Coefficient

DSC_P mean Dice Similarity Coefficient per patient

DSC_S Dice Similarity Coefficient per image slice

DWI Diffusion Weighted Image

ERR Error

FN False Negative

FOV Field Of View

FP False Positive

FPR False Positive Rate

GAN Generative Adversarial Network

HD Hausdorff Distance

HDF5 Hierarchical Data Format version 5

ILSVRC ImageNet Large-Scale Visual Recognition Challenge

LDA Linear Discriminant Analysis

LOOCV Leave One Out Cross Validation

MSD Mean Surface Distance

MR Magnetic Resonance

MRI Magnetic Resonance Imaging

NIfTI Neuroimaging Informatics Technology Initiative

NMBU Norwegian University of Life Sciences

NTNU Norwegian University of Science and Technology

N3 Nonparametric Nonuniform Intensity Normalization

PRE Precision

QDA Quadratic Discriminant Analysis

ReLU Rectified Linear Unit

RF Radio Frequency

ROI Region Of Interest

SML Shallow Machine Learning

SVM Support Vector Machine

TE Echo Time

TN True Negative

TP True Positive

TPR True Positive Rate

TR Repetition Time

T2w T2 Weighted

1 INTRODUCTION

According to the *World Health Organization*, cancer is the second leading cause of death globally [1]. In 2018 there were 17 million new cases of cancer worldwide, while 9.6 million of these cases resulted in deaths [2]. In Norway, a total of 34 979 new cancer cases were reported in 2019 [3]. Out of these incidents, rectum and rectosigmoid cancer were the seventh and eighth most frequent in men and women, respectively.

Before treatment of cancer, image diagnostics of the tumor is an essential step to decide the treatment. Computed Tomography (CT) and Magnetic Resonance Imaging (MRI) are commonly used to capture images within the body. According to the Norwegian guidelines, patients diagnosed with rectal cancer should undergo a diagnostic MRI examination of the pelvis [4]. In addition, CT of the abdomen and thorax should be carried out to detect the possible metastatic spread of the disease. The goal of the Magnetic Resonance (MR) examination is to determine the stage of the disease such that optimal treatment can be decided [4]. The cancer is determined as locally advanced for patients where the cancer has spread to nearby organs and/or grown into the bowel wall. In this case, the patient should undergo preoperative radiation treatment combined with chemotherapy, where the goal is to reduce the size and stage of the tumor. Consequently, the probability of a successful outcome after surgery is increased. In addition, the preoperative radiation treatment and chemotherapy are given in order to reduce the risk of local relapse [4].

1.1 TUMOR DELINEATION

The goal of radiation treatment is to kill as many cancer cells as possible by using ionizing radiation. At the same time, radiation of healthy tissue and critical organs should be avoided. It is therefore essential to know where the cancer cells are located. Delineation of the tumor is the process where the tumor is marked in images. It is a crucial step to calculate radiation dose and create an optimal radiation treatment plan. The delineations of tumor volumes are also needed in order to calculate imaging biomarkers. A biomarker is a measurable indicator that says something about the biological state [5]. Tumor biomarkers can be measured in images and provide objective information about the tumor biology, the tumor environment, and changes in response to an intervention [5]. Hence, these biomarkers can give more information about the tumor aggressiveness, treatment response, and probability of survival. Radiomics is a method in medicine that aims to identify biomarkers from a large number of imaging features. Therefore, it is often beneficial with a standardized delineation method when performing radiomics [5].

Today, delineations of the tumors are done manually by radiologists or oncologists. Hence, the manual delineations are exposed to intra- and interobserver variations. Another aspect to consider in the delineation process is the varying image quality. In radiotherapy, delineations are most commonly carried out in CT images since these images provide essential information about the electron density in various tissues that are needed for dose calculation. However, MR images provide better soft-tissue contrast. Consequently, it is easier to achieve a more accurate delineation when using MR images. Furthermore, the delineation process is considered as one of the weakest links in terms of accuracy during radiotherapy [6]. As the delineation of the tumor occurs in one of the first steps of the radiation treatment plan, it greatly impacts the treatment quality. If a delineation is inaccurate, the proceeding error will propagate throughout the treatment chain. Accordingly, a non-optimal treatment is given to the patient. Another drawback with manual tumor delineation is that it requires a lot of time.

1.2 DEEP LEARNING

Over the last few years, the use of Artificial Intelligence (AI) and Deep Learning (DL) have increased rapidly. Even though the fundamental concepts of DL were already well understood in 1989, it was the advances in hardware and datasets that truly accelerated the progression within the field [7]. The development of high-performance graphic chips, combined with the fact that the internet took off and data was shared across the world, resulted in a wide range of possibilities for AI. Hence, DL emerged in the computer vision field, and in 2012 a DL approach based on a Convolutional Neural Network (CNN) outclassed the competing participants in the ImageNet Classification computer vision competition [8]. Since the breakthrough in 2012, the interest in CNNs has increased significantly and is considered the standard network structure for a wide variety of computer vision tasks.

1.3 AUTOMATIC TUMOR SEGMENTATION

As introduced in Section 1.1, manual tumor delineation is subject to intra- and interobserver variations and is a very time-consuming task. A possible solution to these problems is to create a DL model that automatically segments the tumor in the images. The model could be trained on a set of images and consequently delineate the tumor automatically in new unseen images. In this way, the intra- and interobserver variations would be removed, and a more standardized method would be developed.

The model would also save valuable time for the radiologists and oncologists and increase the efficiency. This would be especially useful in Adaptive Radiation Therapy (ART) which is a radiation process where the treatment plan can be modified using systematic feedback of measurements [9]. The goal of the method is to consider changes in the tumor volume that occur during

treatment. In this way, ART further increases the optimization of the radiation treatment. However, the method requires several sequential CT, or MR, scans with corresponding delineations to estimate the variations of the target volume. Automatic tumor segmentation would therefore be beneficial in order to speed up the process while maintaining delineation accuracy.

1.4 RELATED WORK

Today, DL is applied to several medical imaging problems such as brain segmentation [10], breast cancer segmentation [11, 12] and radiomics [13, 14]. Furthermore, various DL approaches have been applied to perform automatic tumor segmentation in patients with rectal cancer [15–19]. Trebeschi et al. [19] demonstrated that deep learning can perform accurate localization and segmentation of rectal cancer in MR imaging in the majority of the patients. Accordingly, the study concluded that deep learning technologies have the potential to improve the speed and accuracy of MRI-based rectum segmentations. Recently, Xia et al. [16] developed a deep learning-based automatic solution for rectal cancer treatment that showed promising results for improving the efficiency of treatment planning. In 2016, Gambacorta et al. [15] validated an autocontouring software in a clinical practice. According to the study, autosegmentation systems of CT scans from 44 patients with rectal cancer only partially met the acceptability criteria. Hence, the need for further improvement was confirmed.

The limited amount of available data remains a major challenge for medical images [17, 20, 21]. Therefore, it would be advantageous if one could combine data from different cohorts to increase the data size. However, a thorough search of relevant literature yielded that few investigations have been conducted on automatic segmentation of rectal cancer by using two independent patient cohorts.

1.5 AIM

The aim of the thesis was to explore a DL approach with CNNs for automatic segmentation of tumors, based on MR images from two independent cohorts. First, the thesis sought to investigate how different parameters influenced the model performance. Second, the thesis looked into how the model performance was affected when Diffusion Weighted Images (DWIs) were included as an additional input to the T2w images, compared to solely using T2w images. The model performance was evaluated and compared with results obtained from a Shallow Machine Learning (SML) approach where classification was done based on voxel intensities. The final goal was to examine whether or not the DL model was good enough to be implemented in a clinical setting.

1.6 DECLARATION

The thesis is based on the authors project thesis, written during the fall semester in 2020. Hence, the introduction in Section 2.1, Subsection 2.1.1, Section 2.2 and Subsection 2.3.1 in Chapter 2 are adapted from the authors project thesis with minor adjustments. Furthermore, Section 3.1, 3.2, 3.8 in Chapter 3, and Subsection 5.7.2 are taken and adjusted from the authors project thesis.

2 THEORY

2.1 MAGNETIC RESONANCE IMAGING

Magnetic Resonance Imaging (MRI) is a highly sensitive method for imaging the anatomy and functions in the human body [22, 23]. The imaging technique is based on observations of nuclear spins, which is an intrinsic property of the nucleus. A nucleus is said to be Magnetic Resonance (MR) active if it has an odd mass number. This is due to the fact that with an odd mass number there is either a proton or neutron which is not paired up, hence giving the nucleus a net spin [24]. For human applications the most frequently used nuclear spins are hydrogen (^{1}H). The main reason for using hydrogen is because a large amount of the human body consist of water, which means that the body has a large amount of hydrogen available. In addition, hydrogen has a relatively high magnetic moment which contributes to a stronger MR signal. Hydrogen has a spin value of $\frac{1}{2}$, and in the case of an externally applied magnetic field B_0, the spins tend to align parallel or anti-parallel to the magnetic field. The parallel spins will be slightly favored, since they have a lower energy state compared to the anti-parallel spin state [24]. Consequently, a net magnetization vector appears in the same direction as B_0 as illustrated in Figure 2.1a.

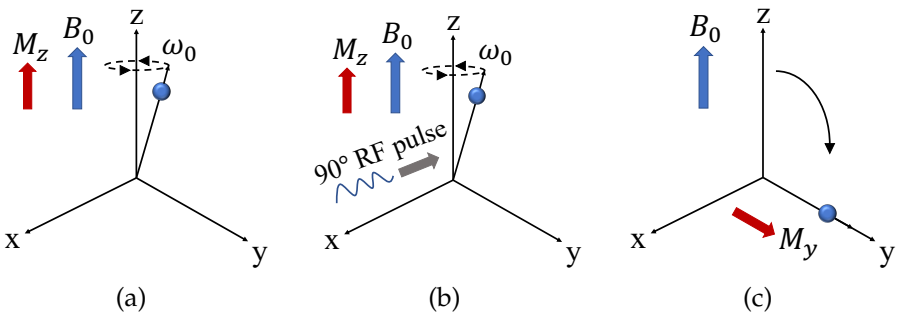


Figure 2.1: Illustration of how an MR signal is created. First, an external magnetic field (B_0) is applied such that the spins align to B_0 and a net magnetization vector M_z is created (a). Second, a Radio Frequency (RF) pulse at the Larmor frequency (w_0) enters the system (b) and disturbs the equilibrium in (a). The RF pulse flips the spins out of equilibrium and a transverse magnetization component M_y is created (c). The distortion of the magnetic field induces a current which creates an MR signal.

The spins precess around B_0 with a given frequency. This frequency is called the *Larmor frequency*, and is defined as

$$\omega_0 = -\gamma B_0 \tag{1}$$

where γ is the gyromagnetic ratio, which describes the relationship between the magnetic momentum and the angular momentum. It is a specific property of the nucleus, and the sign in front of γ specifies the direction of the precession [25].

The net magnetization vector is said to be in equilibrium when it is aligned with B_0. When creating an MR signal the magnetization vector is disturbed by applying RF pulses at the Larmor frequency, as illustrated in Figure 2.1b. The angle at which the net magnetization is moved out of equilibrium when applying RF pulses is referred to as the *flip angle* [22, 23]. Figure 2.1c shows how the deviation from equilibrium result in a transverse component, and hence a change in the magnetic field. Consequently, a current is induced in the MR coils and a signal can be measured. A flip angle of 90° is most commonly used when disturbing the magnetic field. This is due to the fact that with a flip angle of 90° all of the spins are moved into the transverse plane, and one obtains the strongest possible MR signal.

After disturbing the equilibrium position, the net magnetization vector will try to realign itself with B_0. During this process energy is transferred to the surroundings through molecular motion. There are two main relaxation mechanisms that brings the net magnetization vector back to equilibrium, which are called *longitudinal* and *transverse relaxation* [24]. In the case of longitudinal relaxation there is a decrease of the magnetization in the transverse plane and a restoration of the magnetization in the longitudinal plane. It is an exponential process, and it is also referred to as T_1 *relaxation*. During transverse relaxation there is a destruction of the transverse component due to spin-spin interactions and field inhomogenities from the machine. These mechanisms causes a total dephasing (T_2^*) of the spins which is given as

$$\frac{1}{T_2^*} = \frac{1}{T_2} + \frac{1}{T_2'} \tag{2}$$

where T_2 describes the dephasing due to random spin-spin interactions, while T_2' describes the dephasing due to systematic field inhomogenities from the machine [22, 23]. The T_2' dephasing can be refocused by applying a 180° RF pulse such that the total dephasing is only due to spin-spin interactions. This is illustrated in Figure 2.2. The transverse relaxation is commonly referred to as T_2 *relaxation*.

When creating MR signals a Spin Echo sequence is a common sequence to use [23]. An illustration of the sequence is presented in Figure 2.3. The sequence starts out by applying a 90° RF pulse in order to excite the spins into the transverse plane. Next, a 180° pulse is applied to refocus the spins. After a given amount of time another excitation pulse is applied to the system. The time between two successive excitation pulses is called the Repetition Time (TR) which determines the amount of T_1 relaxation allowed before the next excitation occurs [23]. The Echo Time (TE) describes the time between a

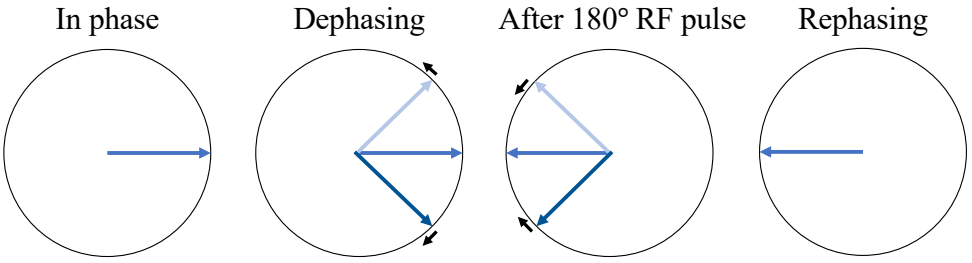


Figure 2.2: Illustration of how the spins begin to dephase due to random spin-spin interactions and systematic field inhomogenities from the machine. The spins are rephased by applying a 180° RF pulse.

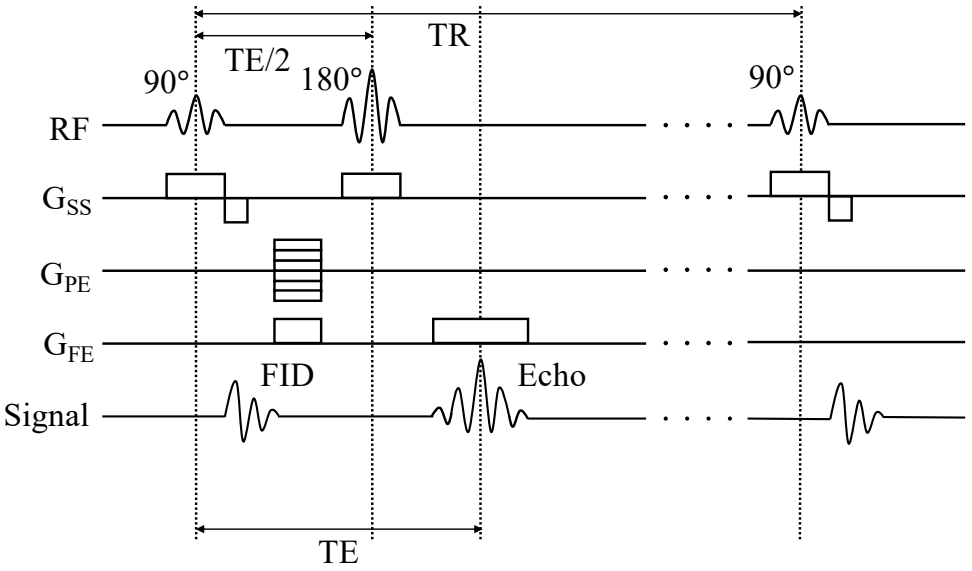


Figure 2.3: Illustraion of the Spin Echo sequence with spatial encoding gradients. RF is the radio frequency pulses, G_{SS} is the slice selecting gradient, G_{PE} is the phase encoding gradient and G_{FE} is the frequency encoding gradient.

given excitation pulse and the actual readout signal induced in the coils. The length of TE determines the amount of T_2 relaxation in the system. In a Spin Echo sequence the 180° pulse is applied after a time TE/2 [23].

A key element while creating MR images is spatial encoding [24]. Spatial encoding is performed by applying magnetization gradients from different directions. The z-direction is usually defined as the axis going through the feet of the patient and through the head. B_0 is most commonly applied along the z-direction [22]. The different magnetization gradients are created by running currents through specialized coils in the MR system, and usually linear magnetic fields are created. The coils are normally oriented along the x-, y- and z-directions of the system, hence a gradient can be created in all of these directions. The magnetic field created by the gradients will modify the externally applied field, and in this way the magnetic field will vary along the position of the spins.

When a gradient is turned on the Larmor frequency at position i can be written as

$$\omega_i = -\gamma(B_0 + \delta_i) \tag{3}$$

where δ_i is the magnetic contribution from the gradient at position i. Equation (3) shows that by applying a gradient in a given direction the spins will have different Larmor frequencies depending on their position [23, 24]. When performing spatial encoding the first step is to apply a gradient in the same direction as B_0. In this way spins at various z-coordinates will have different Larmor frequencies. Slice selection can then be performed by applying a RF pulse which contains a bandwidth of specific Larmor frequencies. Only the spins with the same frequencies as the RF pulse will be excited, and thus only these specific spins will contribute to the MR signal [22, 23]. After selecting a slice in the z-direction the spatial position within the slice needs to be encoded. This can be done by applying a gradient to the frequency and phase direction, which normally corresponds to the x- and y-direction of the slice. During signal readout a gradient is applied in the frequency direction. In this way spins in the given direction will have different frequencies depending on their position, and therefore the various spins will give different signal frequencies. A gradient is also applied to the phase direction [22–24]. This gradient is applied after the excitation pulse, and causes an incremental change in the phases of the spins.

The signal recorded during the sequence is mapped to the frequency domain, which is also referred to as k-space. In k-space the frequencies along the horizontal lines correspond to the frequency direction, while those along the vertical lines correspond to the phase direction. A two dimensional Fourier Transform is then used in order to reconstruct the image from k-space [25].

2.1.1 T_2 *Weighted Images*

Different tissues in the body have different relaxation times due to varying biological properties. The relaxation times depend on the molecular motion, which can be quantified by the correlation time τ_c [22]. Large molecules, such as fat, have slow movement and therefore a long τ_c. The long τ_c of fats result in efficient longitudinal and transverse relaxation, which means they have a short T_1 and T_2 relaxation time. Water molecules on the other hand are much smaller, and hence they also have a shorter τ_c. This gives a more inefficient longitudinal and transverse relaxation, and thus a long T_1 and T_2 relaxation time.

The difference in relaxation times can be utilized to create contrast in the MR image. In a T2 Weighted (T2w) image the goal is to enhance the difference between T_2 values of the tissues [23]. One should therefore try to minimize the difference in T_1 times. This can be done by using a long TR, such that the

longitudinal magnetization is able to fully recover for all tissues. In addition, the difference in T_2 times should be maximized. One should therefore use a TE which is long enough for the signal in various tissues to start decaying due to dephasing [23]. In the case of fat and water, fat will decay much faster than water. Hence the signal should be collected at the TE which gives the maximum difference between the signal decay in fat and water. For a T2w image one should therefore use a long TR and a long TE.

2.1.2 *Diffusion Weighted Images*

Diffusion is the process where molecules undergo a constant random thermal motion. The process is also referred to as random Brownian motion, and occurs for all molecules in a fluid or gas at temperatures above zero kelvin [22, 26]. Hence, all molecules in the body undergo diffusion which leads to a movement of the spins.

In a perfectly homogenous medium the probability of diffusion is equal in all directions. However, the human body is more complex consisting of biological barriers such as cellular membranes, extracellular compartments and intracellular compartments [22]. Water molecules in the extracellular compartments have shown to have relatively free diffusion while intracellular molecules have shown relatively restricted diffusion, *i.e.* the probability of diffusion is not equal in all directions [27]. The tissues in the human body have a characteristic cellular architecture with different proportions of intra- and extracellular compartments. Consequently, different tissues in the body will have different diffusion properties. The characteristic diffusion properties of the tissues can be utilized in order to create contrast in the MR images. Thus, a Diffusion Weighted Image (DWI) is an image weighted such that the movement of spins create the contrast. In the following, we will look more into detail of how a DWI is created.

As presented in Section 2.1 a Spin Echo sequence is commonly used when creating MR signals. However, in the case of a DWI a pair of diffusion sensitive gradients are added to the Spin Echo sequence. The most commonly used DWI sequence is called the Stejskal-Tanner sequence [22, 28] and is illustrated in Figure 2.4.

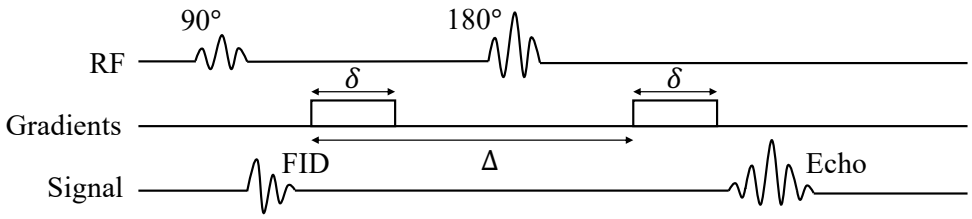


Figure 2.4: The Stejskal-Tanner sequence which consist of a Spin Echo sequence and diffusion gradients. The sequence is commonly used for DWI.

First of all a 90° RF pulse flips the net magnetization into the transverse plane. Then, the first diffusion gradient induces a net phase change, where the amount of phase change depends on the position of the spins. Next, a 180° RF pulse refocuses the spins. Finally, a second diffusion gradient is applied which induces a negative phase shift. The second diffusion gradient reverses the phase change that occurred in the first diffusion gradient, and further refocuses the spins leading to an echo [22, 28]. Consequently, if the spins are not moving the refocusing of the signal will be perfect. However, if there is diffusion the spins will have changed position during the sequence. Hence, a different phase shift will be induced during the second diffusion gradient to the spins that have moved during the sequence. The refocusing of the signal will therefore not be perfect when there is diffusion in the same direction as the gradient is applied [22]. Accordingly, there will be low MR signals in areas where there are high diffusion, while in areas with low diffusion there will be a high MR signal. The signal loss caused by diffusion can be expressed as

$$S(b) = S(0)e^{-\gamma^2 G^2 \delta^2 \Delta D} = S(0)e^{-bD} \tag{4}$$

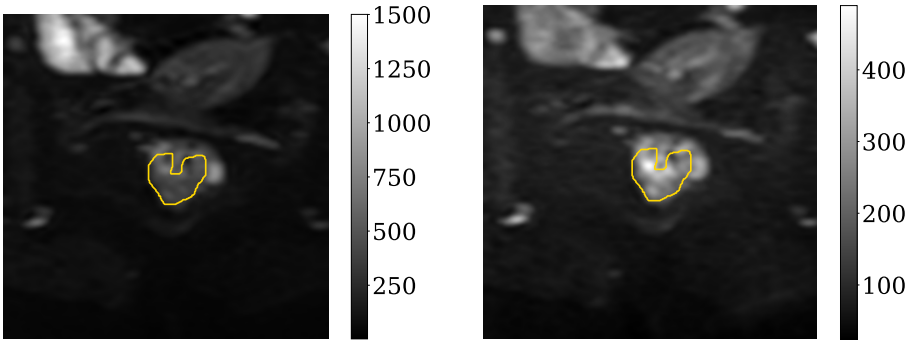
where γ is the gyromagnetic ratio, G is the strength of the diffusion gradient, δ is the amount of time the diffusion gradient is turned on, Δ is the time between the diffusion gradients, while D is the diffusion coefficient which gives the diffusion rate of a molecule [29]. The diffusion weighting factor, also known as the b-*value*, is then defined as

$$b = -\gamma^2 G^2 \delta^2 (\Delta - \delta/3) \tag{5}$$

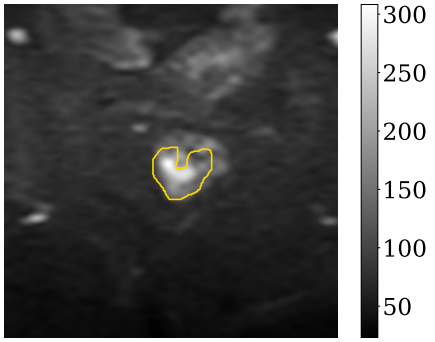
Equation (4) shows that if $b = 0$ then $S(b) = S(0)$, which corresponds to a T2w image with no diffusion weighting. Hence, the b-value determines the amount of diffusion weighting in the image [22]. Tumors have shown to have restricted diffusion, and will therefore appear bright in a DWI [27]. Figure 2.5 shows an example of a DWI with increasing b-value for a patient with rectal cancer. The tumor is marked by the yellow contour, and one can notice how the tumor turns brighter for increasing b-values.

2.1.3 *Artifacts*

In some cases the MR images might have lower quality due to undesired alternation in the data. The undesired alternation in data is also known as an *artifact*, and can be caused by the hardware, the software, the digital processing or by environmental influences [22, 23]. There is a wide range of different MR artifacts, however they are usually classified according to what is causing them: physiological, inherent physics or hardware. Physiological artifacts usually occurs due to patient motion during the MR scan, or due to flow of



(a) $b = 100 \frac{s}{mm^2}$ (b) $b = 500 \frac{s}{mm^2}$



(c) $b = 1000 \frac{s}{mm^2}$

Figure 2.5: Example of a DWI with different b-values for a patient with rectal cancer. The bright areas show regions where the diffusion is low. The tumor delineation made by the radiologist on the corresponding T2W image is marked in yellow. The color bar indicates the image intensities.

molecular spins. Artifacts due to the inherent physics can be caused by chemical shifts due to the different chemical environments of fat and water. Magnetic susceptibility artifact is another type of artifact classified as inherent physics. This type of artifact occurs because different tissues magnetize differently [22]. Finally, the last kind of artifacts can be caused by the hardware. One example of a hardware artifact is the *Zipper* artifact, which is caused by external RF signals entering the room due to a leak in the RF shielding [23]. The artifact appears as a dense line across the image at one or several specific points. Figure 2.6 shows an example of a possible Zipper artifact for a patient with rectal cancer.

2.1.4 *Windowing*

Windowing describes the process where the image gray scale can be adjusted. Thus, the windowing influences the perceived image contrast and image brightness. This is done by attributing certain levels on the gray scale to certain signal intensities, as illustrated in Figure 2.7. The windowing is completely independent of the MR image acquisition and processing [30]. Consequently, one of the major challenges with MRI techniques is that the intensities

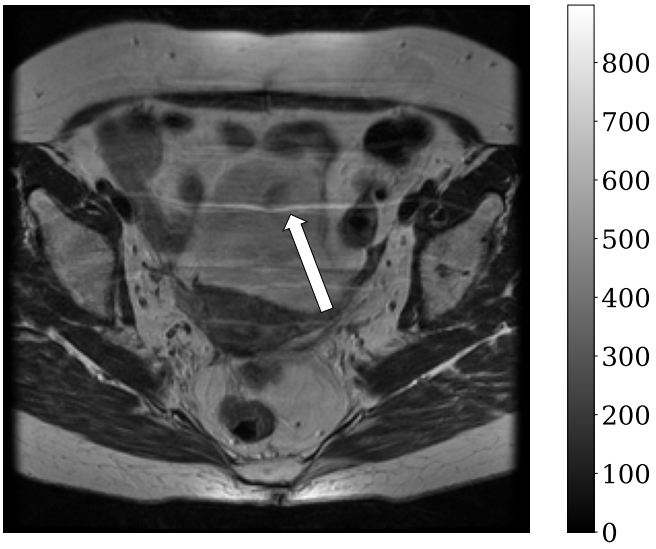


Figure 2.6: Example of a possible Zipper artifact in an image slice of a patient with rectal cancer. The Zipper artifact is pointed out by the white arrow, and appears as a dense line across the image. The color bar indicates the image intensities.

in the images do not have a fixed meaning [31]. The intensities in the images will not be identical even though one uses the same protocol, the same body region, the same scanner, and the same patient each time. Therefore, MR images can not be displayed at preset windows and in most cases the window settings need to be adjusted per patient case. However, when comparing images with each other one should always have the same window level and center. When comparing images with different image settings one is essentially comparing structures with different signal intensities. Consequently, the result might be misleading when comparing images with different window settings [30].

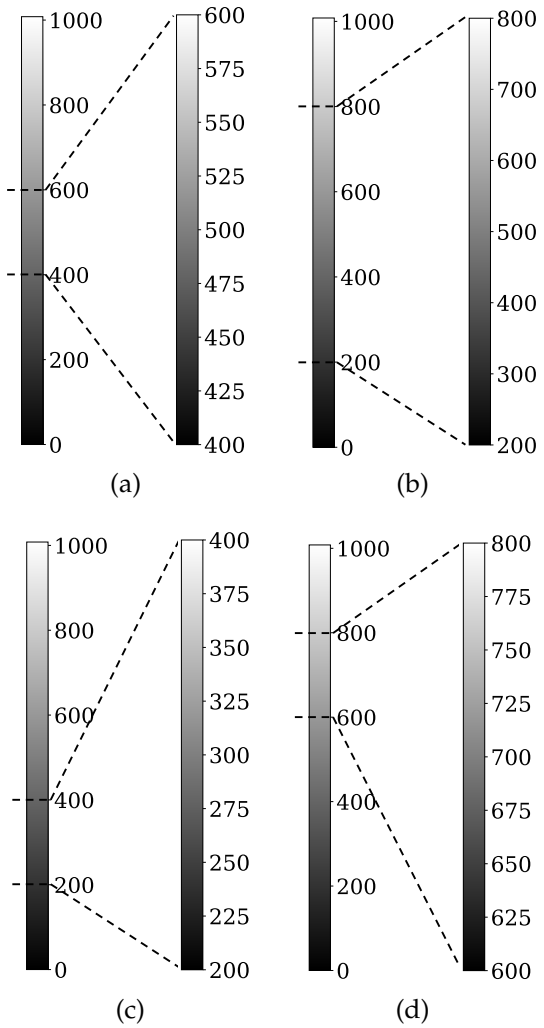


Figure 2.7: Illustration of how windowing adjusts the gray scale such that white corresponds to the highest image intensity and black corresponds to the lowest image intensity. The original image intensity scale is shown to the left and the adjusted image intensity gray scale is shown to the right, in all of the images. In (a) and (b) the windowing level is narrowed and widened, while in (c) and (d) the window center is moved down and up.

2.2 MACHINE LEARNING

Machine learning is a type of Artificial Intelligence (AI) which is frequently used in computer science and computer technology. The goal is to create a computer program that automatically learn from experience, without being explicitly programmed [32]. In this way several tasks can be solved automatically by machines, which have had a huge impact on the world as we know it today. In order for computer programs to learn automatically, a well-defined learning problem is necessary [32].

The machine learning approaches are often divided into supervised and unsupervised learning [32, 33]. In the case of supervised learning the entire training dataset used to gain experience is labeled. The labels in the training data can be described as a teacher, which is providing extra information to the model telling it how it should process the data. The model can later use the gained experience to predict labels on new unseen data. Unsupervised learning on the other hand, do not have labeled training data. In this case the goal is to give a summary or compressed version of the data [32]. Clustering is an example of an unsupervised learning process where the aim is to divide the data into subsets of similar objects [33, 34]. In the next sections an explanation of the theory behind different supervised classification methods is given.

2.2.1 *Linear Discriminant Analysis*

Linear Discriminant Analysis (LDA) is included in the family of linear models for classification and regression [35]. In this family of models it is assumed to be a linear relationship between the input and output of the model. Suppose we have input vectors $x_1, ..., x_n$, each assigned to one of two classes (y_1, y_2). The linear models usually define the decision function as

$$f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + w_0 \tag{6}$$

where $\boldsymbol{w}$ is the weight vector, $\boldsymbol{x}$ is the input vector and w_0 is a constant referred to as the threshold. The classification of input data given with the decision function in (6) can be described as

$$\boldsymbol{x} \in \begin{cases} y_1 & \text{if } f(\boldsymbol{x}) < k \\ y_2 & \text{if } f(\boldsymbol{x}) > k \end{cases} \tag{7}$$

where k is a constant representing the decision boundary. When using LDA one assumes that the two classes have Gaussian distributions and equal covariance matrices [36] as in equation (8).

$$\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma} \tag{8}$$

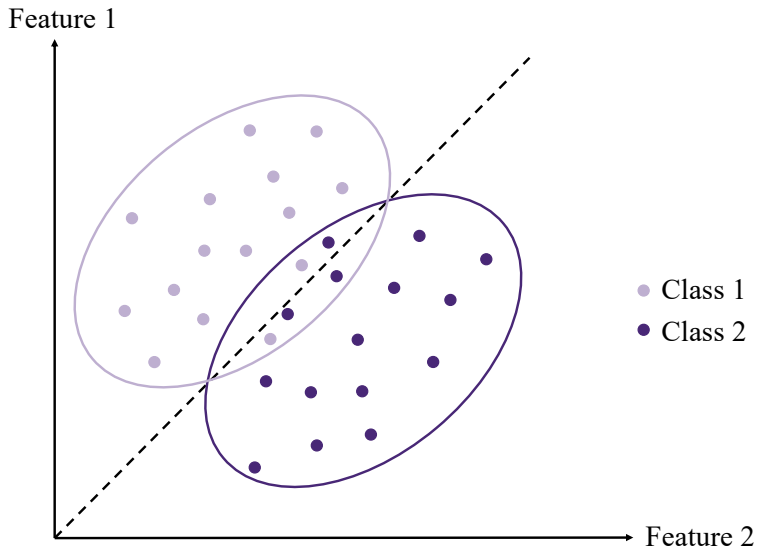


Figure 2.8: Input data, consisting of two features, have been classified into two different classes by utilizing Linear Discriminant Analysis (LDA). The dashed line illustrates the best projection direction found with LDA.

These assumptions result in a linear decision boundary, and equation (6) can be rewritten as

$$w^T x + w_0 = 0 \tag{9}$$

In LDA the values of w is optimized such that the distance between samples from different classes is maximized, while the distance between samples in the same class is minimized. For a problem consisting of two classes this can be done by using the criterion given in equation (10), provided by Fisher

$$J(w) = \frac{w^T S_B w}{w^T S_W w} \tag{10}$$

Here S_B is called the "between" scatter matrix and is defined as $S_B = (m_1 - m_2)(m_1 + m_2)^T$, where m_i is the mean of samples from class i. S_W is called the "within" scatter matrix and is defined as $S_W = S_1 + S_2$. In this case S_i is given as $S_i = \sum_{x \in D_i} (x - m_i)(x - m_i)^T$, with D_i as the collection of samples from class i. The goal is to find the values of w which maximizes the ratio between S_B and S_W [35]. Figure 2.8 gives an illustration of LDA when the input data consists of two different features.

2.2.2 *Quadratic Discriminant Analysis*

Quadratic Discriminant Analysis (QDA) is closely related to LDA [36]. However, when using QDA one do not assume that the covariance matrices are equal

$$\Sigma_1 \neq \Sigma_2 \tag{11}$$

Since the covariance matrices are not equal to each other, the quadratic term cannot be thrown away. This result in a quadratic decision boundary and can be expressed in the following form:

$$x^T W x + w^T x + w_0 = 0 \tag{12}$$

In a similar manner as for LDA the goal of QDA is to maximize the distance between samples from differend classes, while minimizing the distance between samples in the same class [36]. This is again done by optimizing the weights in equation (12).

2.2.3 *Support Vector Machine*

Support Vector Machine (SVM) is another supervised learning method used for classification and regression [34]. However, this method is mainly used in high dimensional feature spaces. Suppose that we have a set of training examples given as $S = (x_1, y_1), ..., (x_m, y_m)$, where each $x_i \in \mathcal{R}^d$ and $y_i \in \{+1, -1\}$. The goal of the SVM is to find a hyperplane in the d-dimensional feature space which divides the space into two halves, and distinctly classifies the data points. The dataset is defined as linearly separable if

$$y_i(\langle w, x_i \rangle + b) > 0, \quad \forall i \in [m] \tag{13}$$

where $\langle \cdot \rangle$ is the inner product and b is the bias term. For any separable dataset there exists several hyperplanes which successfully classifies the data points [34]. The SVMs therefore introduce the concept of *margin* to find the best hyperplane. The margin of a hyperplane with respect to a training set is defined as the minimal distance between a point x in the training set and the hyperplane defined by (w, b). The distance is given in equation (14)

$$|\langle w, x \rangle + b| \tag{14}$$

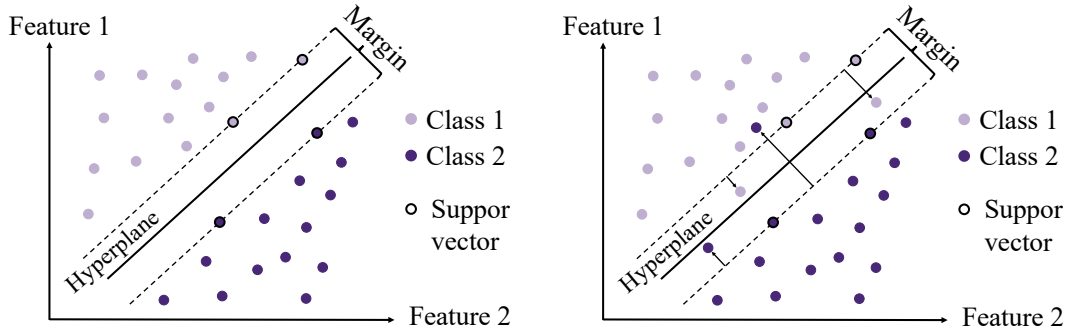
with $||w|| = 1$. A SVM can be divided into *Hard-SVM* or *Soft-SVM*, depending on the learning rule used to choose the optimal hyperplane [34, 37, 38]. An illustration of the two SVM-methods is given in Figure 2.9. For Hard-SVM the aim is to separate the training set with the largest possible margin. Hence, the dataset needs to be linearly separable in order to use Hard-SVM, and the learning rule is defined as

$$(w_0, b_0) = \underset{(w,b)}{\arg\min} ||w||^2 \quad \text{s.t.} \quad y_i(\langle w, x_i \rangle + b) \geqslant 1, \quad \forall i \in [m] \tag{15}$$

where the optimal parameters are given as $\hat{w} = \frac{w_0}{\|w_0\|}$ and $\hat{b} = \frac{b_0}{\|w_0\|}$ [34]. Soft-SVM on the other hand can be applied even if the training set is not linearly separable. In this case the constraint in (15) is allowed to be violated for some examples in the training set. The Soft-SVM learning rule is given as

$$(\hat{w}, \hat{b}) = \min_{w,b,\xi} \left(\lambda \|w\|^2 + \frac{1}{m} \sum_{i=1}^{m} \xi_i \right) \quad \text{s.t.} \quad y_i(\langle w, x_i \rangle + b) \geqslant 1 - \xi_i \quad \text{and } \xi_i \geqslant 0, \ \forall i \in [m] \tag{16}$$

where ξ_i is the slack variable for data point i, and λ is a tradeoff parameter. The slack variable estimates how much the constraint in (15) is being violated, while the tradeoff parameter controls the importance of $\|w\|^2$ [34].



(a) Hard-SVM. (b) Soft-SVM.

Figure 2.9: An illustration of Hard-SVM and Soft-SVM. With Hard-SVM the data is separated with the largest possible margin. With Soft-SVM some of the data points are violating the restriction given in equation (15). The arrows illustrate how much the restriction is violated.

2.3 DEEP LEARNING

Several learning tasks in the real world are highly complex. This makes it very difficult to predict the correct output. A machine learning model needs to be retrained through human intervention if the output is not correct. Deep learning on the other hand, is a subfield of machine learning which is designed to learn through their own errors. In this way human intervention is not needed in order to correct the wrong output [39].

2.3.1 *Neural Networks*

The deep learning models are inspired by the understanding of human brains, and are learned through *neural networks* [39–42]. These neural networks consist of several *neurons* divided into different *layers*. Figure 2.10 illustrates how the neural networks consist of an input layer, several hidden layers and an output layer. Deeper models contains more hidden layers, while shallow models only have one or two hidden layers. Each layer in the network provides a different interpretation to the data it is given. Hence a deep learning model is a multistage way to learn data representations [7].

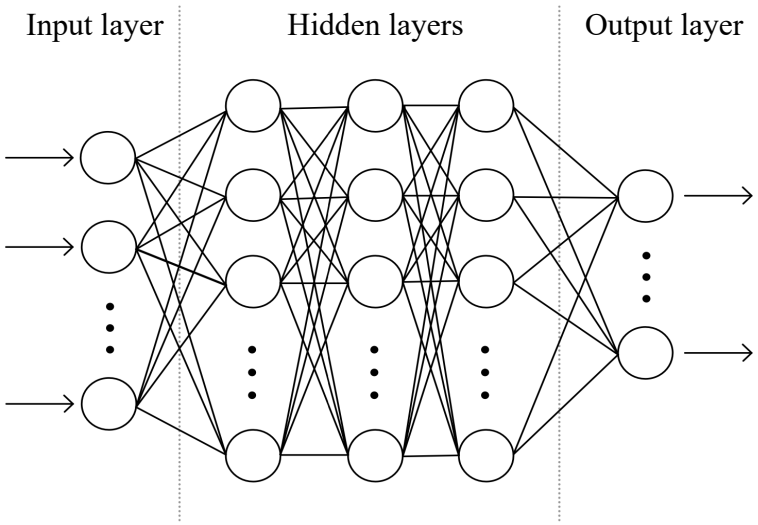


Figure 2.10: Neural networks are made up of several layers, where each layer consist of neurons. The neural networks include an input layer, hidden layers and an output layer. Neurons in different layers are connected through learnable parameters called weights.

Figure 2.10 also indicates how a neuron in one layer is used as input for the neurons in the next layer. The strength of a connection between two neurons in different layers is given by learnable parameters called *weights*. The value of a given neuron can therefore be expressed as

$$a(x) = \sum_i w_i x_i - b \tag{17}$$

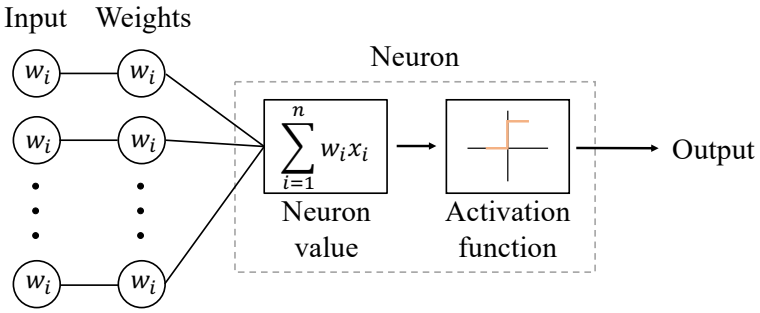


Figure 2.11: An illustration of how the output from a neuron is calculated. The inputs and corresponding weights are summed and sent into an activation function. The activation function decides what the output from the neuron should be.

where x_i is the input from neuron i in the previous layer, w_i is the weight connecting neuron i to the given neuron, and b is the bias term [39]. The output from a neuron is determined by an activation function, as illustrated in Figure 2.11. There are several different activation functions used in deep learning models. The simplest is the linear identity function, $f(a) = a$, which simply outputs the neurons value [7]. However, in most cases non-linear activation functions are used. An example is the binary step function which is defined as

$$f(a) = \begin{cases} 1 & \text{if } a \geqslant 0 \\ 0 & \text{if } a < 0 \end{cases} \tag{18}$$

Another well-known activation function is the Rectified Linear Unit (ReLU) function [42], which is given in equation (19). This activation function is often used as default when implementing neural networks.

$$f(a) = \begin{cases} a & \text{if } a \geqslant 0 \\ 0 & \text{if } a < 0 \end{cases} \tag{19}$$

The logistic sigmoid function is another commonly used activation function [42]. This activation function outputs values in the range between 0 and 1. The function is defined as

$$f(a) = \frac{1}{1 + e^{-a}} \tag{20}$$

The input values are propagated through the neural network by calculating the value of each neuron, as given in equation (17), and activating them with an appropriate activation function. Finally, the outputs are returned as illustrated in Figure 2.11.

2.3.2 *Loss Functions*

A *loss function* is used to measure the error between the predictions of the network and the true target value [7, 34]. For linear regression a common loss function is the squared error, which is given as

$$J(\boldsymbol{w}) = \sum_{i=1}^{N} (y_i - t_i)^2 \tag{21}$$

Here $\boldsymbol{w}$ are the weights of the network, while y_i and t_i are the predicted output and target output of sample i respectively.

For classification models the cross entropy is a frequently used loss function [43], which is defined as

$$J(\boldsymbol{w}) = -\sum_{i=1}^{N}\sum_{k=1}^{K} t_{i,k} \log(f(x_i)) \tag{22}$$

In this case $t_{i,k}$ signify whether or not the target output of sample i belongs to class k. If a sample i of the target output is of class k then $t_{i,k} = 1$, otherwise $t_{i,k}$ is equal to zero. The value of $f(x_i)$ gives the predicted probability that input sample i belongs to class k, with the given weights $\boldsymbol{w}$ [43]. Since $f(x_i)$ represents a probability it is important to use an activation function which has an output between 0 and 1 in the last layer of the neural network. An example of such an activation function is the logistic sigmoid function given in equation (20). For a *binary classification* problem there are only two available classes. Thus, the cross entropy can be written as

$$J(\boldsymbol{w}) = -\sum_{i=1}^{N} t_i \log(f(x_i)) + (1 - t_i)\log(1 - f(x_i)) \tag{23}$$

One problem with the cross entropy is that it is easily affected by imbalance in the dataset, which is often the case for segmentation tasks in medical images [44]. Another loss function was therefore defined by Milletari et al. [44] based on the Dice coefficient given in equation (34). In this way one could directly optimize the objective overlap between two regions. The loss function was called the Dice loss, and for two binary volumes it is defined as

$$D(\boldsymbol{w}) = 1 - \frac{2\sum_i y_i t_i}{\sum_i y_i^2 + \sum_i t_i^2} \tag{24}$$

where y_i is the i-th voxel of the predicted volume, and t_i is the i-th voxel of the target volume. If sample i belongs to the positive class then $t_i = 1$. In the opposite case, $t_i = 0$ if sample i belongs to the negative class. The value of y_i gives the predicted probability that sample i belongs to the positive class.

2.3.3 *Gradient Based Optimization*

The goal of the deep learning model is to minimize the error measured by the loss function. This means that the model needs to find the parameters where the derivative of the loss function goes to zero [39]. The method used to achieve this goal is called *backpropagation*. Backpropagation starts with the final loss value and propagates backwards from the output layer to the input layer [39]. During backpropagation the chain rule is used to compute the derivative of the loss function with respect to the parameters, and in this way the model finds the contribution that each parameter had in the loss value. Each of the parameters is then updated iteratively, in the opposite direction of the gradient, such that the loss will move towards a minimum [39]. This method is called *gradient descent*, and can be expressed in the following way

$$\boldsymbol{w}^{(i+1)} = \boldsymbol{w}^{(i)} - \lambda \nabla J(\boldsymbol{w}^{(i)}) \tag{25}$$

where $\nabla J(\boldsymbol{w}^{(i)})$ is the gradient of the loss function at iteration i, and λ is the learning rate which indicates the update magnitude.

Figure 2.12 illustrates the concept of gradient descent in a one dimensional parameter space, with one available training sample. However, in real neural networks there could be up to several millions of parameters which needs to be updated. In addition, there should also be a lot of training data available in order to tune the parameters. Thus, running gradient descent optimization on all of the training data at once, while updating all of the parameters, would be extremely time consuming and computational expensive. A solution to this problem is to rather run the network on a *batch* of training samples [7]. Hence, the network parameters would be updated based on the performance on the samples in the given batch. This approach is known as *mini-batch stochastic gradient decent*, where the term *stochastic* refers to the fact that each batch of data is drawn at random from all of the training samples [7, 39]. The method will result in less accurate updates of the parameters since the loss calculated from a given batch might not coincide with how it would be if the loss was calculated based on all of the training samples. However, the method saves a lot of time and computational power, and is therefore often used as optimization method in deep learning [7].

2.3.3.1 *Learning Rate and Momentum*

When using the gradient based optimization method it is important to choose a reasonable value of the learning rate, in order to find the model with the lowest possible error. The learning rate is considered as one of the most difficult hyperparameters to set, because it significantly affects the model performance [39]. If the learning rate is too large, the descent may never converge towards a minimum and the iterations might end up in completely random locations of the loss function. In the opposite case, where the learning rate is

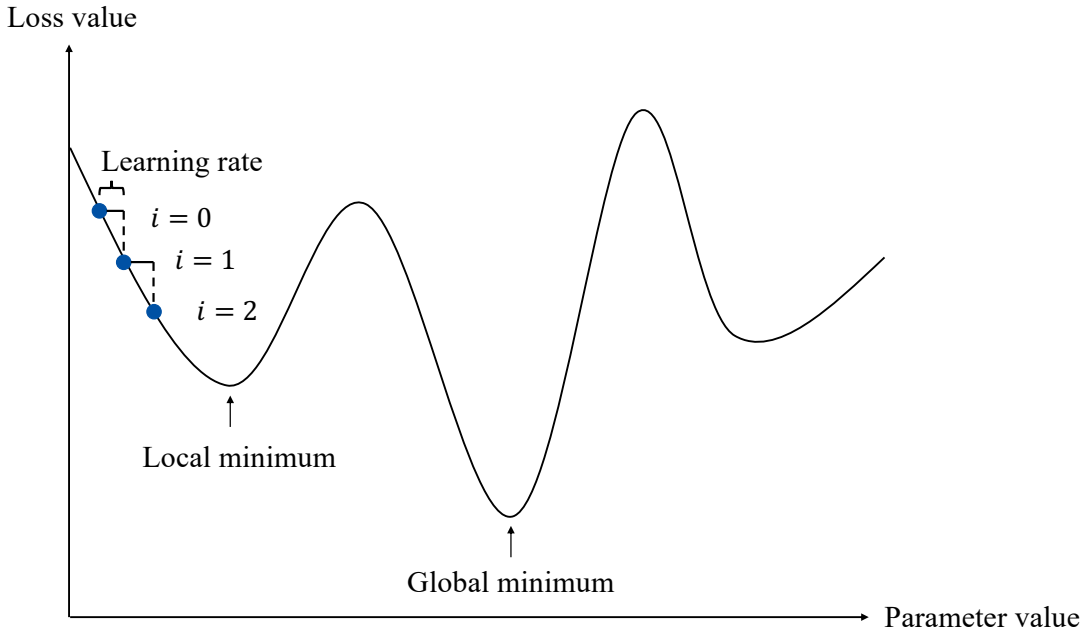


Figure 2.12: An illustration of gradient descent in a one dimensional parameter space, *i.e.* one learnable parameter. The loss function might consist of both local and global minima, and the goal is to reach the global minimum. The parameter value is updated at each iteration i, and is updated such that the loss will move towards a minimum. The size of the step in the opposite direction of the gradient is also known as the learning rate.

too small, the descent towards a minimum will take several iterations, and it might get stuck in a local minimum. However, one could use the concept of *momentum* in order to avoid local minima and reduce the convergence speed [7]. When introducing momentum the parameter values are not updated based solely on the current gradient value, but also based on previous parameter updates. This is done by establishing a variable $\boldsymbol{v}$ which plays the role of velocity. Hence, the velocity gives the direction and speed at which the parameters move through parameter space [39]. The update rule for the parameters, when introducing momentum, can be given as

$$\boldsymbol{v}^{(i)} = \alpha \boldsymbol{v}^{(i-1)} - \lambda \nabla J(\boldsymbol{w}^{(i)}) \tag{26}$$

$$\boldsymbol{w}^{(i+1)} = \boldsymbol{w}^{(i)} + \boldsymbol{v}^{(i)} \tag{27}$$

where $\alpha \in [0, 1)$ is a hyperparameter which determines the contributions of previous gradients. The larger α is to λ, the more previous gradients affect the current direction [39].

2.3.3.2 *Optimization Algorithms*

There are several variants of the gradient descent method which uses momentum and adaptive learning rates in order to optimize the model parameters.

Adaptive learning rates is a technique where a separate learning rate is introduced for each model parameter, and these learning rates are adapted during training [39]. *AdaGrad*, *RMSProp* and *Adam* are some examples of optimization algorithms which uses momentum and adaptive learning rates, combined with stochastic gradient descent, as optimization method. Next, we will have a closer look at the Adam algorithm.

The Adam algorithm was proposed by Diederik P. Kingma and Jimmy Lei Ba in order to deal with the problem of very noisy and/or sparse gradients [45]. The name "Adam" comes from the phrase "adaptive learning", and the algorithm is generally regarded as robust to the choice of hyperparameters [39]. Pseudo-code of the optimizer is given in Algorithm 1. The method uses estimates of first and second moments of the gradient to calculate the individual adaptive learning rates for the different parameters. The first moment is the mean of the gradient, while the second moment is the uncentered variance of the gradient. Algorithm 1 shows how the exponential moving averages of the gradient ($\boldsymbol{w}^{(i)}$), and the squared gradient ($\boldsymbol{v}^{(i)}$), is controlled by the hyperparameters $\beta_1, \beta_2 \in [0, 1)$.

Algorithm 1 The Adam optimization algorithm. All operations on vectors are element-wise. Suggested default settings are: $\lambda = 0.001, \beta_1 = 0.9, \beta_2 = 0.999$ and $\epsilon = 10^{-8}$ [39, 45]

Require: Learning rate λ
Require: Small constant used for numerical stabilization ϵ
Require: Exponential decay rates for the moment estimates $\beta_1, \beta_2 \in [0, 1)$
Require: Loss function $J(\boldsymbol{w})$
Require: Initial parameters $\boldsymbol{w}^{(0)}$
  Initialize 1^{st} and 2^{nd} moment vectors $\boldsymbol{m}^{(0)} = 0$, $\boldsymbol{v}^{(0)} = 0$
  Initialize iteration step $i = 0$
  while $\boldsymbol{w}^{(i)}$ not converged **do**
    Sample a mini-batch of m examples from the training set $\{\boldsymbol{x}^{(1)}, ..., \boldsymbol{x}^{(m)}\}$ with corresponding targets $\boldsymbol{y}^{(i)}$
    $i \leftarrow i + 1$
    Compute gradient: $\boldsymbol{g}^{(i)} \leftarrow \nabla J(\boldsymbol{w}^{(i-1)})$
    Update biased first moment estimate: $\boldsymbol{m}^{(i)} \leftarrow \beta_1 \cdot \boldsymbol{m}^{(i-1)} + (1 - \beta_1) \cdot \boldsymbol{g}^{(i)}$
    Update biased second moment estimate: $\boldsymbol{v}^{(i)} \leftarrow \beta_2 \cdot \boldsymbol{v}^{(i-1)} + (1 - \beta_2) \cdot (\boldsymbol{g}^{(i)})^2$
    Correct bias in first moment: $\hat{\boldsymbol{m}}^{(i)} \leftarrow \boldsymbol{m}^{(i)} / (1 - (\beta_1)^i)$
    Correct bias in second moment: $\hat{\boldsymbol{v}}^{(i)} \leftarrow \boldsymbol{v}^{(i)} / (1 - (\beta_2)^i)$
    Update parameters: $\boldsymbol{w}^{(i)} \leftarrow \boldsymbol{w}^{(i-1)} - \lambda \cdot \hat{\boldsymbol{m}}^{(i)} / (\sqrt{\hat{\boldsymbol{v}}^{(i)}} + \epsilon)$
  end while

2.3.4 *Overfitting*

So far, the main focus has been to optimize the model to achieve the best possible performance on the training data. However, the main goal is to obtain

a model with good generalization, such that it performs well on data which it has never seen before [39]. The problem is that the generalization is difficult to control when the model parameters are adjusted based on the training data. Thus, the model might perform very well on the training data while the generalization stops improving. This issue is a well-known problem referred to as *overfitting*, and describes the situation where the model starts to learn patterns that are specific to the training data, while the same patterns might be misleading or irrelevant when it comes to completely new unseen data [7, 34]. In the opposite case, where the model has not yet learned all relevant patterns in the training data, the model is *underfitting*. Underfitting describes the situation where the loss decreases on the training data and test data, but there is still room for progress [7, 34, 39]. The goal is therefore to find the model which has learned all relevant patterns in the data, without starting to overfit towards the training data.

The best way to deal with overfitting is to get more training data. A model trained on more data will naturally generalize better [7]. However, in many situations the amount of available training data can be very limited. Thus, other approaches which reduces the models ability to memorize patterns from the training data should be used instead.

The process of dealing with the issue of overfitting is often called *regularization*. The goal of regularization is to put some constraints on what information the model is allowed to store [39]. A model which only can memorize a small number of patterns will have a better chance of generalizing well [7]. One type of regularization is to reduce the network size. This can be done by reducing the number of layers and the number of units per layer. By reducing the network size the number of learnable parameters in the model is also reduced. Thus, the capacity of the model to memorize patterns from the training data will be limited. However, if the capacity becomes too small the model will start to underfit. Hence, the goal is to find the perfect network size which neither underfits or overfits.

There are several other methods for regularizing the model. The next section will focus on *data augmentation* and how this can increase the generalization ability of the model. For a more detailed explanation of other regularization methods the reader is encouraged to read Chapter 7 in the book *Deep Learning* written by Goodfellow et al. [39].

2.3.4.1 *Data Augmentation*

Data augmentation is a method which can mitigate overfitting even though there is limited amount of training data available [39]. The method creates new fake data and adds it to the training set. In this way the amount of training data is increased, and consequently the risk of overfitting is reduced. In some cases creating new fake data can be a challenging task. However, it can be easily done for classification problems. The goal during a classification problem is to use some high-dimensional input $\mathbf{x}$ and identify it into

a single category y. Hence, new data pairs (x, y) can easily be generated by transforming the inputs x of the training set [39].

Data augmentation has shown to be especially effective for classification tasks such as object recognition in images [39]. Images consist of a large range of factors which can be varied, such that new fake images can easily be generated. Operations such as changing the contrast, brightness, zoom, rotation or blur are examples of how one could transform the images into new fake training images. One could also flip the images, or translate the training images a few pixels in each direction. Another form of data augmentation is to apply some noise to the training images. Injection of some random noise to the input images can improve the overall robustness of the neural network [39].

All of these transformations have shown to greatly improve the generalization ability in many cases. However, when applying transformations one should be cautious and pay extra attention to the fact that the transformations do not change the correct label class.

2.3.5 *Standardization of Input Data*

When training a neural network it is important to have input data which consists of values in the same range. If the input values have very different ranges, the neural network will have problems during training [7]. In the case where images are used as input data the pixel values might differ, as discussed in Section 2.1.4. Consequently, the neural network will have problems with comparing and learning the image features. It would therefore be beneficial to transform the original values to a standard scale in most cases. In order to make it easier for the neural network, the input data should take small values in roughly the same range [7]. The standardization of input data might result in a decrease of the error and a faster algorithm [34]. Two common standardization methods are presented in the following, namely *z-score normalization* and *histogram matching*.

2.3.5.1 *Z-Score Normalization*

One common way to transform the input data is to use the z-score normalization [46]. The z-score normalization is defined as

$$z = \frac{x - \mu}{\sigma} \tag{28}$$

where x is a training sample, μ is the mean of the training samples and σ is the standard deviation of the training samples. In the case of an image x would represent a pixel value in a given image, while μ and σ would represent the mean and standard deviation of the pixel values in the training images, respectively. Equation (28) shows how the z-score normalization removes the mean and scales the sample to unit variance. It should be noted

that the z-score normalization method assumes that the data follows Gaussian distribution. Hence, the method might not be reliable if the data is not Gaussian distributed [46].

2.3.5.2 *Histogram Matching*

The z-score normalization transforms the data values to the same range. However, the distribution of values within the range might differ depending on the data sample. This is especially a problem for images, where the distribution of pixel intensities can vary a lot even though they are transformed to a standard range. This is often the case for MR images, as discussed in Section 2.1.4. The contrast and brightness in each image may differ, making it difficult for the neural network to learn. A possible solution to this problem was suggested by Nyú et al. [31]. The method is based on transforming the intensity histogram of each MR volume image into a standardized histogram. The approach is especially useful in circumstances where the images have been taken from different sources.

2.3.6 *Training, Validating and Testing*

As previously mentioned, the main goal when training a machine learning model is to find the model that generalizes well to new, unseen data. Hence, it is important to have some never-before-seen data available in order to test the generalization ability of the final model. Therefore, one should always divide the data into training data and test data [7]. The training data is used to train the models and tune the hyperparameters, while the test data is used to evaluate the generalization ability of the final model [34].

During training the configuration of the model is tuned by choosing new hyperparameters, and evaluating the corresponding model performance. Thus, the model should be tested on data which has not been a part of the training procedure, in order to evaluate the performance. Some of the training data should therefore be separated into a validation set. The validation set is used to test the model performance on new data during training. Figure 2.13 illustrates how a dataset can be split into a training set, validation set and test set. This kind of split is often referred to as the *traditional split* or the *simple hold-out validation* [7].

Figure 2.13: An illustration of how the data should be divided into a training set, validation set and test set.

K-fold cross validation is another useful way to split the dataset. In this case the dataset is split into a training set and test set, as previously explained. However, the training set is then divided into K parts of equal sizes [34]. The

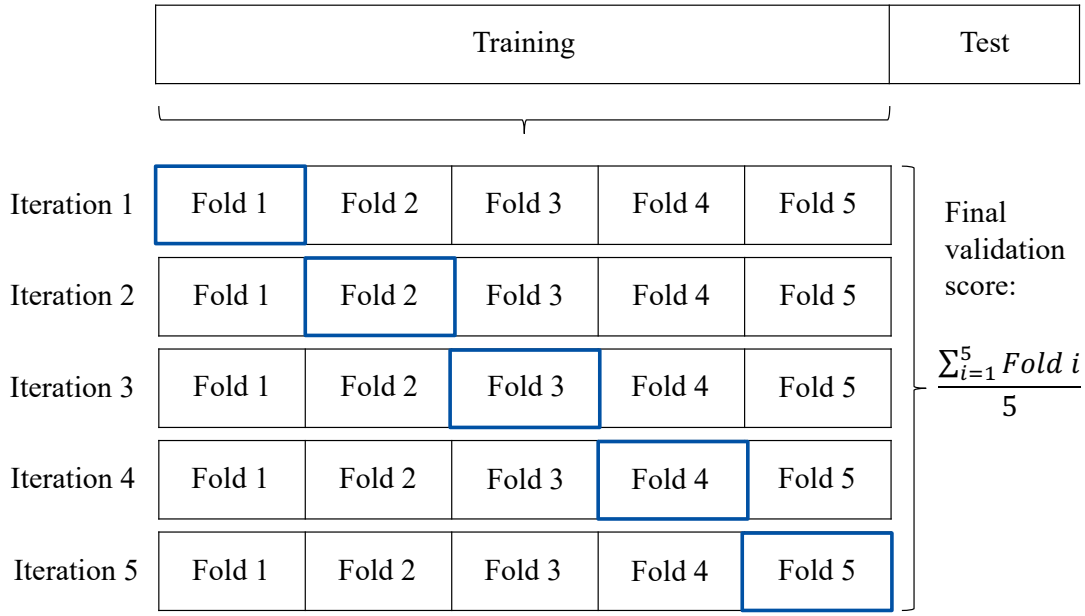


Figure 2.14: An illustration of how the training data can be divided into five folds. The fold with a blue frame is used as validation data, and each fold should be used as validation data once. The final validation score is calculated as the average score over all iterations. In this case there are five iterations, since there are five folds.

model is trained on $K-1$ parts, and validated on the last part. An illustration of how the training set can be divided into $K = 5$ parts is given in Figure 2.14. In this case the K-fold cross validation is also known as 5-fold cross validation. Each part should be used as validation set once, and the final model score is calculated as the average of the K obtained scores. Another popular variant of the K-fold cross validation method is the Leave One Out Cross Validation (LOOCV). In the LOOCV approach one data point is left out of the training data, and is instead used as validation data [34]. This is repeated for all of the data points, such that all data points have been used as validation data once. Consequently, more data is available for training. Thus, the K-fold cross validation method is especially useful for small datasets, where the amount of available training data is limited [39, 47].

The reason why a validation set is introduced in order to test the model performance for each configuration instead of using the test set each time, is mainly due to the phenomenon known as *information leaks*. Information leaks occur when the hyperparameters of the model are tuned based on the model's performance on the validation set [7]. Each time one is running an experiment, evaluating the model on the validation set, and modifying the model as a result, some information about the validation set will leak into the model [7]. If this process is repeated several times the model will perform artificially well on the validation data, because this is what it has been optimized for. Thus, the model is overfitting to the validation set. It is therefore important that the final model is tested on data which the model has

no information about, even indirectly. Consequently, if anything about the model has been tuned based on the test set performance, then the measure of generalization ability will be inaccurate [7].

2.3.7 *Performance Metrics*

During validation of the model the performance can be evaluated in different ways. For a binary classification problem the *confusion matrix* is a popular tool to use in order to measure the model performance [48]. Figure 2.15 shows an example of a confusion matrix when the classification problem is binary. The figure illustrates how the confusion matrix represents the number of True Positive (TP), False Positive (FP), False Negative (FN) and True Negative (TN) when comparing the predicted class from the model with the true target class.

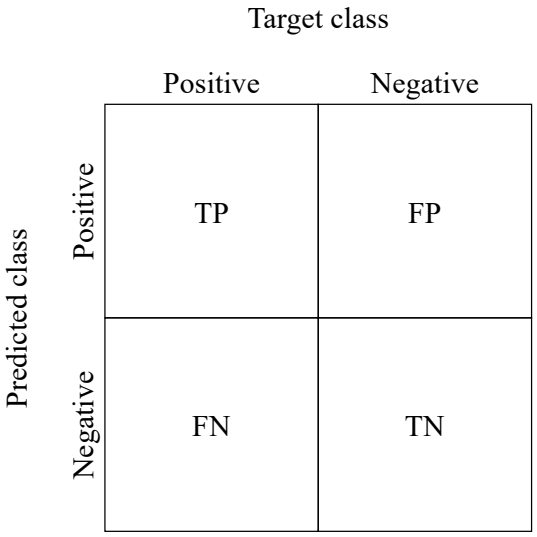


Figure 2.15: Confusion matrix for a binary classification problem. TP gives the number of True Positives, FP gives the number of False Positives, FN gives the number of False Negatives and TN gives the number of True Negatives when comparing the predicted class and target class.

There are several useful performance metrics which can be defined through the confusion matrix. Two examples are the Error (ERR) and Accuracy (ACC), which provide general information about the amount of misclassified samples. The ERR can be understood as the sum of all false predictions divided by the number of total predictions, while the ACC is the sum of correct predictions divided by the total number of predictions [49]. Equation (29) and (30) show how the error and accuracy can be computed when using the confusion matrix.

$$\mathrm{ERR} = \frac{\mathrm{FP} + \mathrm{FN}}{\mathrm{FP} + \mathrm{FN} + \mathrm{TP} + \mathrm{TN}} \tag{29}$$

$$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{FP} + \mathrm{FN} + \mathrm{TP} + \mathrm{TN}} \tag{30}$$

Furthermore, the True Positive Rate (TPR) and False Positive Rate (FPR) are especially useful to look at for imbalanced class problems [49]. The TPR and FPR are defined as

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{FN} + \mathrm{TP}} \tag{31}$$

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} \tag{32}$$

The TPR gives the ratio between the correctly predicted positive observations of the model and the total number of positive observations in the target class, while the FPR gives the ratio between the misclassified positive predictions of the model and the total number of negative observations [49]. These metrics provide useful information of the degree of misclassification within each class.

The Precision (PRE) of the model is another convenient performance metric which gives the ratio between the correctly predicted positive observations and the total number of predicted positive observations. The PRE is given as

$$\mathrm{PRE} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{33}$$

2.3.7.1 *The Dice Similarity Coefficient*

The Dice Similarity Coefficient (DSC) is a useful performance metric when comparing the overlap between two regions. Hence, the DSC is one of the most common statistical validation metrics used to evaluate the similarity between manual and automatic segmentations [50]. The DSC is defined as

$$\mathrm{DSC}(X, Y) = 2 \cdot \frac{|X \cap Y|}{|X| + |Y|} \tag{34}$$

where X and Y are the given target regions. The calculated DSC value ranges from 0 to 1, where 1 corresponds to a complete overlap of the two target regions [51]. In terms of the confusion matrix the DSC can be expressed as

$$\mathrm{DSC} = 2\frac{\mathrm{PRE} \times \mathrm{TPR}}{\mathrm{PRE} + \mathrm{TPR}} = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FP} + \mathrm{FN}} \tag{35}$$

One can notice that a complete overlap of the two target regions will give $\mathrm{FP} = \mathrm{FN} = 0$, thus $\mathrm{DSC} = 1$. In the opposite case, where there is no overlap between the two target regions, the TP will be equal to zero and consequently the DSC will be equal to zero.

2.3.7.2 F_β-measure

The DSC is a special case of the more general performance metric called the F_β-measure. The F_β-measure [52] is a weighted harmonic average, defined as

$$F_\beta = \frac{1+\beta^2}{\frac{\beta^2}{TPR}+\frac{1}{PRE}} = \frac{(1+\beta^2)PRE \times TPR}{\beta^2 PRE + TPR} = \frac{(1+\beta^2)TP}{(1+\beta^2)TP + \beta^2 FN + FP} \tag{36}$$

where β is the weighting variable which decides how much emphasis one should put on PRE and TPR. Equation (35) is retrieved when $\beta = 1$. Hence, the DSC is also commonly referred to as the F1-score.

2.4 DEEP LEARNING FOR IMAGE SEGMENTATION

For a long time the idea of enabling computers to recognize objects in images was thought to be a very difficult task. However, a special kind of deep learning network known as a Convolutional Neural Network (CNN) has shown to be especially effective in computer vision tasks [7]. In the last few years CNNs have shown to exceed human performance on the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC). These special networks have therefore turned into the standard network structure for a wide variety of computer vision problems [7].

Also in medical imaging the interest in CNNs has increased significantly. The CNNs ability to learn useful representations of images and other structured data in a highly efficient way, have made them very useful in several medical imaging problems. Today, deep learning is applied to many central problems such as brain segmentation [10], breast cancer segmentation [11, 12] and radiomics [13, 14]. In the following sections we will have a closer look at CNNs and how they work.

2.4.1 *Convolutional Neural Networks*

The main difference between a neural network and a CNN is that a CNN uses convolution instead of general matrix multiplication in at least one of their layers [39]. The convolution operator is denoted as $*$, and the definition is given as

$$(f * g)(t) = \int_{-\infty}^{\infty} f(a)g(t-a)da \tag{37}$$

where f and g are two functions. In the discrete case equation (37) can be re-written as

$$(f * g)(t) = \sum_{a=-\infty}^{\infty} f(a)g(t-a) \tag{38}$$

In terms of CNNs the first argument of equation (37) is often referred to as the *input*, while the second argument is referred to as the *kernel*. The kernel can also be referred to as the *filter* in several occasions, while the output can sometimes be called the *feature map* [39]. If the input is a two-dimensional image I, then the kernel K needs to be two-dimensional. Equation (39) shows how equation (38) can be written in the two-dimensional case.

$$(I * K)(i,j) = \sum_{m} \sum_{n} I(m,n)K(i-m, j-n) \tag{39}$$

Figure 2.16 gives a visual illustration of how the convolutional layer in a CNN works in the two-dimensional case. As illustrated in Figure 2.16 the kernel size is smaller than the input size. Hence, a given output unit in a convolutional layer interacts only with a limited number of input units, which is defined by the kernel size. The convolutional layers are therefore said to have *sparse interactions*, and consequently the convolutional layers learn local patterns within the kernel window [39]. This is in contrast with the traditional neural networks, where every output unit is interacting with every input unit. Thus, the traditional neural networks learn global patterns instead of local patterns. Since the convolutional layers have sparse interactions the number of parameters in the network is greatly reduced. Consequently, a CNN is more robust against overfitting than a traditional neural network since it has fewer parameters. Besides, the reduced number of parameters means that less memory is required during computation and that the computations require fewer operations [39].

Another advantage with CNNs is that the kernels are translational equivariant at each layer. Hence, they share the exact same weights across the whole input domain [53]. This means that if a certain pattern is learned in one location of the input domain a CNN can recognize it anywhere. A traditional neural network on the other hand would have to re-learn the pattern if it appears at a new location in the input domain. Consequently, CNNs need fewer training samples to recognize patterns with high generalization ability [7].

One more useful property of CNNs is that they are able to learn spatial hierarchies of patterns [7]. Thus, the first convolutional layer will be able to learn small and simple patterns such as edges, while the next convolutional layer can learn larger and more complex patterns based on the features of the first layer. Hence, the more convolutional layers a CNN has, the more complex and abstract visual concepts it can learn.

In Figure 2.16 one can notice that the output size differs from the input size. The output size from the convolutional layers is determined by the kernel size and *stride*. A larger kernel size will connect more of the input units to a given output unit, thus a larger kernel size will give a smaller output size. The stride is a parameter which determines the distance between two successive windows of the convolution process [7, 54]. Consequently, by using a higher stride the output is downsampled by a higher factor which result in a smaller output size [7]. However, in some cases one might want to have the same spatial dimensions as the input. *Padding* has therefore been introduced in order to deal with the shrinking dimension problem. The method adds an appropriate number of rows and columns on each side of the input such that it is possible to fit the center of the convolution windows around every input tile [7, 54]. It is most common to pad the input with zeros, as illustrated in Figure 2.17. A larger kernel size requires more padding of the input in order to get the same output size as the input size.

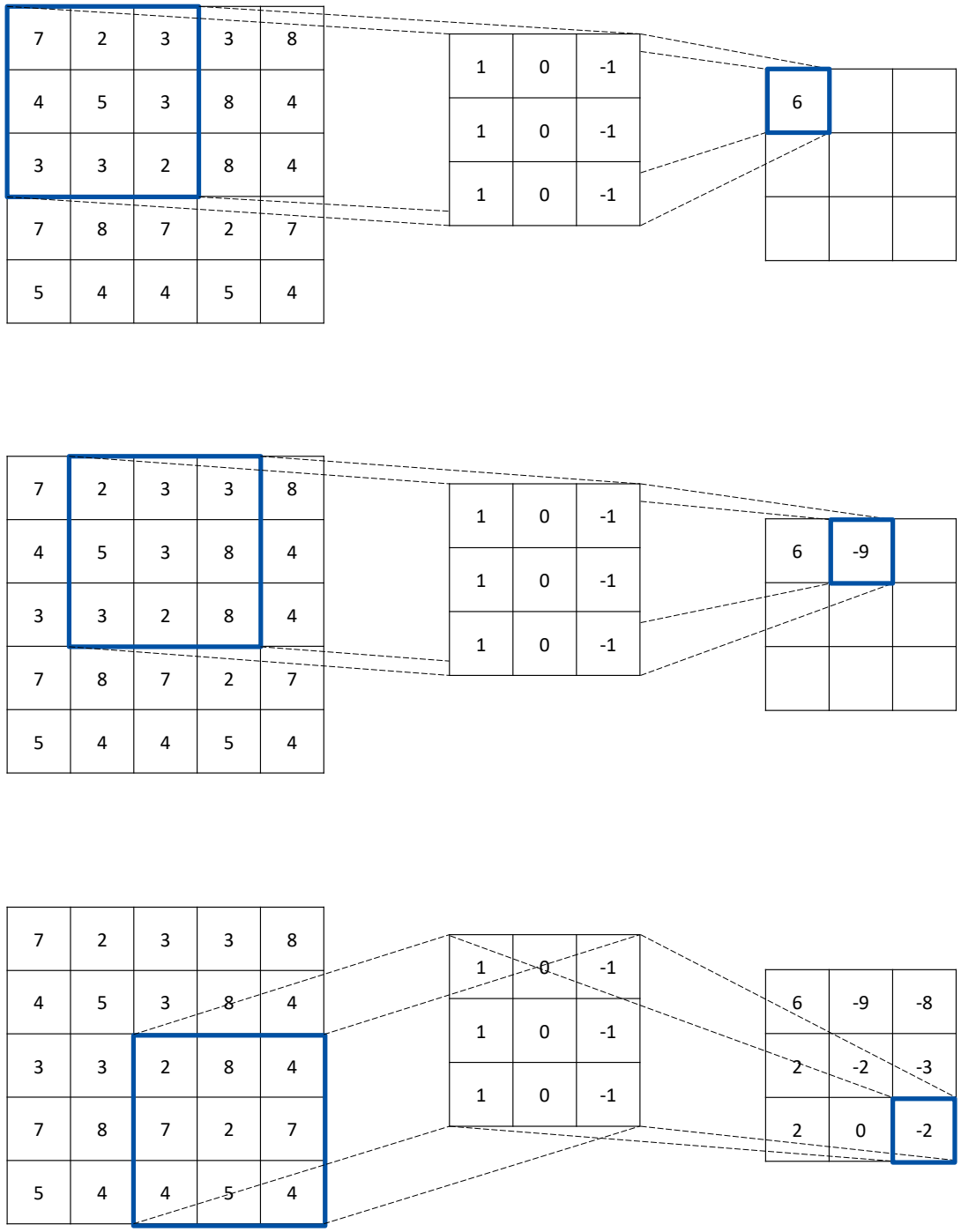


Figure 2.16: The figure shows how a convolutional layer works in a CNN. The two-dimensional input on the left is convolved with a kernel of size 3×3 and with a stride of 1. The corresponding output, also referred to as the feature map, is shown to the right.

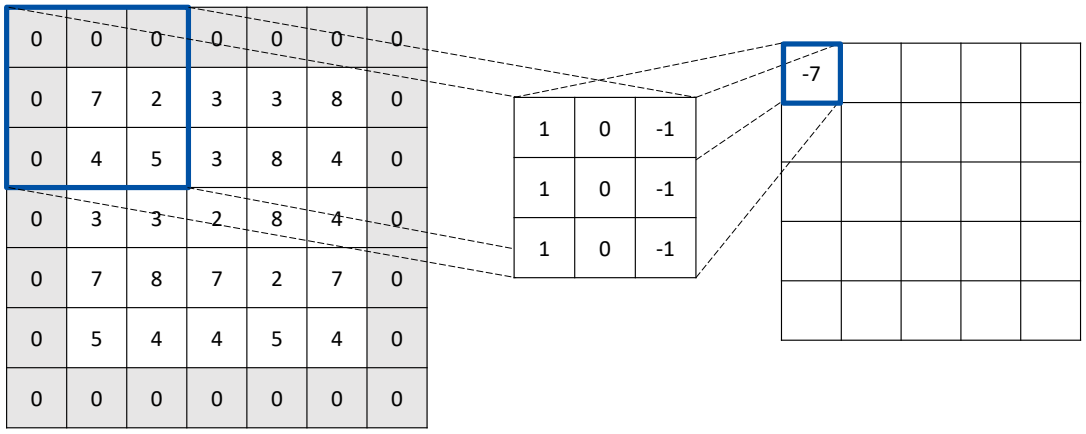


Figure 2.17: An illustration of how the concept of padding works. In this case the input has been padded with zeros in all directions such that the dimensions is equal to 7×7. With a kernel size of 3×3 the output obtains a size of 5×5, which is equal to the original input dimension.

In several CNNs *pooling* layers are used to downsample the output and help make the representation approximately invariant to small translational changes of the input [7, 39]. Hence, the layers are especially useful if one is interested in whether some feature is present rather than exactly where it is. By downsampling the output the number of feature map coefficients to process is greatly reduced. Thus, this is an essential step to mitigate overfitting. In addition, the pooling layers are useful to give some information of the totality of the input [7].

When using pooling it is common to use a kernel size of 2×2, with a stride of 2 [7]. The pooling process consists of extracting some information from the input with where the kernel window is at. The most frequently used pooling function is the *max pooling* operator. The concept of max pooling is illustrated in Figure 2.18, where the maximum value within the kernel window is extracted and used as output. There are also other pooling operations such as *average pooling* and *weighted average pooling* [39]. However, the max pooling operator has proven to be the most successful solution. This is due to the fact that it is more informative to look at the maximal presence of different features rather than their average presence [7].

Pooling layers are also important when dealing with input of different sizes. In the case of image classification the input image to the classification layer should have a fixed size. This can be accomplished by changing the offset size of the pooling regions such that the classification layer receives the same information regardless of the input size [39].

2.4.2 *The U-Net Architecture*

As previously introduced, CNNs have shown to be especially effective in computer vision tasks which includes object recognition and classification. How-

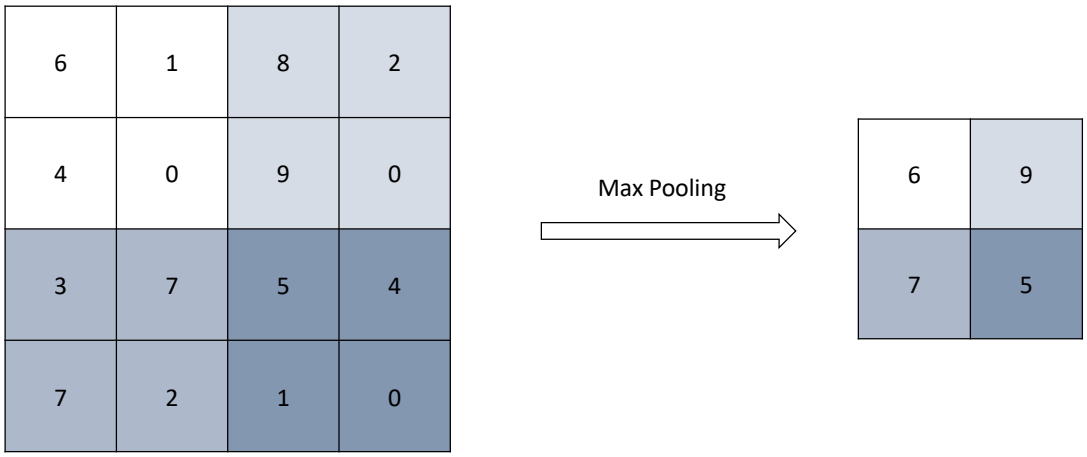


Figure 2.18: An illustration of how the concept of max pooling works. A kernel size of 2×2, and a stride of 2 is used. The maximum value within the kernel window is extracted and used as output.

ever, in the case of biomedical image tasks there is a higher demand of accuracy than for natural images. This is due to the fact that it is especially important to be able to distinguish between interesting areas in an image, such as tumor regions or organs [55]. Thus, biomedical image segmentation often consists of *semantic segmentation*. Semantic segmentation is the process where the model is predicting a category for each pixel in the image and outputs a pixelwise mask for each object in the image [55]. Hence, every pixel in the input image should be assigned to a class and the output of the network should have the same resolution as the input. However, this is not possible with a network consisting of only convolutional layers, since the convolution operation reduces the size of the feature maps as illustrated in Figure 2.16. In addition, the CNNs require a lot of training data in order to avoid overfitting [56].

A possible solution to these problems were suggested by Ronneberger et al. [56], where a new architecture type for image segmentation was proposed. Figure 2.19 illustrates the network architecture. The suggested architecture type consisted of two parts. The first part was a contracting path which followed the typical architecture of a CNN. This part consisted of convolutional layers followed by ReLU units and max pooling operations. The second part was an expansive path which mirrored the contracting path. It consisted of up-convolution layers which mapped each pixel in the input to four pixels in the output. The expansive path was completed by two convolutional layers and a ReLU unit. The combination of a contractive and expansive path creates a "U"-shape when visualized, and the architecture type was therefore called the U-Net architecture. The network has shown good performance on different biomedical segmentation applications, especially when combined with data augmentation [56].

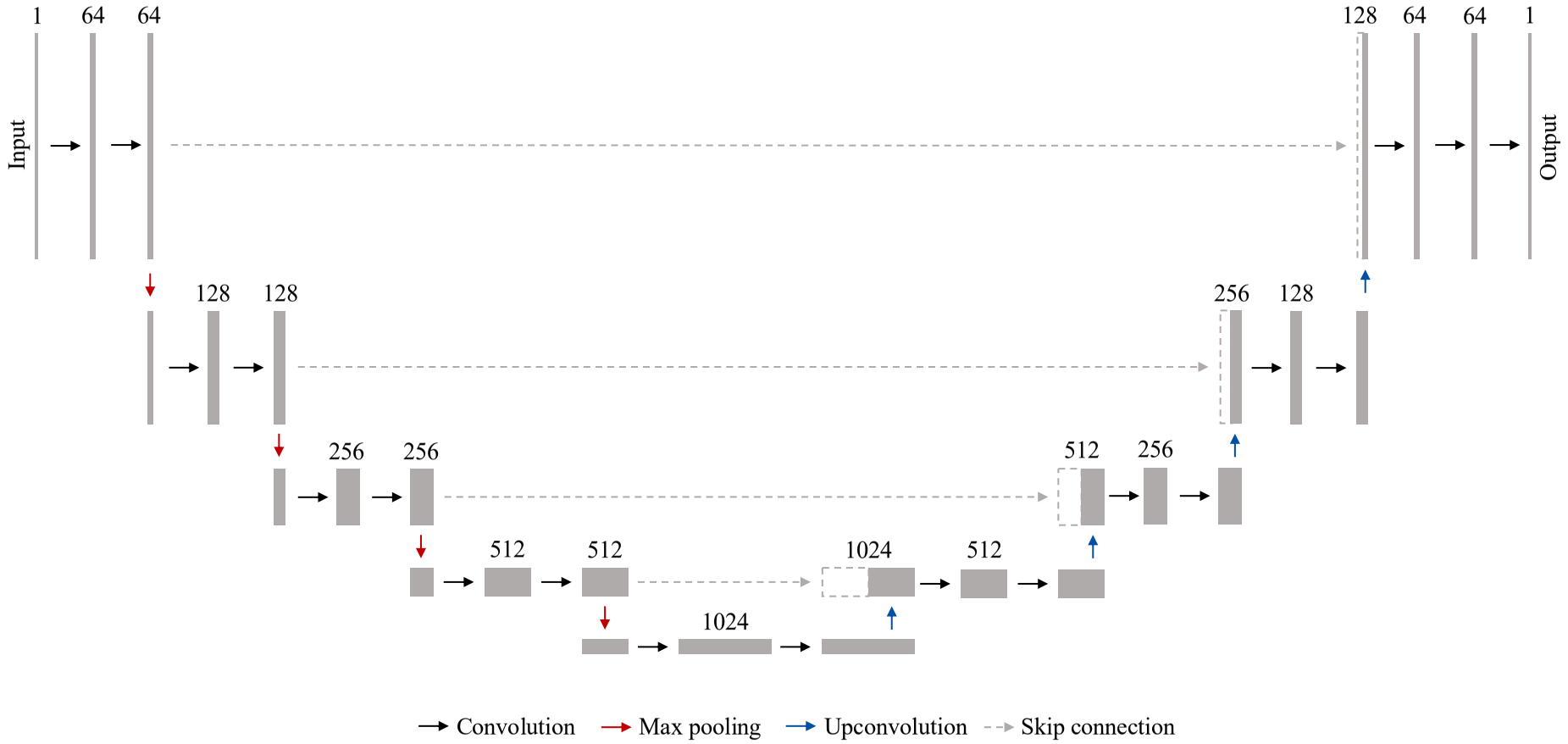


Figure 2.19: Illustration of the U-Net architecture [56]. The gray boxes represent the input and output layers, where the height represents the spatial size. The number of channels in each layer is denoted above each box and is also represented by the box length. The black arrows illustrate convolutional layers with a kernel size of 3×3, while the red arrows represent max pooling layers with a kernel size of 2×2. The blue arrows illustrate the upconvolutional layers which maps each pixel in the input to four pixels in the output. Skip connections are marked as gray dashed arrows. These connections concatenate the inputs to the max pooling layers to the output of the upsampling layers. The copied layers which are sent through the skip connections are represented as gray dashed boxes.

3 MATERIALS AND METHODS

In the following chapter the materials and methods used during the thesis are presented. Section 3.1, 3.2 and 3.3 presents the datasets, while Section 3.4 provides a detailed explanation of the pre-processing steps. The pre-processing consisted of image cropping, splitting of the datasets, standardizing of the images and conversion to the Hierarchical Data Format version 5 (HDF5) file format. The Deep Learning (DL) model and its various parameters are presented in Section 3.5, while Section 3.6 provides information about the code and software used in the thesis. Section 3.7 describes how the models were analyzed and evaluated, while the Shallow Machine Learning (SML) model used for comparison of the DL model is presented in Section 3.8. Finally, an overview of the experimental setup with an explanation of the motivation behind each experiment, is given in Section 3.9.

3.1 THE LARC-RRP STUDY

The first set of images used in this thesis was from patients participating in the prospective non-randomized study *Locally Advanced Rectal Cancer - Radiation Response Prediction* (LARC-RRP) [ClinicalTrials NCT00278694]. The study enrolled a total of 113 patients from September 2005 to March 2010. The treatment protocol included neoadjuvant chemotherapy and radiation followed by surgery and then no further treatment [57].

Out of the 113 patients enrolled, T2w images from 89 patients were used in this thesis. A 1.5-T GE Signa® LS scanner (GE Healthcare, Milwaukee, WI), which gave voxel sizes equal to $(0.391, 0.391, 5.0)$ mm, was used to image 55 of the study patients. Due to the upgrading of the GE MRI scanner, the last 34 study patients were imaged using a 1.5T Siemens Espree scanner (Siemens, Erlangen, Germany). These patients had voxel sizes equal to $(0.375, 0.375, 5.0)$ mm. The same scanner was always used for the same patient [58]. A radiologist ($\text{Radiologist}_\text{L}$) delineated the tumor on the T2w images, and these delineations were used as ground truth during training and evaluation of the model.

3.2 THE OXYTARGET STUDY

The second set of images used in this thesis was from patients participating in *The OxyTarget study - Functional MRI of Hypoxia-Mediated Rectal Cancer Aggressiveness* (OxyTarget) [ClinicalTrials NCT01816607]. The study enrolled a total of 192 patients from October 2013 to December 2017. Study participation was offered to all rectal cancer patients treated at Akershus University Hospital. The study aimed to identify novel imaging biomarkers of hypoxia-induced

rectal cancer aggressiveness. This is useful in order to reliably predict patients with poor response to chemoradiotherapy and high risk of poor metastasis-free survival at time of diagnosis [59].

Out of the 192 patients enrolled, T2w images from 110 patients were used in this thesis. In addition, DWIs were acquired from 109 out of the 110 patients. The study patients were imaged using a Philips Achieva 1.5-T system (Philips Healthcare, Best, the Netherlands) [60]. Consequently, the T2w images had voxel sizes equal to $(0.352, 0.352, 2.75)$ mm. One radiologist ($\text{Radiologist}_{\text{O}}^{1}$) delineated the tumors on the T2w images, and these delineations were used as the ground truth during training and evaluation of the model. A second radiologist ($\text{Radiologist}_{\text{O}}^{2}$) delineated the tumors in 76 of the 110 patients in the T2w images. The delineations made by $\text{Radiologist}_{\text{O}}^{2}$ were used as a second ground truth in order to further evaluate the model performance.

3.3 DATASETS

The LARC-RRP and OxyTarget studies were used to create three different datasets. The first dataset consisted of patients solely from the OxyTarget study. T2w images from 110 patients, which were delineated by $\text{Radiologist}_{\text{O}}^{1}$, were used for training and evaluation. In some cases, DWIs from 109 out of the 110 patients were used as additional input during training. The second dataset consisted of patients solely from the LARC-RRP study. This dataset had a total of 89 patients with T2w images delineated by $\text{Radiologist}_{\text{L}}$. The third dataset consisted of a combination of patients from the OxyTarget study and the LARC-RRP study. This dataset will from now on be referred to as the Combined dataset, and consisted of 199 patients with T2w images. $\text{Radiologist}_{\text{L}}$ delineated the LARC-RRP patients of the dataset, while $\text{Radiologist}_{\text{O}}^{1}$ delineated the OxyTarget patients of the dataset.

Table 3.1 gives an overview of the datasets used in this thesis, while Figure 3.1 shows an example of a T2w image for a patient with rectal cancer where the manual tumor delineation is marked in yellow.

Table 3.1: Overview of the datasets used in this thesis. The Combined dataset consisted of a combination of patients from the LARC-RRP dataset and OxyTarget dataset, where one radiologist delineated the LARC-RRP patients ($\text{Radiologist}_\text{L}$) and another radiologist ($\text{Radiologist}^1_\text{O}$) delineated the OxyTarget patients. $\text{Radiologist}^2_\text{O}$ represents a second radiologist which delineated 76 of the OxyTarget patients. These delineations were not used during training, but were used as an additional measure to evaluate the model performance.

Dataset	Patients	Images	Number of Image Slices	Delineation
OxyTarget	110	T2w	2809	$\text{Radiologist}^1_\text{O}$
	109	DWI	2783	$\text{Radiologist}^1_\text{O}$
	76	T2w	1990	$\text{Radiologist}^2_\text{O}$
LARC-RRP	89	T2w	3133	$\text{Radiologist}_\text{L}$
Combined	199	T2w	5942	$\text{Radiologist}_\text{L}$, $\text{Radiologist}^1_\text{O}$

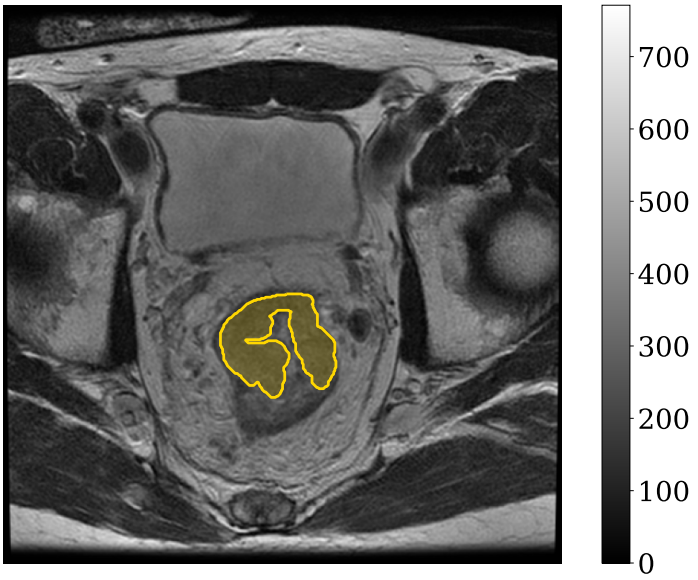


Figure 3.1: Example of a T2w image from a patient with rectal cancer. The manual tumor delineation made by the radiologist is marked in yellow. The color bar indicates the image intensities.

3.4 PRE-PROCESSING

The T2w images in the datasets were originally stored using the Digital Imaging and Communications in Medicine (DICOM) file format. The DICOM file format is considered the communications standard for medical imaging, and is used to store, exchange and transmit medical images. Today it is widely used in several hospitals and clinics [61]. During the author's project thesis the LARC-RRP images were de-identified using the *PixelMed DICOM Cleaner* software [62], and converted to the Neuroimaging Informatics Technology Initiative (NIfTI) file format. The reader is encouraged to study the author's project thesis for a more detailed explanation of the conversion process [63]. The NIfTI file format stores the images as a single file containing both the header metadata and the pixel data [64]. Thus, the NIfTI file format is a convenient way to store data obtained during MRI. The following subsections provide a description of how the data was further pre-processed during the master thesis.

3.4.1 *Cropping of Images*

The original data consisted mainly of images with a size equal to 512×512. However, as illustrated in Figure 3.2a, some of the images had different sizes. Hence, the images were cropped to a standard size of 352×352 in order to reduce the size of the data, and to reduce the varying image dimensions. The standard size of 352×352 was chosen based on the smallest possible image size which did not exclude any tumor voxels in the images. It was also important that the image dimensions were divisible by 16, in order to meet the criteria of the DL network. The cropping was done by removing an equal amount of pixels at all edges of the image. During the image cropping the number of tumor voxels in the given image was calculated. If the number of tumor voxels was reduced during the cropping, the process was repeated with a new starting point and ending point away from the original image edges. In this way it was ensured that none of the tumor voxels were excluded. An example of how a T2w image looked before and after cropping is presented in Figure 3.3.

Figure 3.2b shows that images from six patients were not formatted to the standard size of 352×352. These patients had an image size equal to 256×256, which was smaller than the standard size. However, it was concluded that the DL model could handle images of two different sizes. This was done by changing the stride of the up-convolutional layers to 2 instead of 1. See Section 2.4.1 for more information on pooling layers and how different strides can help the network handle images of different sizes. The images of the patients with a size of 256×256 were therefore not padded in order to meet the standard dimension of 352×352, but where left as they were originally.

In some experiments it was decided to only use image slices which contained tumor as input. This was done due to the imbalance of image slices with and

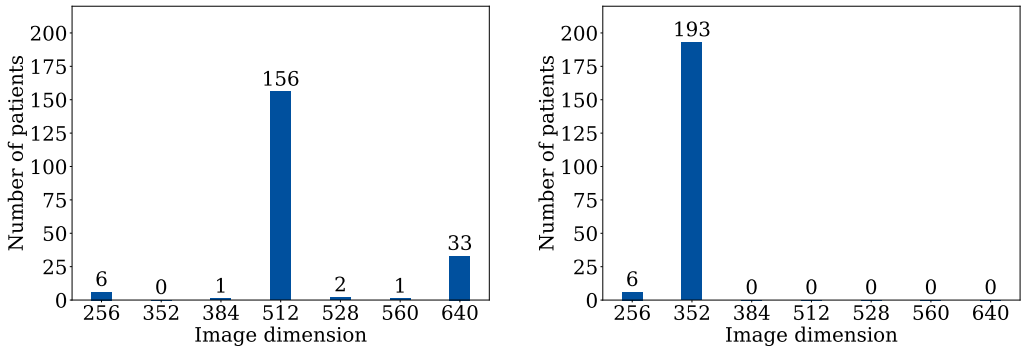


Figure 3.2: The histograms illustrate the distribution of image dimensions in the data before (a) and after (b) cropping. The numbers along the x-axis represents the image dimension in both x- and y-direction.

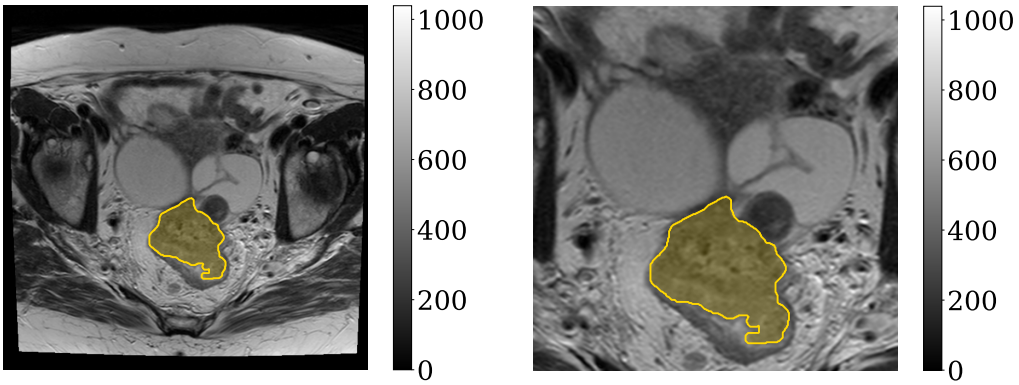


Figure 3.3: Example of how a T2w image looked before (a) and after (b) cropping. The manual delineation made by the radiologist is given in yellow. The color bar indicates the image intensities.

without tumor, as presented in Table 3.2. Thus, all datasets were duplicated and the image slices without any tumor were removed.

Furthermore, DWIs were used as additional input for the OxyTarget dataset in the last few experiments. The DWIs and T2w images were acquired with slightly different slice thickness and with different number of slices. In addition, the two image modalities were acquired on different grids. The DWIs were therefore rigidly registered towards the grid of the T2w images by Franziska Knuth[1]. Consequently, some image slices in the DWIs did not cover the entire tumor when positioning the DWIs and T2w images on the same grid. The image slices where the DWI did not cover the entire tumor were therefore removed. Table 3.2 provides an overview of the number of image slices in each dataset for the different input scenarios.

[1]PhD Candidate at the Department of Physics, NTNU

Table 3.2: Overview of the number of image slices in each dataset.

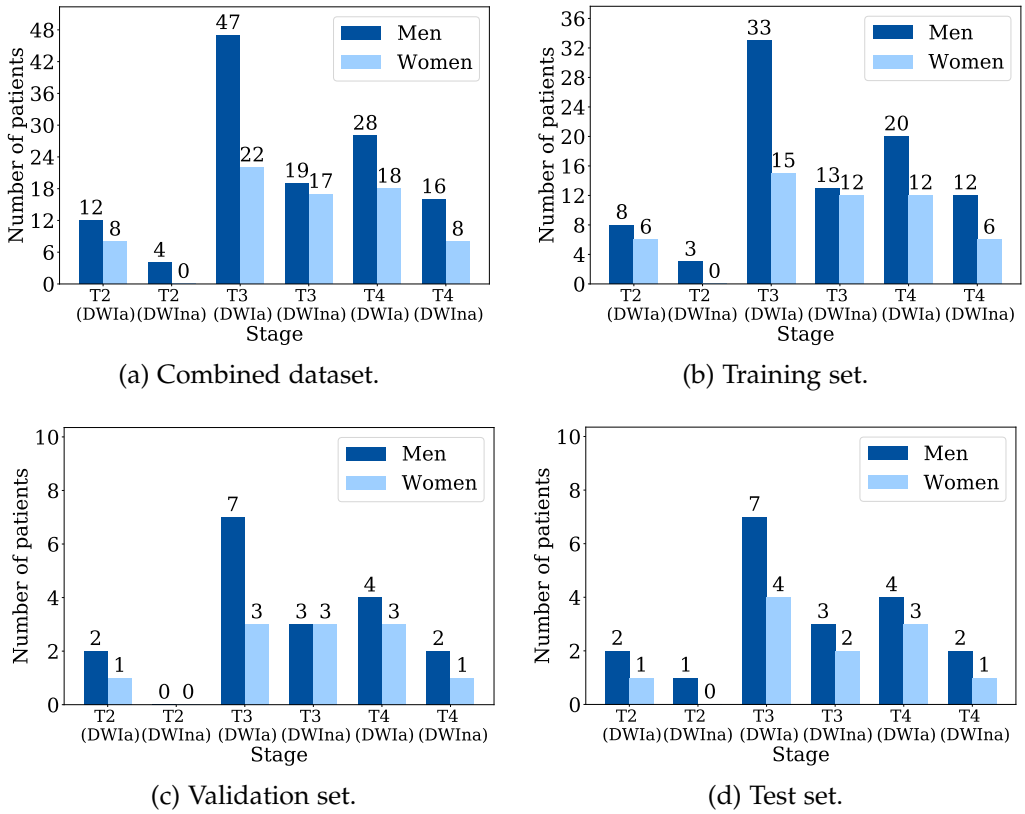
Dataset	T2w (All Images)	T2w (Tumor Images)	T2w + DWI (Tumor Images)
OxyTarget	2791	1988	1826
LARC-RRP	3133	917	
Combined	5942	2905	

3.4.2 *Splitting into Training, Validation and Test Sets*

The datasets were each divided into a training set, validation set and test set. In this way the model was trained on the training data, while predictions were made on the validation data. The Dice Similarity Coefficient (DSC) between the model predictions on the validation set and the corresponding manual delineations made by the radiologist was calculated during training. The hyperparameters were then tuned to achieve the highest possible DSC. Finally, the test set was stored and used to evaluate the model performance on completely new and unseen data. A more detailed explanation of the traditional splitting method is provided in Section 2.3.6.

To make sure that all subsets contained a representative selection of the patients, the data was stratified based on gender, tumor stage and availability of DWI. The tumor stage was categorised according to the national guidelines for rectal cancer [4] into T1, T2, T3 or T4, depending on how deeply the tumor had grown into the bowel lining of the patient. In the T1 stage the tumor had grown into the submucosa, which is the lining of the colon. In the T2 stage the tumor had grown into the muscularis propria, which is a deeper, thick layer of muscle outside of the submucosa. In the T3 stage the tumor had grown through the muscularis propria and into the subserosia. In the T4 stage the tumor had grown either through all layers of the colon (T4a) or into surrounding organs (T4b) [65]. The datasets consisted of patients with tumor stage T2, T3 and T4.

The availability of DWI was defined as DWIa or DWIna. DWIa represented patients where DWI was available, while DWIna represented patients where DWI was not available. Hence, the patients could be divided into 12 different groups. Figure 3.4 shows how the Combined dataset was distributed into a training set, validation set and test set. It should be noted that the Combined dataset was stratified by combining the stratified splits of the OxyTarget data and LARC-RRP data. Hence, the Combined dataset was not stratified from scratch. An overview of the corresponding percentage of patients, number of patients and number of image slices in each subset is given in Table 3.3. Information of how the OxyTarget dataset and LARC-RRP dataset were divided can be found in Appendix A.1 and A.2, respectively.



(a) Combined dataset.

(b) Training set.

(c) Validation set.

(d) Test set.

Figure 3.4: The histograms illustrate how the patients in the Combined dataset (a) were distributed into a training set (b), validation set (c) and test set (d).

The K-fold cross validation method was used as an additional step to evaluate whether or not the model performance depends on how the datasets were divided. Hence, the training set and validation set were combined and split into five folds. The folds were stratified to make sure that all patient groups were equally represented in each fold. The model was trained on four of the folds, while one fold was used as validation set. This was repeated five times such that all folds were used as validation set once. A more detailed explanation of the K-fold cross validation is given in Section 2.3.6. Table 3.4 presents an overview of how the Combined dataset was divided into five folds during K-fold cross validation. Further information of how the OxyTarget dataset and the LARC-RRP dataset were divided into five folds is given in Appendix A.3 and A.4, respectively.

Table 3.3: Overview of how the patients and image slices in the Combined dataset were distributed into a training set, validation set and test set.

	Percentage of Patients	Number of Patients	Number of Image Slices
Train	70 %	139	4235
Validation	15 %	30	820
Test	15 %	30	887
Total	100 %	199	5942

Table 3.4: Overview of how the patients and image slices in the Combined dataset were distributed into five folds. The total number of patients is a combination of the training split and validation split from Table 3.3.

	Percentage of patients	Number of patients	Number of image slices
Fold 1	20 %	34	988
Fold 2	20 %	34	1066
Fold 3	20 %	34	1025
Fold 4	20 %	34	955
Fold 5	20 %	33	1021
Total	100 %	169	5055

3.4.3 *Conversion to the Hierarchical Data Format Version 5 File Format*

The Hierarchical Data Format version 5 (HDF5) file format is a convenient way to store large quantities of numerical data. The file format has a hierarchical structure of groups and attributes. The groups make it possible to store related datasets together, like folders in a filesystem. The user-defined attributes make it possible to attach descriptive metadata directly to the data it describes [66]. In addition to the useful hierarchical structure, one of the HDF5 files greatest strengths is its ability to load the appropriate data into memory of the computer when needed. Hence, the large data volumes are stored on disk until it is required. When the data is required only the appropriate data will be loaded into memory. In this way the computer is able to handle large amount of data without running out of memory when using the HDF5 file format [66]. These are some of the main reasons why the datasets were converted from NIfTI files to HDF5 files before they were used as input for the DL model.

For each of the datasets five HDF5 files were created, as presented in Table 3.5. Four out of the five HDF5 files were created based on the traditional split

method, and the structure is illustrated in Figure 3.5. These HDF5 files consisted of three groups; one for the training set, one for the validation set and one for the test set. Each group had one or two subgroups, depending on the number of images with different sizes in the datasets. The subgroup named "352" consisted of images with the standard size of 352×352. The subgroup named "256" consisted of images with size equal to 256×256. All of the subgroups contained three datasets. The structure of these datasets is shown in Table 3.6. The dataset named "input" consisted of all of the image slices for the patients in the group, "target_an" consisted of the tumor delineation made by the radiologist, while "patient_ids" consisted of the patient IDs for all of the patients in the group.

One of the HDF5 files was instead created based on the 5-fold cross validation method. Hence, these files consisted of five groups, one for each fold. Each fold contained the same subgroups and datasets as illustrated in Figure 3.5.

Table 3.5: Summary of the HDF5 files created for each dataset during the thesis. The dataset was split either according to the traditional splitting method (Traditional), or according to the 5-fold cross validation method (5-fold). The input images were standardized in four different ways; no standardization (No), z-score normalization (Z-Score), matching of pixel histograms (MH) or by a combination of z-score normalization and matching of pixel histograms (MH + Z-Score).

HDF5 File Number	Split Method	Standardization of Input Images
1	5-Fold	No
2	Traditional	No
3	Traditional	Z-Score
4	Traditional	MH
5	Traditional	MH + Z-Score

Table 3.6: Structure of the datasets in the HDF5 files. In this case `n_images` is the number of images in the dataset, `dim_x` is the number of pixels in the x-direction, `dim_y` is the number of pixels in the y-direction, `channels` is the number of image modalities, while `masks` is the number of mask modalities. In this thesis only one mask modality (one radiologist) was used during training. Hence, the `masks` variable was equal to 1.

Dataset	Shape	Content
input	`(n_images, dim_x, dim_y, channels)`	Input images
target_an	`(n_images, dim_x, dim_y, masks)`	Manually delineated masks
patient_ids	`(n_images)`	Patient ID numbers

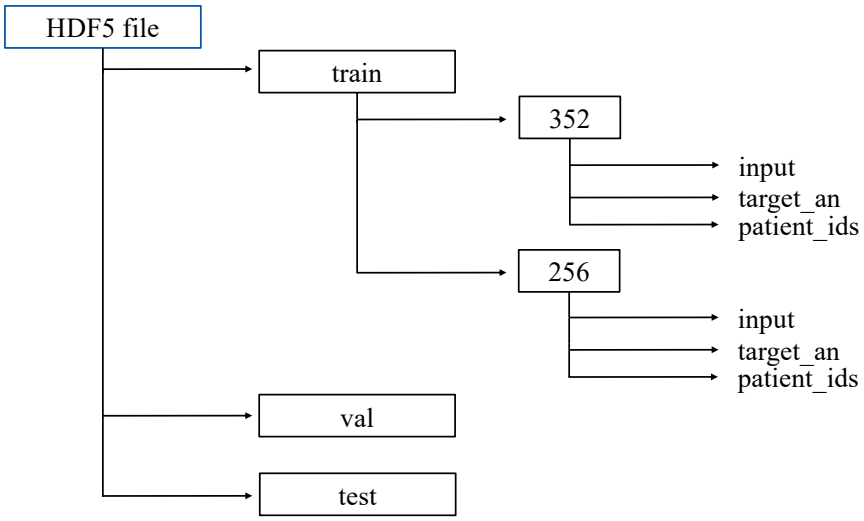


Figure 3.5: Illustration of how the HDF5 file was structured when the file was created based on the traditional split method. The "train", "val" and "test" groups contained the training set, validation set and test set respectively. The subgroups "352" and "256" signify whether the images had a size equal to 352×352 or 256×256. The "input", "target_an" and "input_ids" are datasets containing image slices used as input, tumor delineated made by the radiologist and patient IDs. All of the groups had the same structure as illustrated for the "train" group.

The input images in the HDF5 files were standardized in different ways, as shown in Table 3.5. For two of the HDF5 files the input images were not normalized or standardized in any way. However, the third, fourth and fifth HDF5 files were normalized and standardized in various ways. The images in the third HDF5 file were normalized using the z-score normalization method (Z-Score). The images in the fourth HDF5 file were standardized by matching the pixel histograms (MH). For the last HDF5 file the images were standardized by combining the z-score normalization and the pixel histogram matching (MH + Z-Score). See Section 3.4.4 for a more detailed explanation of how the input images were standardized.

In the final part of the thesis it was decided to train the model only using image slices which contained tumor. Hence, the HDF5 files $2-5$ in Table 3.5 were re-created by using purely tumor slices. The same HDF5 files were also re-created one more time with DWI as an additional input channel. In this case the `channels` variable in Table 3.6 was equal to 2. An overview of all of the HDF5 files created during the thesis is given in Appendix B.

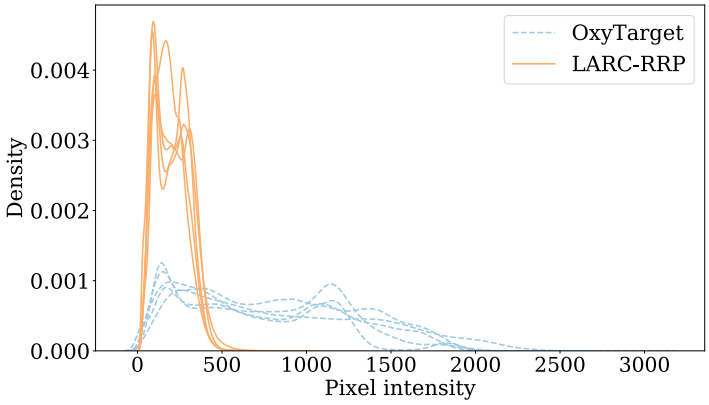
3.4.4 *Standardization of Input Data*

As introduced in Section 2.1.4, one of the main issues with MR images are the non-uniform intensities. The input images were therefore standardized in three different ways, namely z-score normalization (Z-Score), matching of

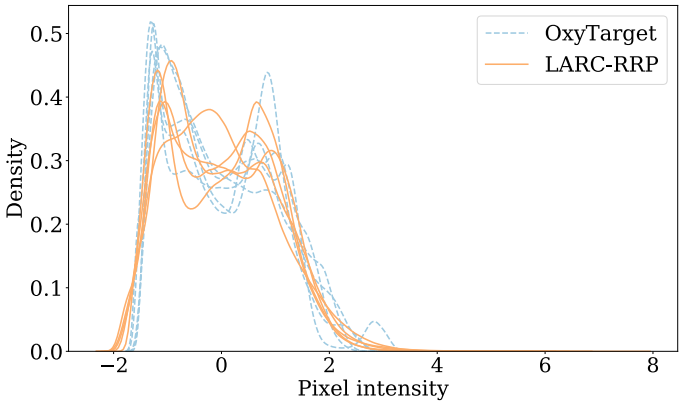
histograms (MH) and a combination of z-score normalization and matching of histograms (MH + Z-Score). The Z-Score method used equation (28) to transform the pixel intensities to the same range of values. This was done on a per patient basis. Hence, the mean and standard deviation for all of the image slices for a given patient were used to transform the pixel values for the same patient. Another possible way to perform the Z-Score method would be to calculate the mean and standard deviation for the entire dataset, and use these variables to transform the pixel intensities. However, in this case the mean and standard deviation could be very high compared to other pixel values in the images. Hence, the transformation could result in very small pixel values which would be difficult to distinguish. In the opposite case, one could calculate the mean and standard deviation for each image slice and consequently perform the Z-Score method on a per image basis. However, this could result in a pixel distribution consisting of a larger range of pixel values and the Z-Score method might not be as effective. Consequently, it was decided that applying the Z-Score on a per patient basis would be the best approach. Figure 3.6a shows an example of how the distribution of pixel intensities looked for five patients in the LARC-RRP dataset and five patients in the OxyTarget dataset before normalizing the images. Figure 3.6b on the other hand shows the distribution of pixel intensities for the same patients as in Figure 3.6a, after applying the Z-Score method.

In the case of matching of histograms (MH), the pixel histograms of the patients in a given dataset were matched in order to deal with the windowing problem of MR images. Figure 3.7 shows an example of how the contrast and brightness in the images changes when matching the pixel histograms. The MH method was performed on a per patient basis. Thus, the distribution of pixel intensities for all images of a given patient was used as reference when matching the histograms. The patient selected as reference was chosen based on image quality. Hence, a patient without any artifacts was selected as reference patient. Figure 3.7b shows one of the image slices from the patient used as reference histogram for the OxyTarget dataset, while the image in Figure 3.7a shows one of the original input images from another patient. The resulting image after applying MH is shown in Figure 3.7c, where the contrast and brightness have been adjusted according to the reference images. The distribution of pixel intensities for five patients in the LARC-RRP dataset and five patients in the OxyTarget dataset after combining the Z-Score method and MH method is presented in Figure 3.6c. When combining the two methods the MH method was applied firstly, then the Z-Score method was applied.

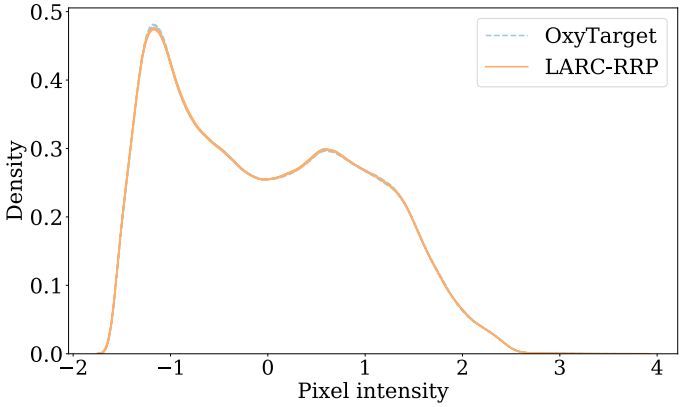
It should be noted that all of the input images were normalized to some degree when entering the DL model. As described in Section 2.3.3 the training images entered the model in batches. The input images in each batch were either automatically normalized based on the minimum and maximum intensity value in the given batch, or by deciding a minimum and maximum intensity value manually.



(a) Original



(b) Z-Score



(c) MH + Z-Score

Figure 3.6: An example of how the pixel intensities for five patients in the LARC-RRP dataset (orange) and five patients in the OxyTarget dataset (blue) looked before any normalization (a), after performing z-score normalization (b) and after combining z-score normalization with matching of pixel histograms (c).

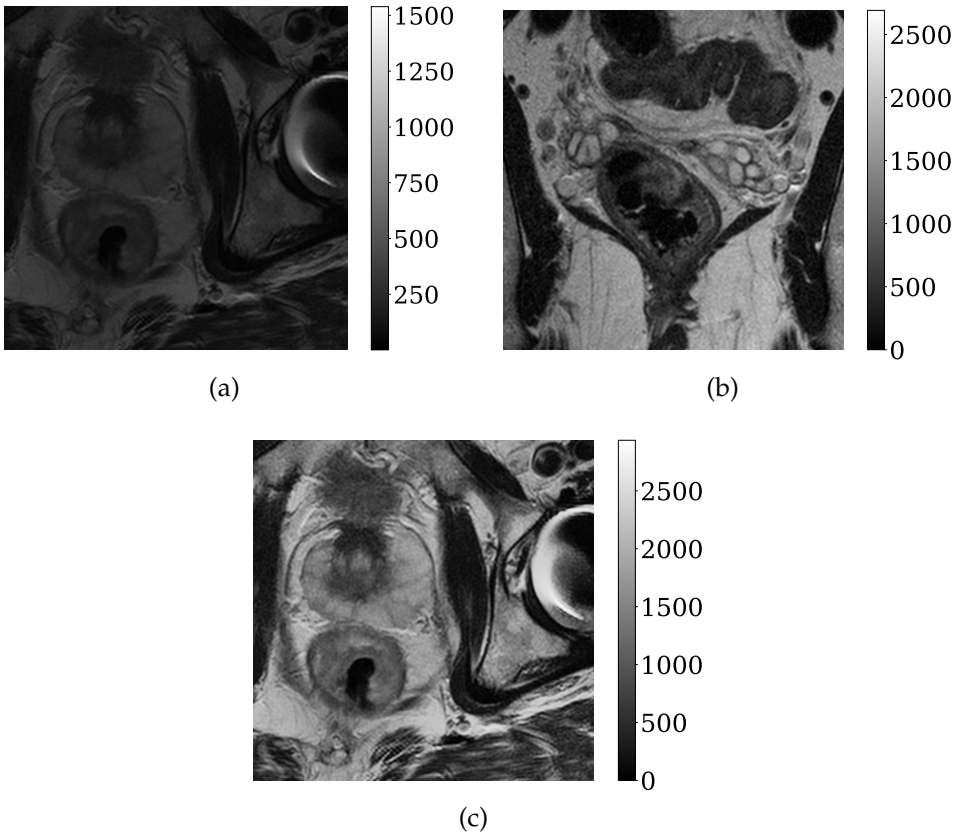


Figure 3.7: An example of how the contrast and brightness in an image changes when matching the pixel histograms. Image (a) shows one of the original image slices from a patient in the OxyTarget dataset, while image (b) shows one of the image slices from the patient used as reference in the same dataset. Hence, the pixel histogram of the images from the patient in (a) was matched to the pixel histogram of the images of the reference patient in (b). Image (c) shows how the original image looked after matching the pixel histograms. The color bar indicates the image intensities.

For the HDF5 files which consisted of input images that were not standardized in any way before entering the model, the automatic normalization option was used. For the HDF5 with standardized input images the minimum pixel intensity was set to -2, while the maximum pixel intensity was set to 2. This was decided based on the histograms in Figure 3.6, and was implemented to ensure that all image intensities were within the same range.

3.5 DEEP LEARNING MODEL

A standard U-Net was used as architecture for the neural network. The network was created by using the *deoxys* framework, which is available from the GitHub repository `https://github.com/huynhngoc/deoxys`. See Section 3.6 for further information on the framework. Figure 2.19 gives an illustration of how the architecture looks, while Table 3.7 gives an overview of the different layers in the network. All the convolutional layers had a kernel size of 3×3,

while all the max pooling layers had a size of 2×2. The ReLU function given in equation (19) was used as activation function in all layers.

Table 3.7: Overview of the U-Net architecture used in the thesis. All convolutional layers had a kernel size of 3×3, while all max pooling layers had a size of 2×2.

Layer	Type	Input	No. output channels
Conv 1	Convolutional	Input image	64
Conv 2	Convolutional	Conv 1	64
MaxPool 1	Max Pooling	Conv 2	64
Conv 3	Convolutional	MaxPool 1	128
Conv 4	Convolutional	Conv 3	128
MaxPool 2	Max Pooling	Conv 4	128
Conv 5	Convolutional	MaxPool 2	256
Conv 6	Convolutional	Conv 5	256
MaxPool 3	Max Pooling	Conv 6	256
Conv 7	Convolutional	MaxPool 3	512
Conv 8	Convolutional	Conv 7	512
MaxPool 4	Max Pooling	Conv 8	512
Conv 9	Convolutional	MaxPool 4	1024
Conv 10	Convolutional	Conv 9	1024
Upconv 1	Upconvolutional	Conv 10	512
Conv 11	Convolutional	Upconv 1, Conv 8	512
Conv 12	Convolutional	Conv 11	512
Upconv 2	Upconvolutional	Conv 12	256
Conv 13	Convolutional	Upconv 2, Conv 6	256
Conv 14	Convolutional	Conv 13	256
Upconv 3	Upconvolutional	Conv 14	128
Conv 15	Convolutional	Upconv 3, Conv 4	128
Conv 16	Convolutional	Conv 15	128
Upconv 4	Upconvolutional	Conv 16	64
Conv 17	Convolutional	Upconv 4, Conv 2	64
Conv 18	Convolutional	Conv 17	64
Conv 19	Convolutional	Conv 18	1

3.5.1 *Hyperparameters*

The process of tuning the hyperparameters of a neural network is a time consuming task. Hence, some of the hyperparameters were fixed and untouched

throughout the thesis. The input value of these hyperparameters were chosen based on recommendations from scientists with previous U-Net experience at the Faculty of Science and Technology, at NMBU. The corresponding input values are presented in Table 3.8. One can notice how the table contains a hyperparameter called "Patience", which is related to the *EarlyStopping* callback. The callback determines when the model should stop training by monitoring a given quantity and checking if the quantity improves over a given number of epochs provided by the "Patience" variable. In this thesis the loss on the validation set was used as monitor quantity. Hence, if the loss on the validation set did not improve over the last 30 epochs the model would stop training.

Table 3.8: Hyperparameters that were kept fixed throughout the thesis.

Fixed Hyperparameters	Input
Activation function	ReLU
Optimizer	Adam
Batch size	16
Epochs	200
Patience	30

Still, different learning rates, loss functions, standardization methods and data augmentations were tested for each dataset in order to find the optimal configuration of the model. The tunable hyperparameters with the different input values tested for each dataset are presented in Table 3.9. The Table provides a new loss function called the *Modified Dice*. This is a modified version of the Dice loss (24) and was introduced in Skjelbred's master thesis [67] by the scientists at NMBU. It is defined by removing the square operation in the denominator. Hence, the Modified Dice loss can be expressed as

$$D_M(\boldsymbol{w}) = 1 - \frac{2\sum_i y_i t_i}{\sum_i y_i + \sum_i t_i} \tag{40}$$

where $D_M(\boldsymbol{w})$ denotes that it is a modified version of the Dice loss (24), y_i is the i-th voxel of the predicted volume and t_i is the i-th voxel of the target volume. The value t_i represents whether sample i belongs to the positive class ($t_i = 1$) or the negative class ($t_i = 0$). The value y_i gives the predicted probability that sample i belongs to the positive class.

A detailed explanation of the standardization methods in Table 3.9 is given in Section 3.4.4, while the data augmentation configurations are presented in Section 3.5.2.

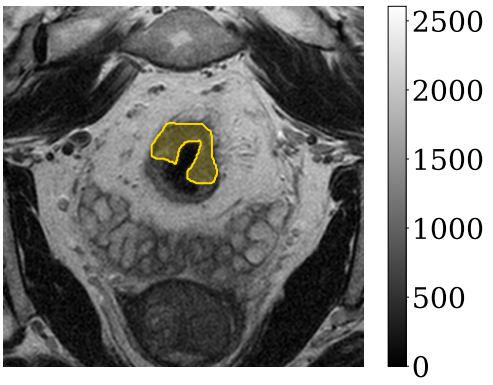
Table 3.9: Hyperparameters which were tuned for each dataset. A set of values were tested for each hyperparameter, as presented in the "Input" column. The standardization methods are explained in Section 3.4.4, while the data augmentation methods are explained in Section 3.5.2.

Tunable Hyperparameters	Input
Learning rate	$[1e-03, 1e-04, 1e-05]$
Loss function	[Dice, Modified Dice]
Standardization	[No, Z-Score, MH, MH + Z-Score]
Data augmentation	[No, Default, BC]

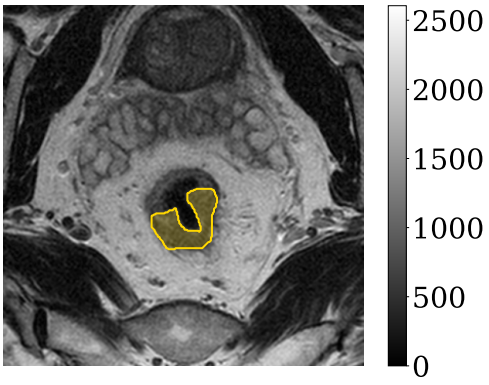
3.5.2 *Data Augmentation*

For the data augmentation two different configurations were tested, as presented in Table 3.9. The Default augmentation was suggested by Ngoc Huynh Bao[2] and consisted of image rotation, zooming, shifting, vertical flipping, as well as changes in brightness, contrast, noise and blur. The second augmentation method is referred to as the Best Combination (BC) and was developed by Maria Ødegaard[3] during her master thesis. This augmentation method was quite similar to the Defualt configuration. However, the BC augmentation did not make any changes in the image brightness or contrast, and the zooming range was larger than in the Default configuration. The exact configuration used for the Default and BC input are presented in Appendix C.1 and C.2, respectively. Documentation of the augmentation preprocessor, *ImageAugmentation2D*, can be found at `https://deoxys.readthedocs.io/en/latest/data.html#module-deoxys.data.preprocessor` [68]. Some examples of possible data augmentations are given in Figure 3.7.

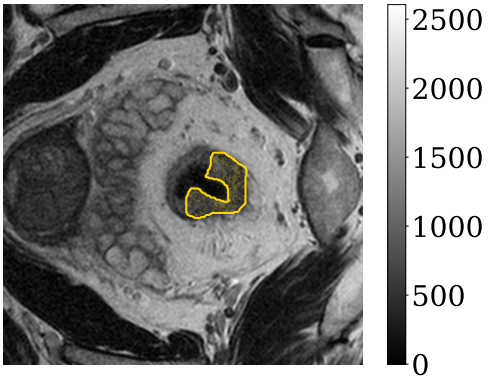
[2]PhD Candidate at the Faculty of Science and Technology, NMBU
[3]Master Student at the Faculty of Science and Technology, NMBU



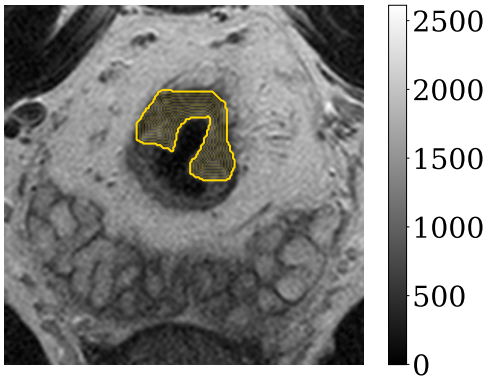
(a) Original image.



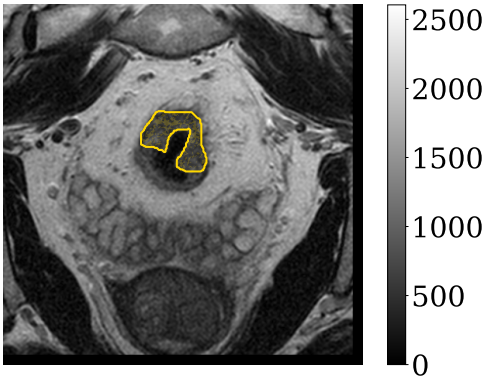
(b) Flipped vertically.



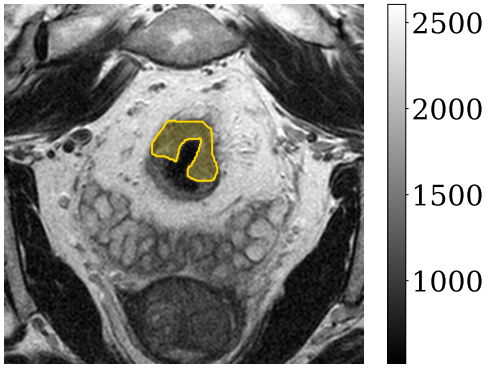
(c) Rotated 90 degrees.



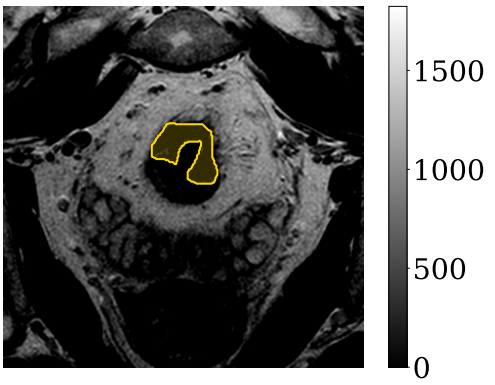
(d) Zoomed in with a factor of 1.5.



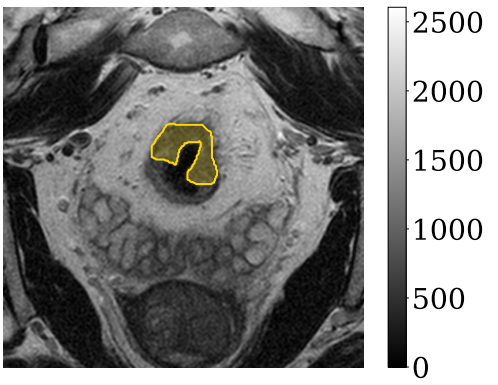
(e) Shifted 10 mm to the left and 10 mm upwards.



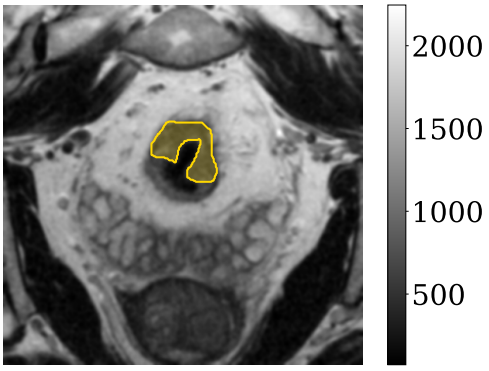
(f) Brightness increased with a factor of 1.2.



(g) Contrast decreased with a factor of 0.7.



(h) Applied Gaussian noise with a variance of 0.05



(i) Blurring with a factor of 1.5

Figure 3.7: Examples of how the original image can be changed by applying different types of data augmentation. The manual delineation made by the radiologist is marked in yellow. The color bar indicates the image intensities.

3.6 CODE AND SOFTWARE

The *deoxys* framework was used for running and creating the models. The framework was developed by Ngoc Huynh Bao[2], and is a Keras-based framework especially developed for automatic segmentation of tumors [68]. A U-Net structure is used, and the code can be accessed from the GitHub repository `https://github.com/huynhngoc/cnn-template`.

The computations were performed on the Orion Compute Cluster, which is hosted and operated by the NMBU IT department [69]. All of the model configurations used in the thesis can be found in the forked GitHub repository `https://github.com/IngvildAskimAdde/cnn-template`.

The post-processing script where the model performance was mapped to each patient in the validation/test set was provided by Ngoc Huynh Bao[2]. The complete code used during the thesis can be accessed from the GitHub repository `https://github.com/IngvildAskimAdde/MasterThesis`.

3.7 ANALYSIS OF THE MODEL

The output from the model was given as a heatmap where each voxel had a score between zero and one. The score of a voxel indicated the probability that the voxel contained tumor or not. Hence, a score close to one would most likely contain tumor while a score close to zero would most likely contain healthy tissue. In order to generate a binary mask a threshold value of 0.5 was applied to the heatmap. Thus, all values above 0.5 was assumed to be tumor, while all values below 0.5 was assumed to be healthy tissue.

The model performance was evaluated by calculating the DSC (34) between the predicted mask and the manually delineated mask, for each patient in the validation set. Hence, the DSC per image slice (DSC_S) gives the overlap between the predicted mask and the manual delineation in a single image slice. The DSC_P provides the mean Dice Similarity Coefficient per image slice (DSC_S) for a patient in the given set. The distribution and median value of the DSC_P of the patients in the validation set were used to evaluate and analyze the model.

3.7.1 *Box Plots*

In some cases box plots can be useful to visualize data. An illustration of the structure of a box plot is presented in Figure 3.8. The black line separating the box gives the median value of the distribution. The lower edge of the box is called the first quartile (Q1) and is given as the median between the smallest number in the dataset and the median of the entire dataset. Hence, the Q1 quartile is often referred to as the 25th percentile, since 25% of the data lays below this line. The upper edge of the box is called the third quartile (Q3) and is given as the median between the highest value in the dataset and

the median of the entire dataset. This quartile is also known as the 75th percentile, since 75% of the data lays below this line. The height of the box is also referred to as the interquartile range (IQR), and 50% of the values are located within this range. The lower and upper whiskers are defined as $Q1 - 1.5 \cdot IQR$ and $Q3 + 1.5 \cdot IQR$, respectively, and represents the values outside of the box. Hence, the exact position of the whiskers depends on the distribution of the dataset. In some cases values can be observed outside the upper or lower whiskers. These values are called outliers and are defined as values which are more than 1.5 times the box height away from the closest edge of the box.

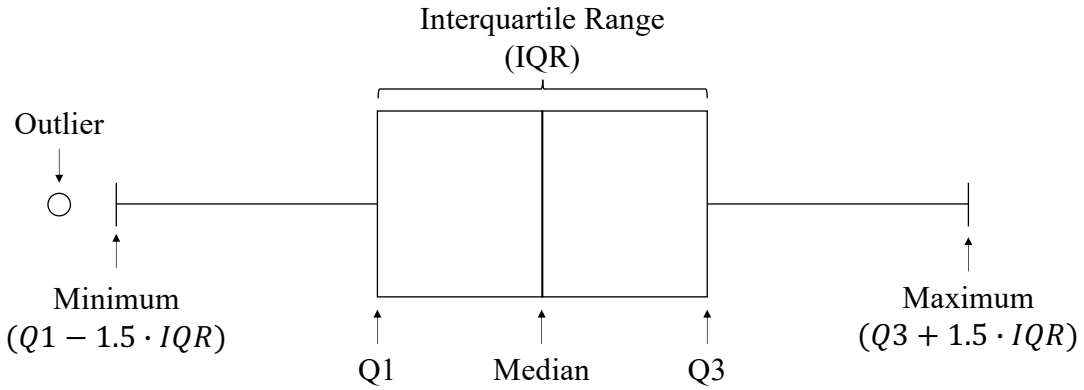


Figure 3.8: Structure of a box plot.

3.7.2 *Violin Plots*

The box plots present a summary of the statistics, such as the median and interquartile ranges, and therefore offer just a limited amount of information about the data. Violin plots, on the other hand, are more informative than box plots. Hence, in most cases, violin plots have been used to evaluate and analyze the models. The violin plots can be considered as a combination of a box plot and a kernel density plot, as illustrated in Figure 3.9. A kernel density plot corresponds to a smooth and continuous alternative of the histogram [70], and is represented by the light blue shaded area in Figure 3.9. Thus, these plots show the full data distribution in addition to the statistical information provided by the box plot. It can be advantageous to show the full data distribution when it has more than one peak, *i.e.*, when it is multimodal [71]. All violin plots in this thesis have been cut at the extreme values of the observed data. In this way the violin plots are more comprehensible, and nothing outside the range of observed data is shown.

3.8 SHALLOW MACHINE LEARNING MODEL

In her project thesis fall 2020 the author evaluated different Shallow Machine Learning (SML) models by testing three classification methods and three unfolding methods on the same datasets as previously introduced in Section 3.3

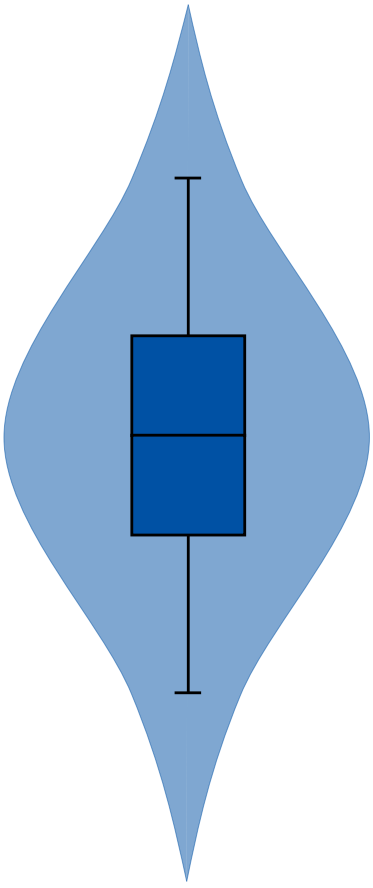
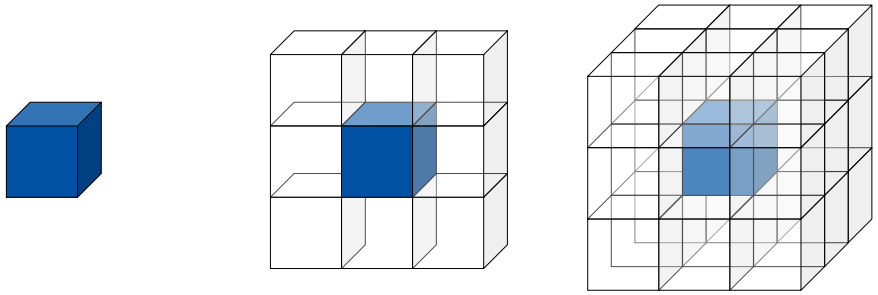


Figure 3.9: Illustration of a violin plot. The light blue shaded area represents the kernel density plot.

[63]. However, in this case the images were cropped by creating a 3D bounding box around the tumor, and adding a 10 mm margin in all directions which was restricted by the Field Of View (FOV). Hence, the input images had varying sizes. The Leave One Out Cross Validation (LOOCV) method was used on a patient level in order to split the datasets into separate training and test subsets, as described in Section 2.3.6.

The properties that a voxel in the MR images represented was determined by using three different unfolding methods which are illustrated in Figure 3.10. First, each voxel was represented by its own intensity value (1D). Second, the intensity information of the closest neighbors in two dimensions was included (2D). In this case eight additional features were included, giving a total of nine features to represent a voxel. The third option was to include the intensity information of the closest neighbors in three dimensions (3D), which gave a total of 27 features representing a voxel. By including the closest neighbors in two or three dimensions the spatial relationship of each voxel was represented. In addition, the intensities were sorted in order to make the model more robust against changes in rotation.

The model was run in MATLAB with three classification techniques, namely Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA)



(a) 1D unfolding: The voxel was represented by its own intensity value.

(b) 2D unfolding: The voxel was represented by including the intensity information of the closest neighbors in two dimensions.

(c) 3D unfolding: The voxel was represented by including the intensity information of the closest neighbors in three dimensions.

Figure 3.10: Illustration of how the properties that a voxel in the MR images represented was determined by using three different unfolding methods; 1D (a), 2D (b) or 3D (c).

and Support Vector Machine (SVM). For LDA and QDA the *fitcdiscr(x,y)* function in MATLAB was used, while for SVM the *fitcecoc(X,Y)* function in MATLAB was used. The computations were performed on resources provided by the NTNU IDUN/EPIC computing cluster [72]. The code used during the project thesis is available from the GitHub repository `https://github.com/IngvildAskimAdde/Prosjektoppgave`.

3.8.1 *Post-Processing*

Post-processing was applied in order to improve the initial predictions made by the SML models. Firstly, a median filter was used in order to smooth the boundaries of the initial predictions. In addition, watershed segmentation was used to separate connected regions into image objects. Watershed segmentation is a region-based technique which uses image morphology to separate objects [73]. This was done by creating a marker ("seed") in each slice of the ground truth images containing tumor. The idea of the marker is to simulate a radiologist clicking into the tumor of the patient, and telling the program if the predicted segment is a tumor or not. Only the predicted regions which contain a marker will therefore be kept, while the other regions are removed. In this way the marker improves the watershed segmentation, which is used to separate the tumor region in the initial predictions from other regions in the images.

3.9 EXPERIMENTAL SETUP

Figure 3.11 shows a flow chart of the experiments conducted during the thesis. The 5-fold cross validation method was run on the OxyTarget data and LARC-RRP data in order to check if the model performance depended on the data splitting. It was decided that it was sufficient to run the 5-fold cross validation on two of the three datasets, since the method is very time consuming. Consequently, the Combined dataset was not included in the 5-fold cross validation step as illustrated in Figure 3.11. Next, different learning rates, loss functions, standardization methods and data augmentations were tested on each dataset with the aim of finding the parameters which gave the overall best model performance. The most optimal model configuration was defined as the model that gave the overall highest median DSC_P on the validation set for each dataset. Thus, the parameters found within the blue box in Figure 3.11 were used as input for all models in later experiments. The result from the validation set of the best model for the OxyTarget data were then compared with the manual delineation of Radiologist_O^2, as an additional way to measure the model performance.

In the following step, the DL models were compared with the SML models presented in Section 3.8, in order to evaluate whether or not the DL models outperformed the SML models. Then, the DL models were run only using image slices that contained tumor as input, again with the same parameters found within the blue box of Figure 3.11 earlier. This was done to investigate whether or not the model performance was affected by the imbalance of tumor slices and non-tumor slices. After this step, it was decided to continue only using tumor slices as input to the models. The test sets were then applied to the models as input data to evaluate the generalization abilities.

As a final step DWIs were included as an additional input for the OxyTarget model, to check if this could increase the model performance. A b-value of $500 \frac{s}{mm^2}$ was used for the DWIs. The b-value was chosen based on litteratur suggesting that a high b-value is advantageous for DL segmentation [74, 75].

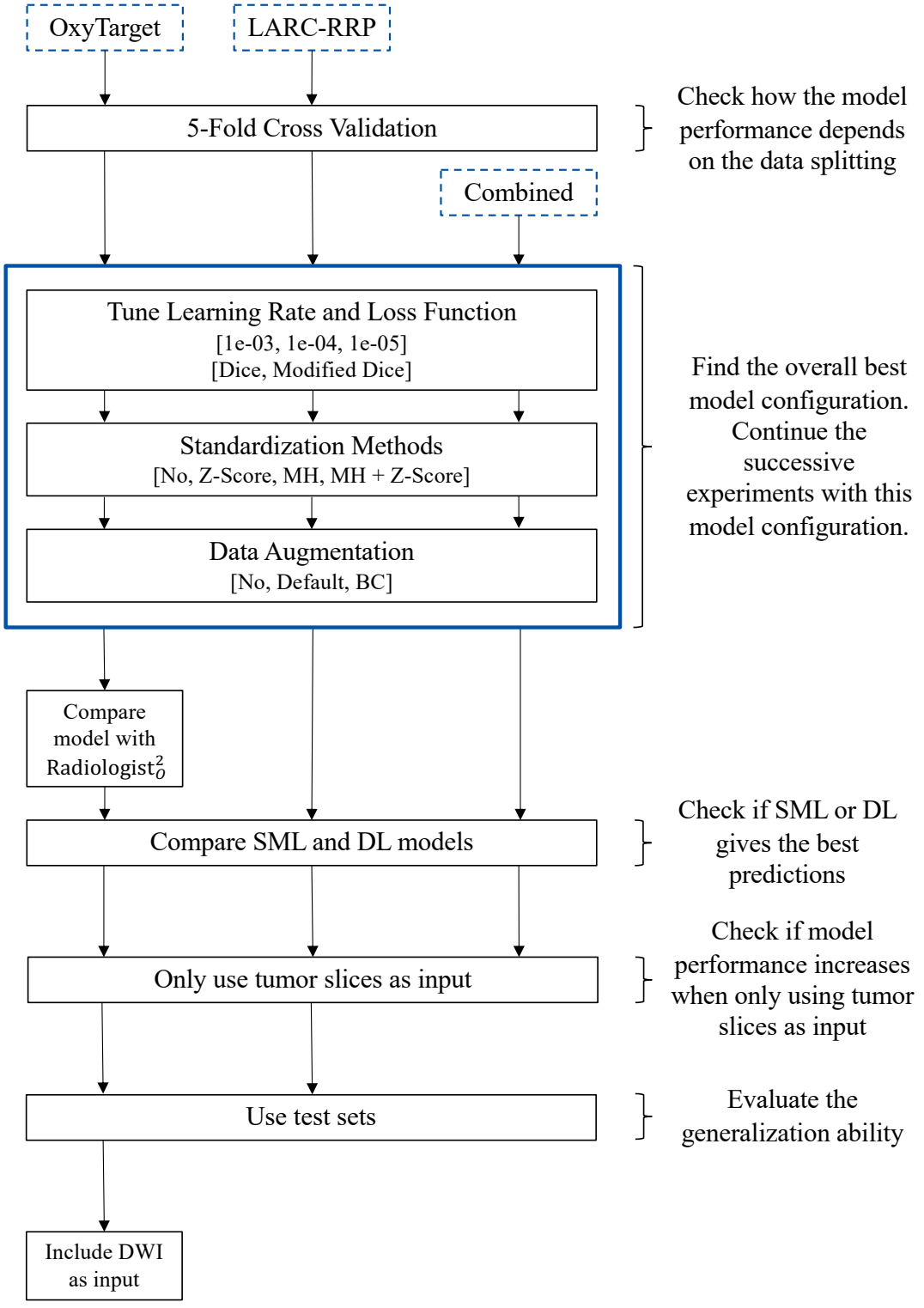


Figure 3.11: Flow chart with an overview of the experiments that were run during the thesis for each dataset.

4 RESULTS

In the following Chapter the results from the experiments are presented. The Chapter follows the same structure as presented in Figure 3.11. Hence, the first section investigates how the model performance was influenced by the splitting of the datasets. This was done by investigating the model performance for the OxyTarget dataset and LARC-RRP dataset when the 5-fold cross validation was used as splitting method.

Next, in Section 4.2, the model configuration was optimized for each dataset. Section 4.2.1 investigates different learning rates and loss functions, Section 4.2.2 explores how the different standardization methods of the datasets affects the model performance, while Section 4.2.3 investigates the effect of data augmentation for each dataset. The overall best combination of these parameters were then used as input for the subsequent experiments.

Section 4.3 compares the OxyTarget model with the delineations made by Radiologist^2_O, as an additional measure of the model performance. Thereafter, the DL models are compared with the SML models in Section 4.4. In Section 4.5 the models were run only using image slices containing tumor as input.

The generalization abilities of the models were evaluated by using the test sets as input in Section 4.6. Lastly, DWIs were included as an additional input for the OxyTarget dataset in Section 4.7 to investigate whether or not another MR sequence could increase the model performance.

4.1 5-FOLD CROSS VALIDATION

The 5-fold cross validations were run with a learning rate of $1e-04$, and the Dice loss (24) was used as loss function. During training the mean DSC_S was calculated on the validation set for each epoch. The maximum value of the mean DSC_S was later used as decision ground for choosing the best model obtained during training. Hence, the value gives an indication of the variation of the best models obtained during training when splitting the datasets differently. In addition, the median DSC_P of the fold used as validation set was used to evaluate the model performance. The maximum value of the mean DSC_S and the median DSC_P for each fold used as validation for the OxyTarget dataset and LARC-RRP dataset are presented in Table 4.1 and 4.2, respectively. Figure 4.1 presents violin plots of the mean DSC_S across the epochs in the training period, calculated on the fold used as validation set. Figure 4.1a shows the results for the OxyTarget dataset, while Figure 4.1b shows the results for the LARC-RRP dataset. In Figure 4.2a and 4.2b violin plots of the DSC_P are shown for the OxyTarget dataset and LARC-RRP dataset, respectively.

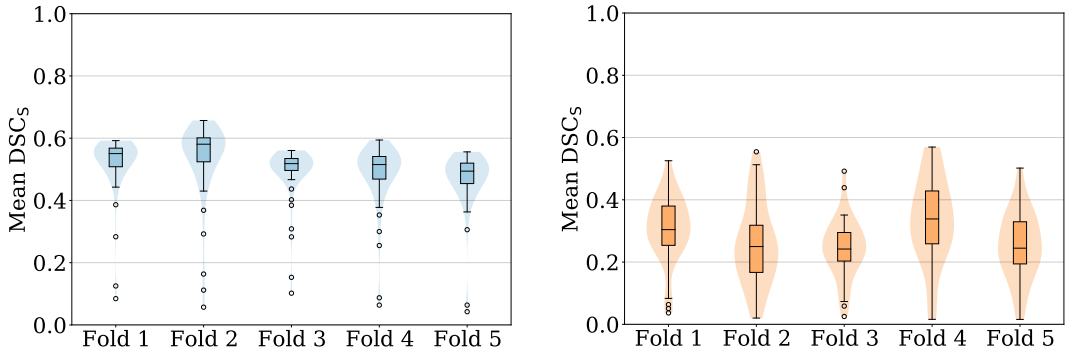
The figures and tables show a variability in the model performance depending on how the datasets were split. However, due to the extensive time demand of finding an optimal solution to the data splitting, as later discussed in section 5.1, it was decided to stay with the traditional split method as initially planned.

Table 4.1: Maximum value of the mean DSC_S and median DSC_P obtained on the fold used as validation for the OxyTarget dataset. A learning rate of $1e-04$ and the Dice loss function (24) were used as input.

Validation Fold	Number of Image Slices	Maximum of Mean DSC_S on Validation Set	Median DSC_P on Validation Set
Fold 1	485	0.592	0.718
Fold 2	507	0.657	0.750
Fold 3	477	0.561	0.673
Fold 4	456	0.594	0.618
Fold 5	445	0.556	0.639

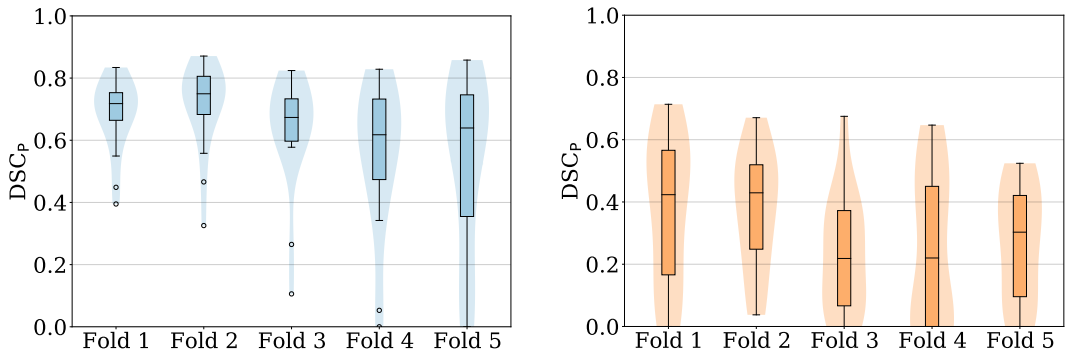
Table 4.2: Maximum value of the mean DSC_S and median DSC_P obtained on the fold used as validation for the LARC-RRP dataset. A learning rate of $1e-04$ and the Dice loss function (24) were used as input.

Validation Fold	Number of Image Slices	Maximum of Mean DSC_S on Validation Set	Median DSC_P on Validation Set
Fold 1	503	0.525	0.423
Fold 2	559	0.554	0.429
Fold 3	548	0.492	0.219
Fold 4	499	0.569	0.220
Fold 5	576	0.502	0.303



(a) OxyTarget (b) LARC-RRP

Figure 4.1: Violin plots of the mean DSC_S when the datasets were validated with the 5-fold cross validation method. The folds on the x-axis represent the fold used as validation set. The mean DSC_S score of the validation set was calculated for each epoch in the training period. A learning rate of $1e-04$ and the Dice loss function (24) were used as input parameters.



(a) OxyTarget (b) LARC-RRP

Figure 4.2: Violin plots of the DSC_P when the datasets were validated with the 5-fold cross validation method. The folds on the x-axis represent the fold used as validation set. A learning rate of $1e-04$ and the Dice loss function (24) were used as input parameters.

4.2 MODEL TUNING

As a next step of the thesis each dataset was tested with different learning rates, loss functions, standardization methods and data augmentation methods to find the configuration which gave the overall highest median DSC_P on the validation set. The results are presented in the following subsections. An overview of the final parameters which gave the overall best model configuration is given in Table 4.6. These parameters were used as input for later experiments, unless something else is stated.

4.2.1 *Learning Rates and Loss Functions*

Each dataset was run with different learning rates and loss functions in order to find the model which gives the highest median DSC_P on the validation set. Table 4.3 presents the median DSC_P while Figure 4.3 presents violin plots of the DSC_P, calculated on the validation sets when combining different learning rates and loss functions. Figure 4.4 illustrates how the median DSC_P varies when changing the learning rate and keeping the loss function constant. In a similar manner, Figure 4.5 shows how the median DSC_P differs when the learning rate is held constant and the loss functions are changed. Figure 4.6 provides image examples of how the prediction changes for the same image slice when changing the learning rate and loss function used as input parameters.

Figure 4.4 shows that a learning rate of $1e-03$ gives a higher median DSC_P on the LARC-RRP dataset and Combined dataset, than with a learning rate of $1e-04$. However, in later experiments a learning rate of $1e-03$ proved to be unstable when combined with different standardization methods and augmentation methods. Thus, it was decided to use a learning rate of $1e-04$ for all datasets in later experiments.

The Modified Dice loss function gave a higher median DSC_P for the LARC-RRP dataset and Combined dataset than with the Dice loss function, as illustrated in Figure 4.3 and 4.5. The OxyTarget dataset obtained a slightly better median DSC_P when the Dice loss was used as loss function. However, the difference in median DSC_P achieved with the Dice loss and Modified Dice loss was small for the OxyTarget dataset. Hence, it was decided to use the Modified Dice loss function for the following experiments, since this gave the overall best performance for all datasets. Consequently, a learning rate of $1e-04$ and the Modified Dice loss function were used as input parameters for the subsequent experiments.

Table 4.3: Overview of median DSC_P achieved on the validation sets when combining different learning rates and loss functions for each dataset.

Dataset	Loss Function	Learning Rate	Median DSC_P on Validation Set
OxyTarget	Dice	$1e-03$	0.732
OxyTarget	Dice	$1e-04$	0.784
OxyTarget	Dice	$1e-05$	0.661
OxyTarget	Modified Dice	$1e-03$	0.684
OxyTarget	Modified Dice	$1e-04$	0.746
OxyTarget	Modified Dice	$1e-05$	0.665
LARC-RRP	Dice	$1e-03$	0.331
LARC-RRP	Dice	$1e-04$	0.283
LARC-RRP	Dice	$1e-05$	0.048
LARC-RRP	Modified Dice	$1e-03$	0.000
LARC-RRP	Modified Dice	$1e-04$	0.303
LARC-RRP	Modified Dice	$1e-05$	0.125
Combined	Dice	$1e-03$	0.489
Combined	Dice	$1e-04$	0.462
Combined	Dice	$1e-05$	0.460
Combined	Modified Dice	$1e-03$	0.334
Combined	Modified Dice	$1e-04$	0.541
Combined	Modified Dice	$1e-05$	0.467

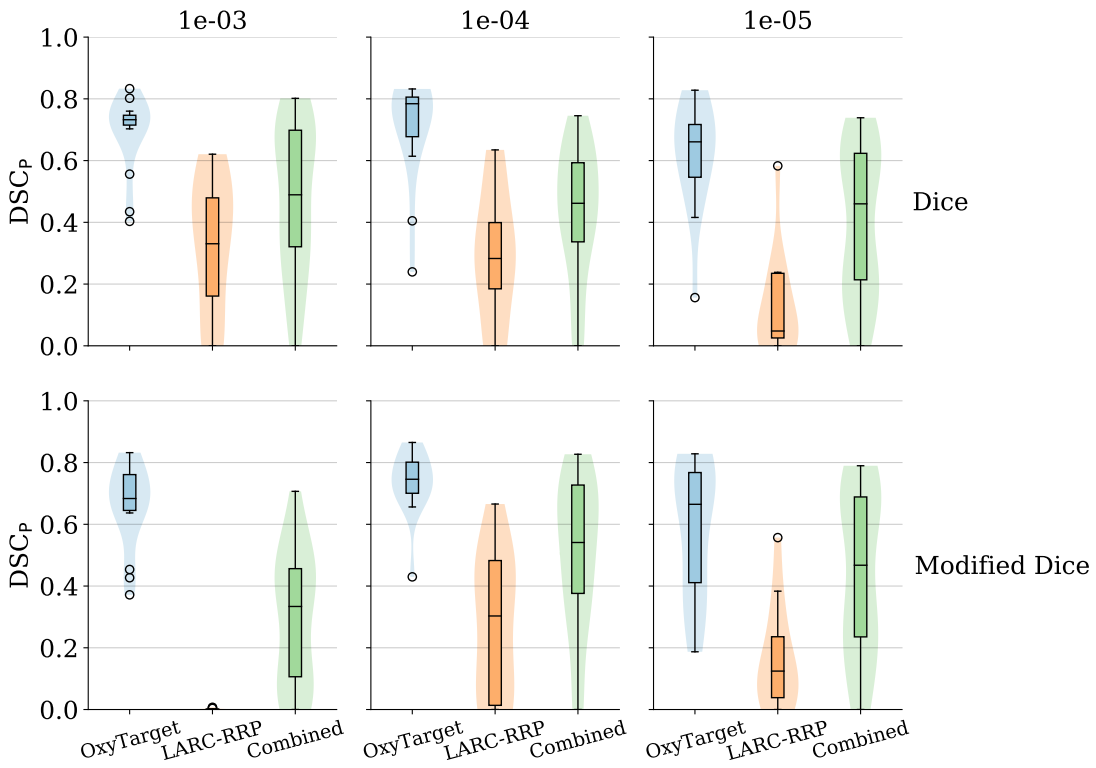


Figure 4.3: Violin plots of DSC$_P$ calculated for the validation set in the OxyTarget dataset (blue), LARC-RRP dataset (orange) and Combined dataset (green). The columns specify the learning rate, while the rows specify the loss function, used as input parameters.

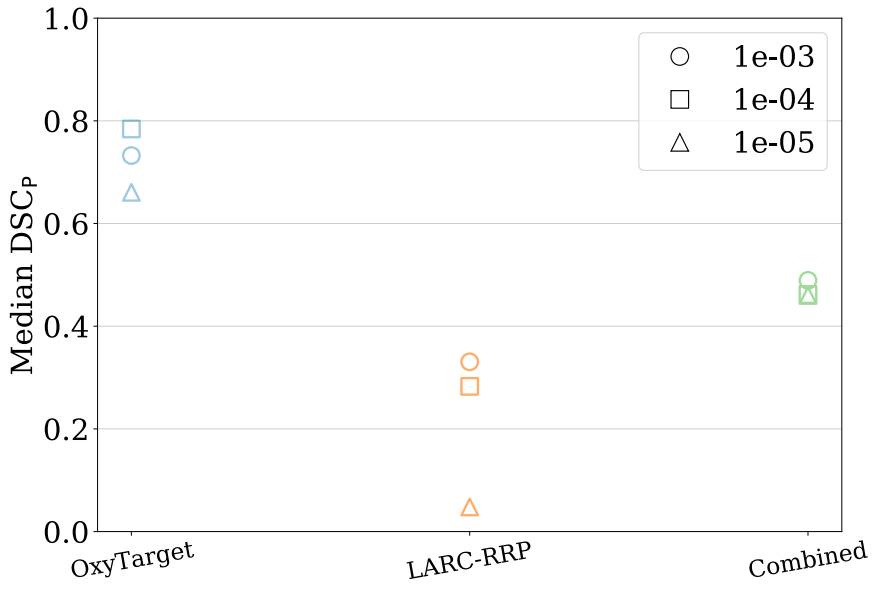


Figure 4.4: Median DSC$_P$ calculated for the validation set of the OxyTarget data (blue), LARC-RRP data (orange) and Combined data (green) when changing the learning rate. The Dice loss (24) was used as loss function, and the different learning rates are presented with different marker shapes.

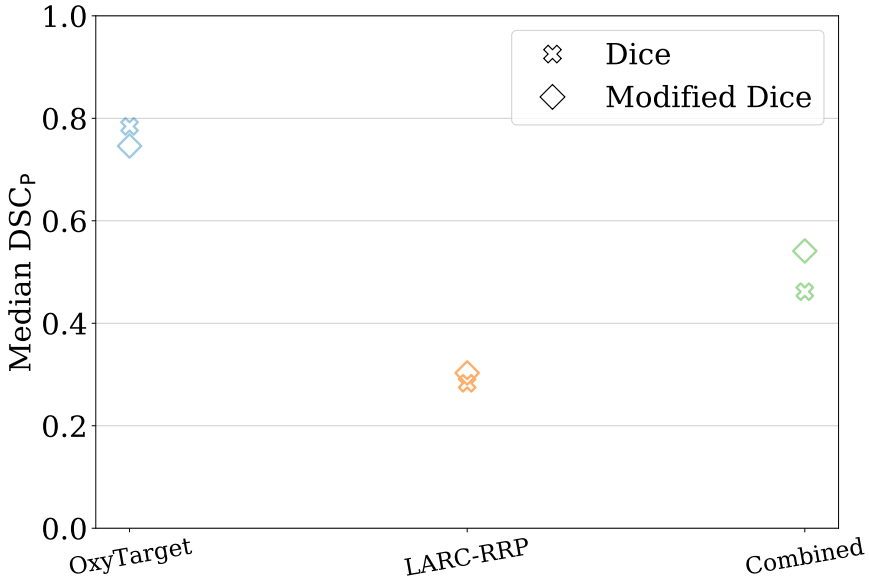
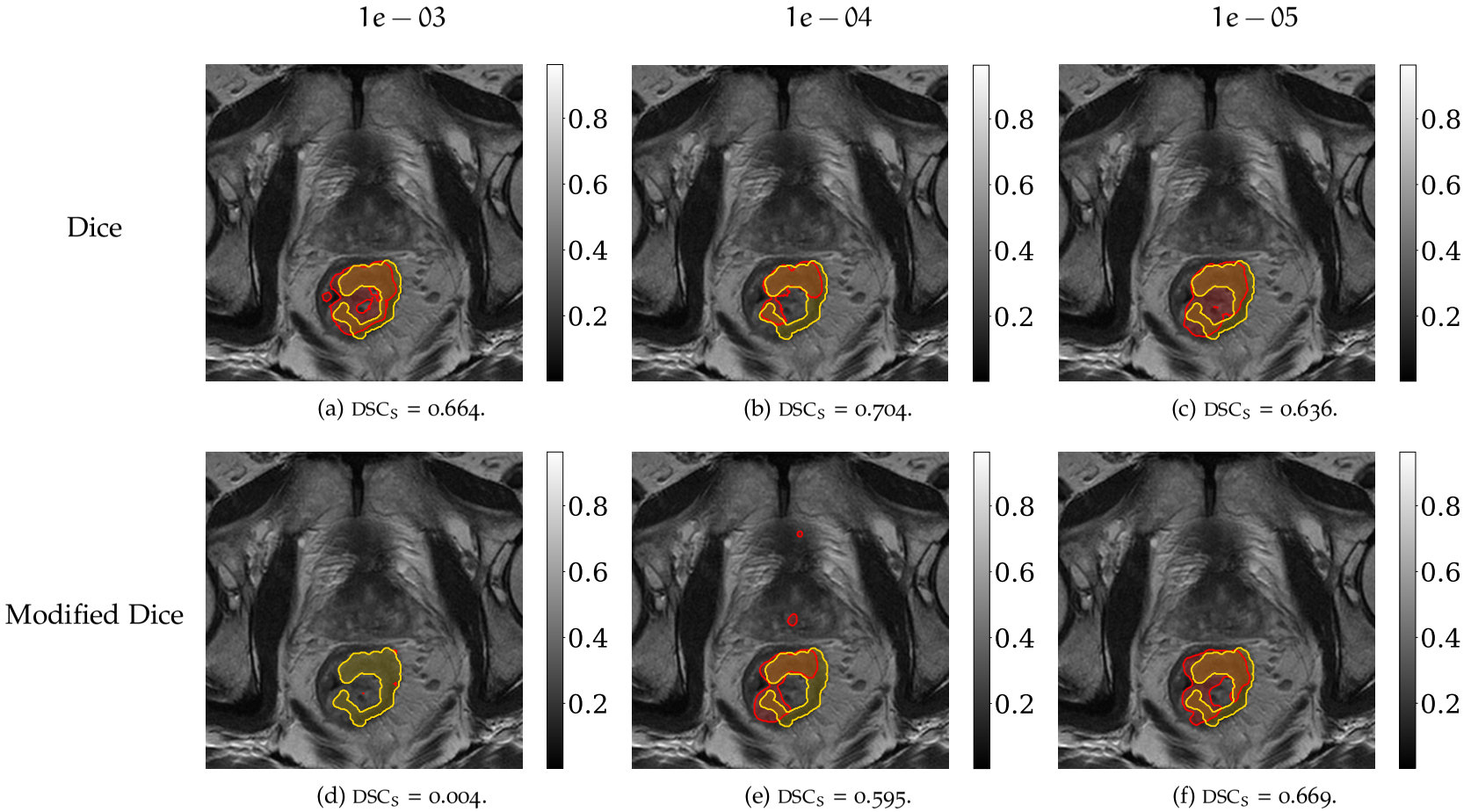


Figure 4.5: Median DSC_P calculated for the validation set of the OxyTarget data (blue), LARC-RRP data (orange) and Combined data (green) when changing the loss function. The learning rate was equal to $1e-04$, and the different loss functions are presented with different marker shapes.



(a) DSC_S = 0.664. (b) DSC_S = 0.704. (c) DSC_S = 0.636.

(d) DSC_S = 0.004. (e) DSC_S = 0.595. (f) DSC_S = 0.669.

Figure 4.6: Image examples of how the prediction (red) changes for the same image slice from a LARC-RRP patient when changing the learning rate and loss function used as input parameters. The manual delineation made by the radiologist is shown in yellow. The columns specify the learning rate used as input, while the rows specify the loss function used as input. The color bar indicates the image intensities.

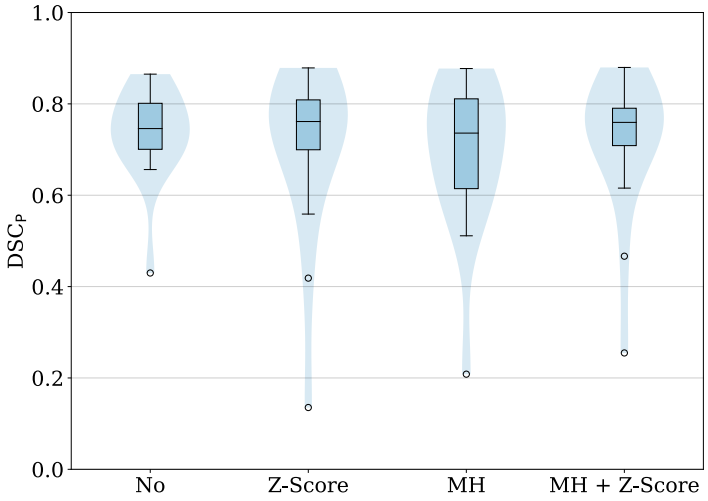
4.2.2 *Standardization of Input Images*

Thereafter, the input images of the three datasets were normalized in four different ways, in order to evaluate the influence of standardized images. The images were standardized as described in Section 3.4.4, and the median DSC_P scores on the validation sets for the various standardization methods are presented in Table 4.4. Figure 4.7 presents violin plots of the DSC_P achieved on the validation sets. The OxyTarget dataset is presented in Figure 4.7a, the LARC-RRP dataset is presented in Figure 4.7b, while the Combined dataset is presented in Figure 4.7c.

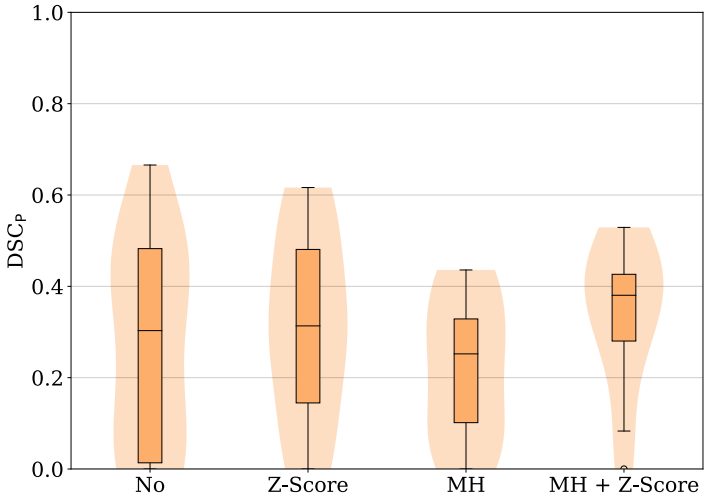
Table 4.4 shows an overall increase in median DSC_P when the input images were standardized according to the "MH + Z-Score" method. Thus, in the following experiments these standardized images have been used as input.

Table 4.4: Median DSC_P achieved on the validation sets when different standardization methods were applied to the input images; no normalization (No), z-score normalization (Z-Score), matching of histograms (MH) and with a combination of z-score normalization and matching of histograms (MH + Z-Score). The parentheses presents the contribution from the OxyTarget/LARC-RRP dataset in the Combined dataset.

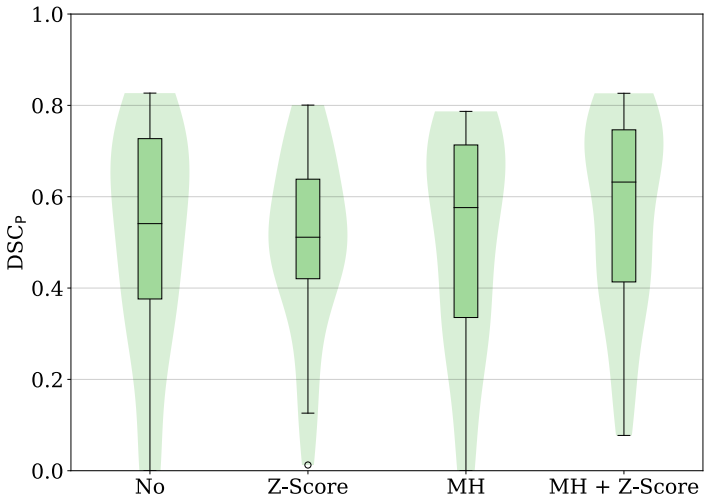
Dataset	Standardization	Median DSC_P on Validation Set
OxyTarget	No	0.746
OxyTarget	Z-Score	0.761
OxyTarget	MH	0.736
OxyTarget	MH + Z-Score	0.760
LARC-RRP	No	0.303
LARC-RRP	Z-Score	0.313
LARC-RRP	MH	0.252
LARC-RRP	MH + Z-Score	0.380
Combined	No	0.541 (0.700/0.376)
Combined	Z-Score	0.511 (0.566/0.507)
Combined	MH	0.576 (0.710/0.371)
Combined	MH + Z-Score	0.632 (0.726/0.478)



(a) OxyTarget dataset.



(b) LARC-RRP dataset.



(c) Combined dataset.

Figure 4.7: Violin plots of DSC_P calculated for the validation set of the OxyTarget dataset (a), LARC-RRP dataset (b) and Combined dataset (c). The datasets were standardized in four different ways before entering the model; no normalization (No), z-score normalization (Z-Score), matching of histograms (MH) and with a combination of matching of histograms and z-score normalization (MH + Z-Score).

4.2.3 *Data Augmentation*

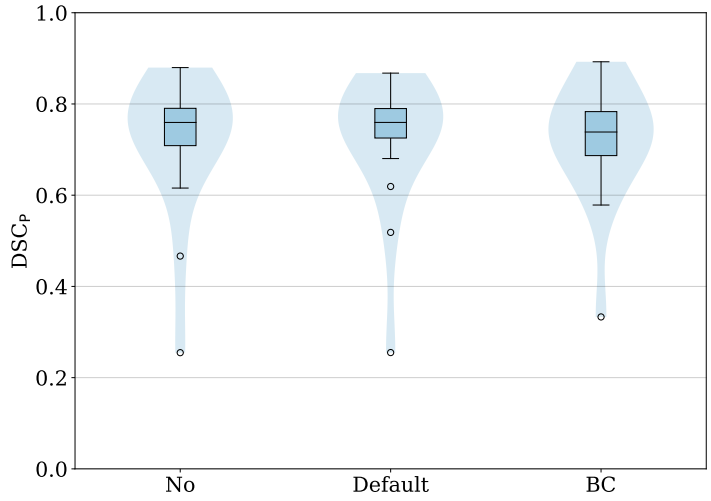
Data augmentation was investigated as a final step in the process of finding the overall best model configuration. Table 4.5 presents the median DSC_P on the validation sets when no augmentation (No), the Default augmentation (Default) and the Best Combination augmentation (BC) were applied to the input images, as described in Section 3.5.2.

Figure 4.8a shows violin plots of the calculated DSC_P on the validation set of the OxyTarget data when the various augmentation methods were applied to the input images. Figure 4.8b and 4.8c show similar violin plots for the LARC-RRP dataset and Combined dataset, respectively.

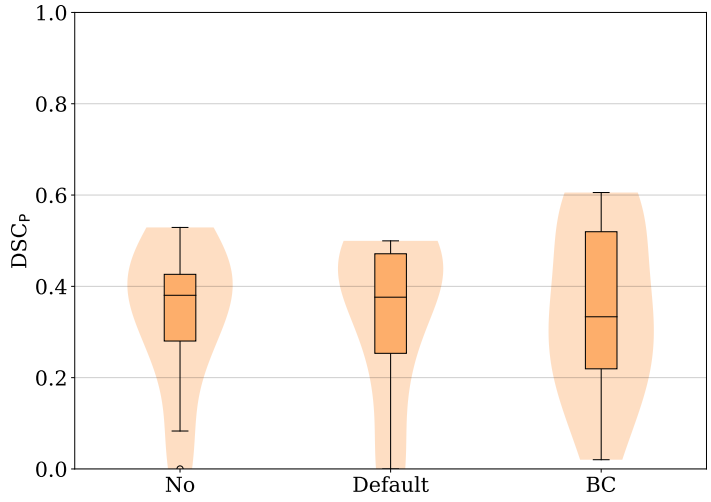
Table 4.5 shows that the median DSC_P decreases in some cases when data augmentation is used. However, as stated in Section 2.3.4.1 the main advantage with applying data augmentation is to increase the generalization ability of the model, and making the model more robust. It was therefore decided to use the BC augmentation method as input for the following experiments, even though this decreased the median DSC_P for the OxyTarget dataset and LARC-RRP dataset as shown in Table 4.5. The BC method was preferred over the Default method due to the data distributions presented in Figure 4.8, where the BC method avoids any predictions with a DSC_P equal to zero.

Table 4.5: Median DSC_P achieved on the validation sets when various data augmentation methods were applied to the input images; no augmentation (No), Default augmentation (Default) and Best Combination augmentation (BC). The exact configuration of the data augmentation methods are provided in Appendix C.1 and C.2. The parentheses presents the contribution from the OxyTarget/LARC-RRP dataset in the Combined dataset.

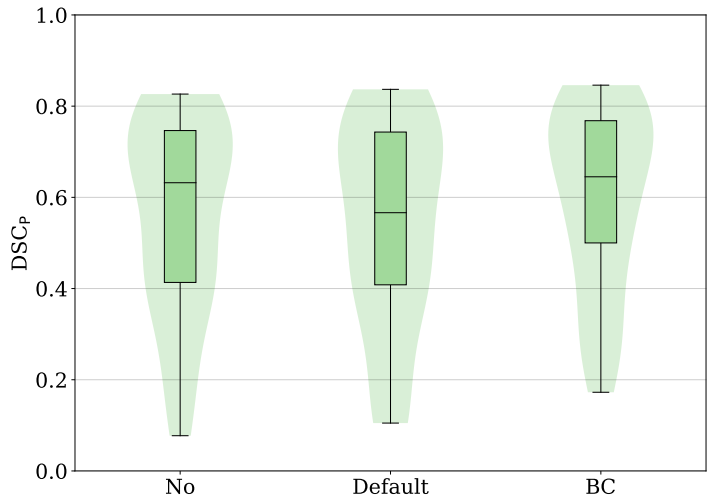
Dataset	Data Augmentation	Median DSC_P on Validation Set
OxyTarget	No	0.760
OxyTarget	Default	0.760
OxyTarget	BC	0.738
LARC-RRP	No	0.380
LARC-RRP	Default	0.376
LARC-RRP	BC	0.333
Combined	No	0.632 (0.726/0.478)
Combined	Default	0.566 (0.743/0.408)
Combined	BC	0.645 (0.756/0.500)



(a) OxyTarget dataset.



(b) LARC-RRP dataset.



(c) Combined dataset.

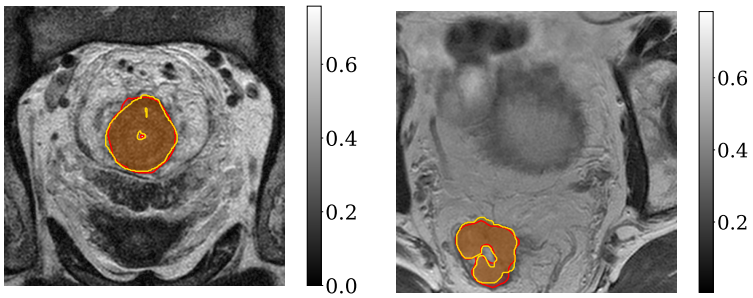
Figure 4.8: Violin plots of DSC_P calculated for the validation set of the OxyTarget dataset (a), LARC-RRP dataset (b) and Combined dataset (c). The input images were augmented in three different ways; no augmentation (No), Default augmentation (Default) and Best Combination augmentation (BC). The exact configuration of the data augmentation methods are provided in Appendix C.1 and C.2.

4.2.4 *Summary of Model Tuning*

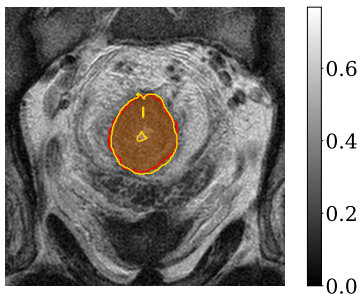
The final parameters used as input in the DL models are given in Table 4.6. If nothing else is stated these are the parameter values used in the following sections. Figure 4.9 shows the image slice with the highest DSC_S in each of the validation sets, when the parameters presented in Table 4.6 were used as input. The predicted mask is given in red, while the mask delineated by the radiologist is given in yellow.

Table 4.6: Summary of model parameters used in the DL models which gave the best overall performance on the validation sets. These input parameters were used in later experiments, unless something else is stated.

Parameter	Input Value
Learning rate	$1e-04$
Loss function	Modified Dice
Standardization	MH + Z-Score
Data Augmentation	BC



(a) DSC_S = 0.946. (b) DSC_S = 0.909.



(c) DSC_S = 0.943.

Figure 4.9: The image slice in the validation set with the highest DSC_S for the Oxy-Target data (a), LARC-RRP data (b) and Combined data (c) when using the parameters given in Table 4.6 as input. The predicted mask is given in red, while the mask delineated by the radiologist is given in yellow. The color bar indicates the image intensities.

4.3 COMPARISON OF OXYTARGET MODEL AND RADIOLOGIST^2_O

In the following section the patients with a second delineation have been investigated in order to analyse how the OxyTarget DL model performs when a new radiologist is used as ground truth. The patients in the validation set were used to compare the performance with the radiologists. However, five patients in the original validation set were not delineated by Radiologist^2_O. Hence, the number of patients in the validation set was reduced to 11 in the following section for the OxyTarget data.

Table 4.7 presents the median DSC_P on the validation set when comparing the predicted mask with the delineations made by the two radiologists. In addition, the delineations made by the radiologists on the validation patients were also compared with each other in order to calculate the interobserver variations. The interobserver variation on all of the patients delineated by both Radiologist^1_O and Radiologist^2_O (76 patients) had a median DSC_P equal to 0.805.

Figure 4.10 presents violin plots of the DSC_P when comparing the different delineations, while Figure 4.11 shows the difference in DSC_P per patient in the validation set when comparing the different delineations.

Table 4.7: Median DSC_P achieved on the validation patients when comparing with two different delineations made by Radiologist^1_O and Radiologist^2_O. The median DSC_P of the interobserver variation is also presented. Note that the model was trained with Radiologist^1_O as ground truth.

First Mask	Second Mask	Median DSC_P on Validation Set
Prediction	Radiologist^1_O	0.741
Prediction	Radiologist^2_O	0.737
Radiologist^1_O	Radiologist^2_O	0.821

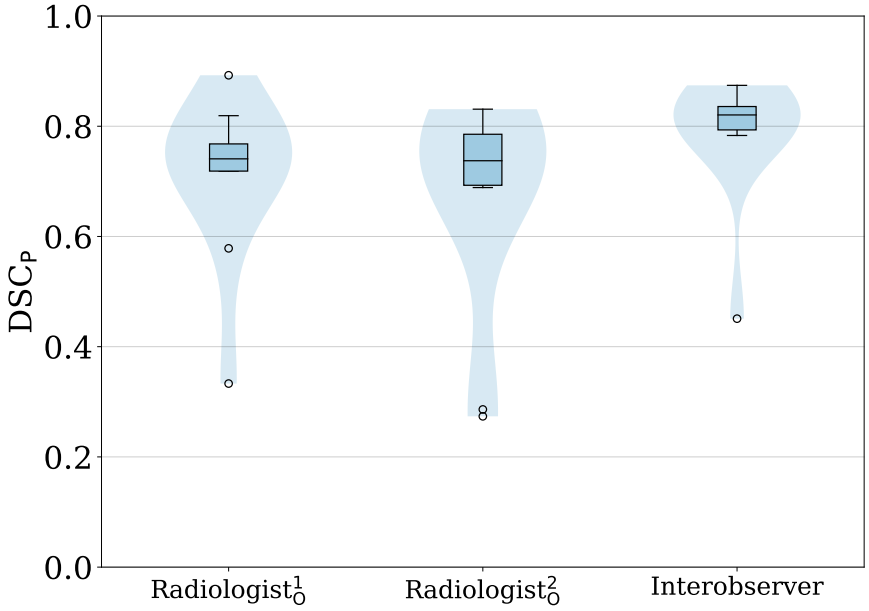


Figure 4.10: DSC$_P$ calculated for two different delineations made by Radiologist^1_O and Radiologist^2_O on the validation set of the OxyTarget data. The "Interobserver" violin plot shows the DSC$_P$ when comparing the two delineations with each other. Note that the model was trained with Radiologist^1_O as ground truth.

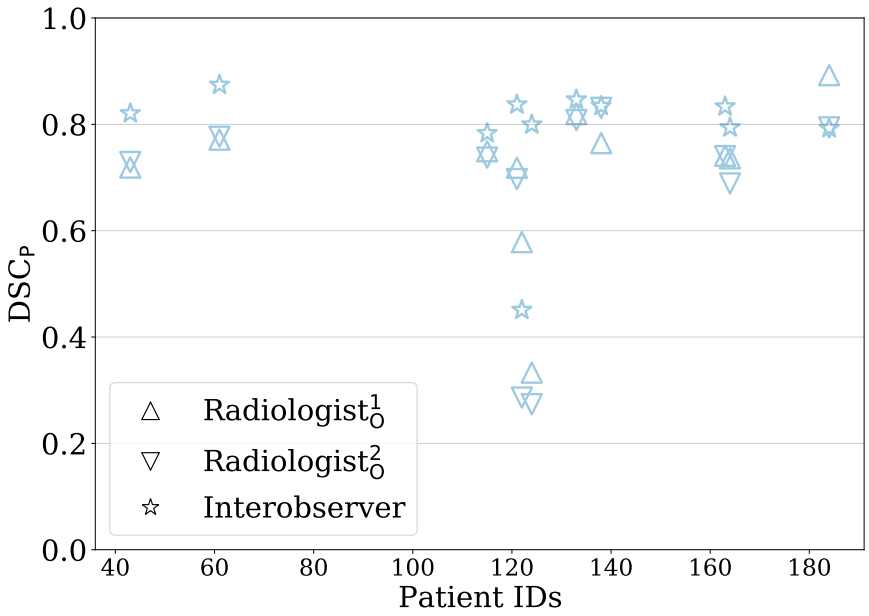


Figure 4.11: Median DSC$_P$ calculated for two different delineations made by Radiologist^1_O and Radiologist^2_O, per patient on the validation set of the OxyTarget data. The "Interobserver" shows the DSC$_P$ when comparing the two delineations with each other.

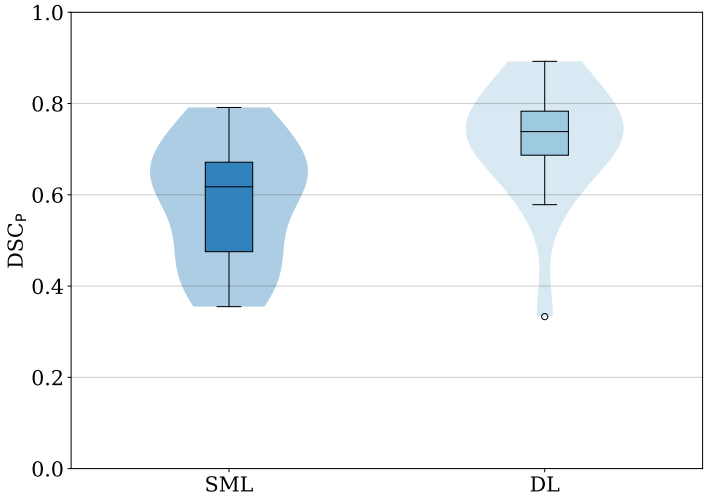
4.4 SHALLOW MACHINE LEARNING VS. DEEP LEARNING

In the following section the DL models are compared with the SML models investigated during the author's project thesis. Table 4.8 presents the median DSC_P for the SML models, when combining different classification methods and unfolding methods. In this case the median DSC_P was calculated based on the LOOCV method, as described in Section 2.3.6. The model which gave the best median DSC_P for a given dataset was used for comparison with the DL models. Hence, for all of the datasets, the model with QDA as classification method and 3D as unfolding method was used for comparison.

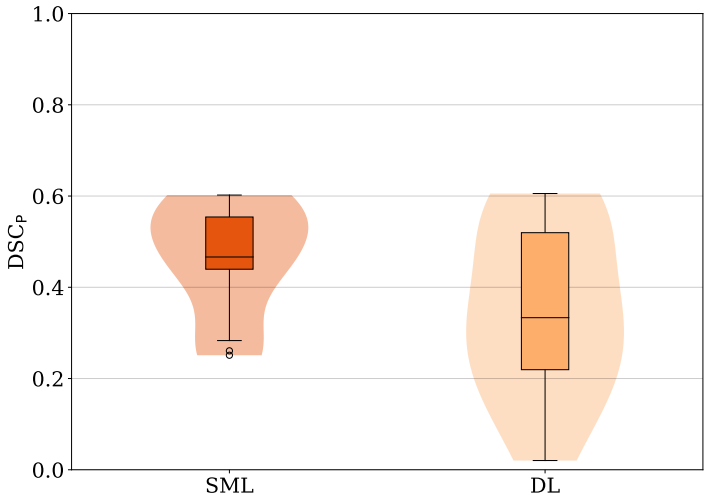
The DSC_P of the patients in the validation sets were used to compare the SML models and DL models. The median DSC_P of these patients are presented and compared in Table 4.9. Figure 4.12 shows violin plots of the distribution of DSC_P for the validation patients from the SML model and DL model for each dataset. Figure 4.12a compares the OxyTarget data, Figure 4.12b compares the LARC-RRP data, while Figure 4.12c compares the Combined data.

Table 4.8: Median DSC_P for the SML models, with different combinations of classification and unfolding methods. The various parameters were run on the LARC-RRP dataset, OxyTarget dataset, and on the Combined dataset. The table was collected from the author's project thesis. Entries marked with * represent analysis not executed.

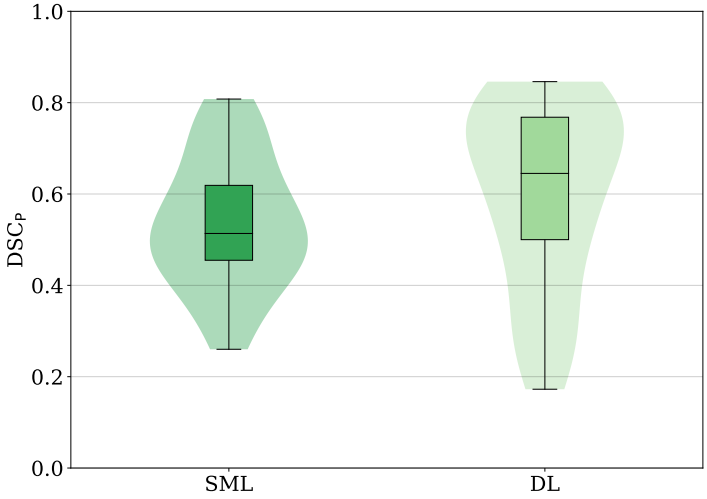
Parameters	Median DSC_P (LARC-RRP)	Median DSC_P (OxyTarget)	Median DSC_P (Combined)
LDA + 1D	0.338	0.499	0.452
LDA + 2D	0.398	0.553	0.482
LDA + 3D	0.446	0.552	0.497
QDA + 1D	0.430	0.558	0.478
QDA + 2D	0.420	0.571	0.503
QDA + 3D	0.452	0.612	0.544
SVM + 1D	0.350	0.512	0.441
SVM + 2D	0.397	0.553	*
SVM + 3D	0.449	0.553	*



(a) OxyTarget dataset.



(b) LARC-RRP dataset.



(c) Combined dataset.

Figure 4.12: Violin plots of the DSC_P calculated on the validation patients of the SML model and DL model. The SML model used QDA as classification method and 3D as unfolding method.

Table 4.9: Median DSC_P achieved on the validation patients for the SML model and DL model. The SML model used QDA as classification method and 3D as unfolding method.

Dataset	Model	Median DSC_P on Validation Set
OxyTarget	SML	0.618
OxyTarget	DL	0.738
LARC-RRP	SML	0.466
LARC-RRP	DL	0.333
Combined	SML	0.514
Combined	DL	0.645

4.5 MODEL PERFORMANCE WHEN ONLY USING TUMOR SLICES

So far, the model has shown poor performance on the LARC-RRP dataset with a median DSC_P of 0.380. When studying the training performance of the LARC-RRP dataset in more detail it is clear that the model struggles to learn anything during training. This is confirmed in Figure 4.13a where the loss stops decreasing after a few epochs, and never reaches a value below 0.7. Accordingly, the DSC also stops improving after a few epochs, indicating that the model stops learning.

Investigation of Table 3.2 gives that 71.2% of the image slices in the OxyTarget dataset contains tumor, while there is only 29.3% of the image slices in the LARC-RRP dataset which contains tumor. Thus, the training set, validation set and test set of the LARC-RRP dataset contains 29.6%, 30.7% and 26.3% tumor slices, respectively. Due to the small percentage of tumor slices in the LARC-RRP dataset it was decided to train the models exclusively with image slices which contained tumor. The proceeding training performance is presented in Figure 4.13b, where the loss decreases and the DSC increases over several epochs. Hence, the model is able to learn during training when only using tumor slices as input.

Figure 4.14 compares the distribution of DSC_P achieved when using all image slices as input and when solely using tumor slices as input, for each dataset. Table 4.10 compares the median DSC_P achieved on the validation sets for the same situation.

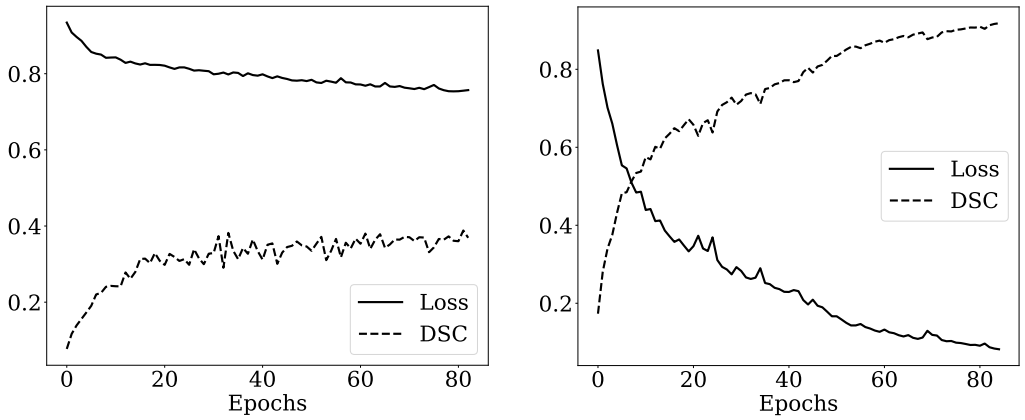
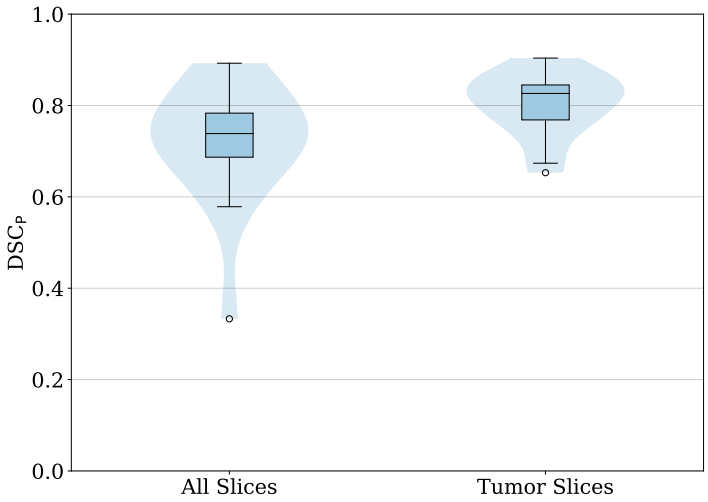
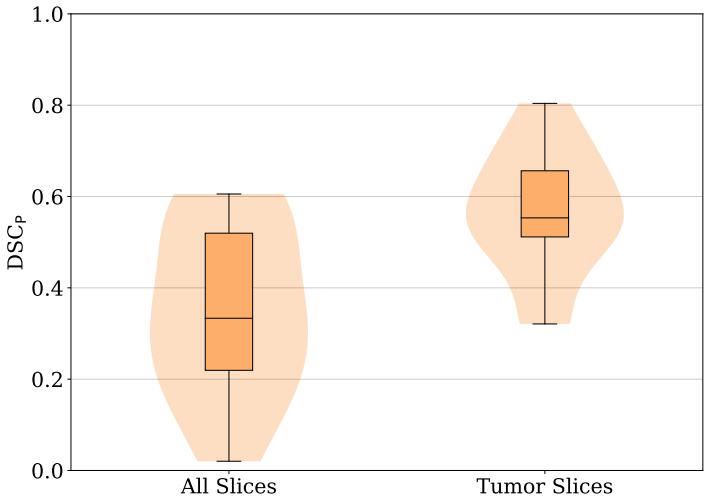


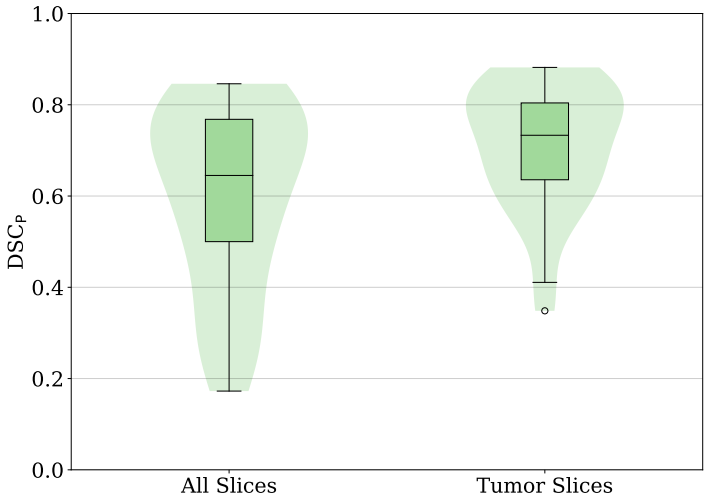
Figure 4.13: Example of how the training performance on the LARC-RRP dataset looked when training the model on all image slices (a) and when training the model exclusively on image slices which contains tumor (b).



(a) OxyTarget dataset.



(b) LARC-RRP dataset.



(c) Combined dataset.

Figure 4.14: Comparison of the DSC$_P$ calculated on the validation patients when all image slices were used as input and when solely using tumor slices as input.

Table 4.10: Comparison of the median DSC_P calculated on the validation patients when all image slices were used as input, and when exclusively using the tumor slices as input. The parentheses presents the contribution from the OxyTarget/LARC-RRP dataset in the Combined dataset.

Dataset	Image slices	Median DSC_P on Validation Set
OxyTarget	All	0.738
OxyTarget	Tumor	0.826
LARC-RRP	All	0.333
LARC-RRP	Tumor	0.553
Combined	All	0.645 (0.756/0.500)
Combined	Tumor	0.733 (0.804/0.636)

4.6 MODEL PERFORMANCE ON TEST SETS

In the following section the models were tested with the test sets created in section 3.4.2, in order to evaluate the generalization abilities. The models were trained using exclusively tumor slices and the input parameters listed in Table 4.6. Table 4.11 shows the median DSC_P achieved on the test set for each dataset, while Table 4.12, 4.13 and 4.14 presents the DSC_P for each patient in the OxyTarget test set, LARC-RRP test set and Combined test-set, respectively. Table 4.12 also includes the interobserver DSC_P between $Radiologist^1_O$ and $Radiologist^2_O$ for those patients where it is applicable. Figure 4.15 shows violin plots of the DSC_P for the test set of each dataset, while Figure 4.16 shows a selection of T2w image slices from the patient in the test sets with the highest DSC_P.

Table 4.11: Median DSC_P achieved on the test patients for each dataset. The parenthesis presents the contribution from the OxyTarget/LARC-RRP dataset in the Combined dataset.

Dataset	Median DSC_P on Test Set
OxyTarget	0.691
LARC-RRP	0.558
Combined	0.673 (0.718/0.597)

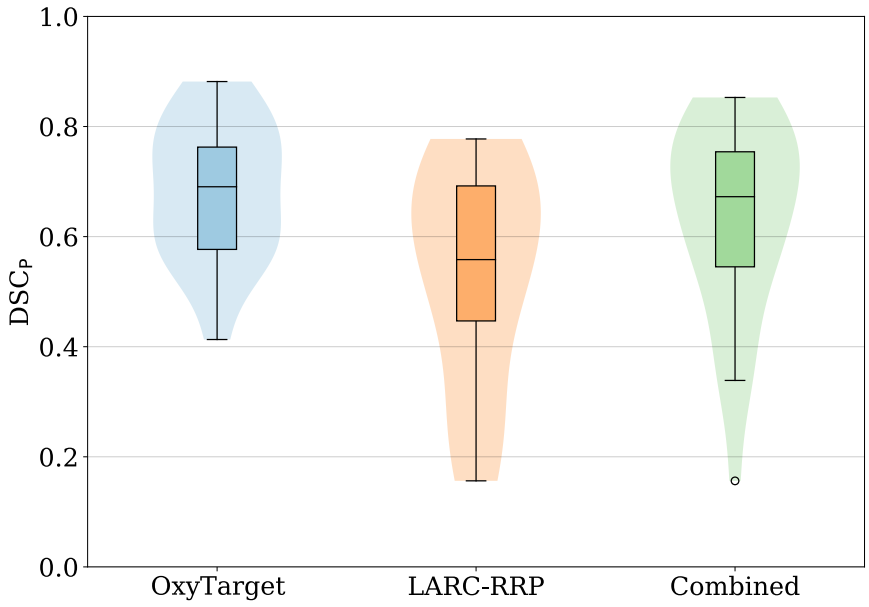


Figure 4.15: Violin plots of the DSC_P for the OxyTarget test set (blue), LARC-RRP test set (orange) and Combined test set (green).

Table 4.12: DSC_P for the OxyTarget test patients. The interobserver DSC_P between $Radiologist^1_O$ and $Radiologist^2_O$ is given for the patients which were delineated by $Radiologist^2_O$ as well. Entries marked with * represent patients where $Radiologist^2_O$ did not delineate the Region Of Interest (ROI).

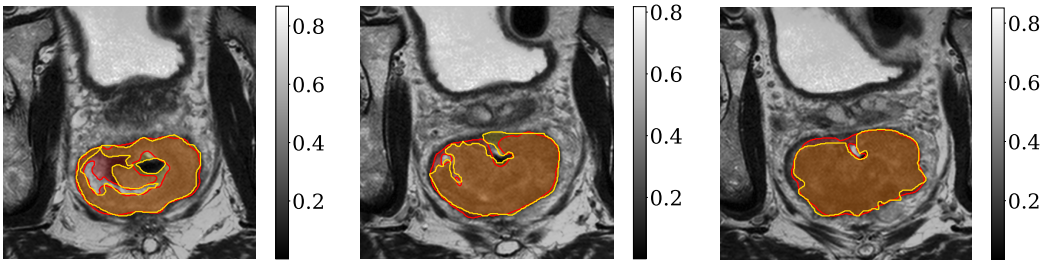
Patient ID	DSC_P	Interobserver DSC_P
OxyTarget 29	0.596	0.730
OxyTarget 41	0.882	0.900
OxyTarget 49	0.572	0.848
OxyTarget 73	0.833	*
OxyTarget 83	0.757	*
OxyTarget 87	0.619	*
OxyTarget 88	0.691	*
OxyTarget 89	0.577	*
OxyTarget 97	0.763	*
OxyTarget 99	0.569	*
OxyTarget 111	0.661	*
OxyTarget 128	0.715	0.700
OxyTarget 131	0.820	0.817
OxyTarget 145	0.818	0.838
OxyTarget 146	0.413	0.724
OxyTarget 157	0.543	0.575
OxyTarget 176	0.754	0.813

Table 4.13: DSC_P for the LARC-RRP test patients.

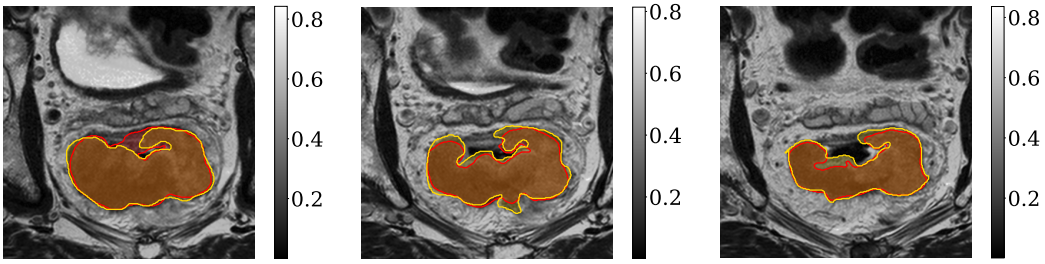
Patient ID	DSC_P
LARC-RRP 3	0.260
LARC-RRP 5	0.492
LARC-RRP 9	0.692
LARC-RRP 15	0.156
LARC-RRP 23	0.705
LARC-RRP 24	0.777
LARC-RRP 29	0.544
LARC-RRP 38	0.558
LARC-RRP 52	0.672
LARC-RRP 79	0.271
LARC-RRP 81	0.730
LARC-RRP 83	0.447
LARC-RRP 99	0.635

Table 4.14: DSC_P for the Combined test patients.

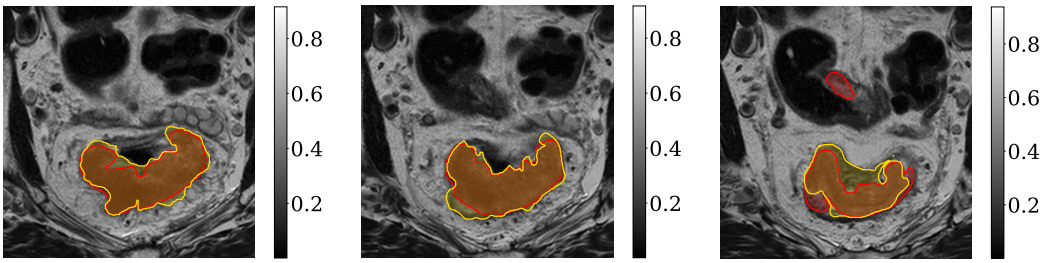
Patient ID	DSC_P
OxyTarget 29	0.717
OxyTarget 41	0.845
OxyTarget 49	0.575
OxyTarget 73	0.821
OxyTarget 83	0.757
OxyTarget 87	0.637
OxyTarget 88	0.718
OxyTarget 89	0.359
OxyTarget 97	0.745
OxyTarget 99	0.655
OxyTarget 111	0.607
OxyTarget 128	0.770
OxyTarget 131	0.853
OxyTarget 145	0.841
OxyTarget 146	0.461
OxyTarget 157	0.463
OxyTarget 176	0.790
LARC-RRP 3	0.339
LARC-RRP 5	0.520
LARC-RRP 9	0.737
LARC-RRP 15	0.156
LARC-RRP 23	0.743
LARC-RRP 24	0.843
LARC-RRP 29	0.597
LARC-RRP 38	0.595
LARC-RRP 52	0.670
LARC-RRP 79	0.343
LARC-RRP 81	0.742
LARC-RRP 83	0.535
LARC-RRP 99	0.624



(a) Slice 1, DSC_S = 0.858. (b) Slice 3, DSC_S = 0.925. (c) Slice 5, DSC_S = 0.952.



(d) Slice 7, DSC_S = 0.936. (e) Slice 9, DSC_S = 0.906. (f) Slice 11, DSC_S = 0.923.



(g) Slice 13, DSC_S = 0.903. (h) Slice 15, DSC_S = 0.900. (i) Slice 17, DSC_S = 0.743.

Figure 4.16: A selection of T2w image slices from the patient with highest DSC_P when comparing Table 4.12, 4.13 and 4.14. The predicted delineation made by the model is marked in red, while the manual delineation made by $Radiologist^1_O$ is marked in yellow. The corresponding DSC_S between the model and radiologist is presented for each image slice. The color bar indicates the image intensities.

4.7 INCLUDING DIFFUSION WEIGHTED IMAGES

As a last step of the thesis DWIs were added as an additional channel to the input of the OxyTarget dataset. This was done in order to evaluate whether or not the model performance increased when including another MR sequence. The DWIs used as input had a b-value equal to $500 \frac{s}{mm^2}$. Out of the 110 patients in the OxyTarget data, 109 patients had DWI available. Only the image slices containing tumor were used as input.

As presented in Table 3.2 the DWIs had a total of 1826 image slices due to fewer and thicker slices in some cases, compared to the T2w images. The very same image slices were included when only using the T2w images as input, in order to make the results comparable. Table 4.15 presents the median DSC_P, while Figure 4.17 shows violin plots of the DSC_P achieved on the validation set when adding DWI as an additional input channel. Some image examples of the input images, manual delineations made by the radiologist, and the predictions made by the model when it was trained purely on T2w images or when it was trained on a combination of T2w images and DWIs are presented in Figure 4.18.

Table 4.15: Median DSC_P achieved on the validation patients when only using T2w images as input, and when including DWIs as an additional input. The DWIs had a b-value equal to $500 \frac{s}{mm^2}$.

Dataset	Input	Median DSC_P on Validation Set
OxyTarget	T2w	0.822
OxyTarget	T2w + DWI	0.831

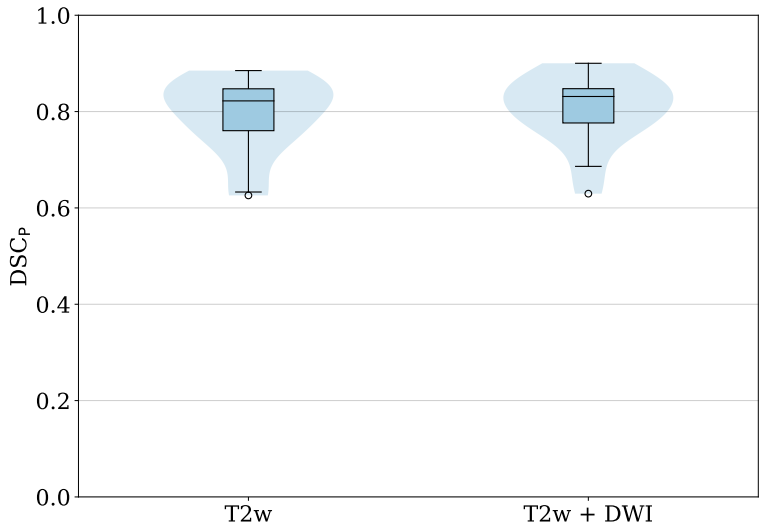


Figure 4.17: Violin plots of the DSC_P achieved on the validation patients when only using T2w images as input, and when including DWIs as an additional input. The DWIs had a b-value equal to $500 \frac{s}{mm^2}$.

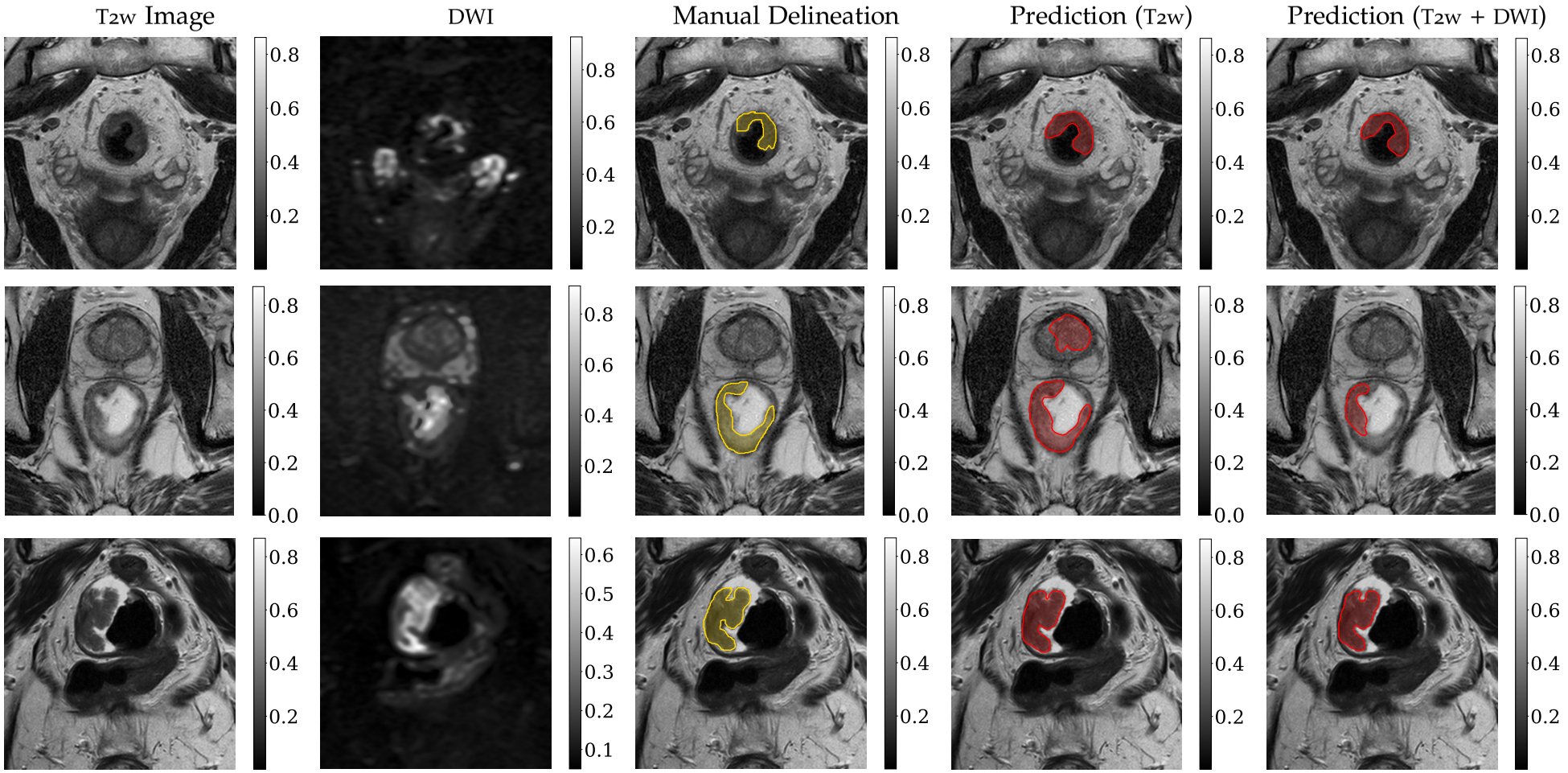


Figure 4.18: Image slices from three different patients in the validation set of the OxyTarget data. The two first columns present the available input images, namely a T2w image and a DWI. The manual delineation made by the radiologist is shown on the T2w image in yellow. The two last columns shows the predicted delineation in red, made by the model when it was trained purely on T2w images (T2w) and when adding DWI as an additional input channel (T2w + DWI). The predictions are presented on the T2w image. The color bar indicates the image intensities.

5 DISCUSSION

The DL models developed in Chapter 4 showed varying model performance depending on the hyperparameters, standardization methods, data augmentation methods, datasets and MR sequences used as input. The final models had a median DSC_P in the range 0.558-0.691 when applied to the test sets. Thus, the models show promising results for performing automatic tumor segmentation in patients with rectal cancer. However, there is still room for improvement. In the following sections we will have a closer look at the results obtained in Chapter 4 with the aim of explaining the varying model performances and how one could further improve the predictions. Lets start off with one of the first steps in the thesis, namely the process of splitting the datasets.

5.1 SPLITTING OF DATASETS

During the thesis it was decided to use the traditional splitting method, as described in Section 2.3.6, to divide the datasets into a training set, validation set and test set. The splitting of datasets have shown to be of great importance for the model performance according to LeBaron et al. [76]. The article demonstrated that the model uncertainty due to data splitting may be of significantly higher impact than other factors such as training, model architecture and initialization. Hence, if the data splits are selected inappropriately the training data, validation data and test data might not be equally representative of the data domain [77]. Consequently, the training set will consist of several features which are not represented in the test set. The generalization to new unseen data will therefore most likely fail.

Accordingly, Section 4.1 made a brief investigation of how the dataset splitting could impact the model performance. The mean DSC_S was calculated over all images in the fold used as validation set. This was done for each training epoch. The epoch which had the maximum mean DSC_S was then used as decision ground for choosing the best model obtained during training. Table 4.1 and 4.2 show that the maximum value of the mean DSC_S for each fold is quite similar for both datasets. However, when calculating the median DSC_P on the fold used as validation set the results showed a higher degree of variation. This is especially true for the LARC-RRP dataset in Table 4.2, where the difference between the minimum and maximum median DSC_P is equal to 0.210. The same trend is presented in Figure 4.2, where Figure 4.2b of the LARC-RRP dataset shows particularly varying results.

Furthermore, the results in Table 4.1 and 4.2 show that the degree of influence the data splitting has on the model performance depends on the dataset

itself. The OxyTarget dataset in Figure 4.2a shows less variability in the median DSC_P than the LARC-RRP dataset. This could indicate that the OxyTarget dataset contains more similar features, and consequently the model performance does not depend as much on the dataset splitting. The higher variability in the median DSC_P for the LARC-RRP data on the other hand, indicates that the LARC-RRP dataset is more dissimilar than the OxyTarget dataset. Still, for both datasets the interpatient spread shows large variations depending on the fold used as validation set.

The results confirm that one should be especially cautious when splitting the datasets. In later experiments one should therefore consider to tune the model parameters for several different dataset splits. In this way one could be more certain that the final model is the most optimal model, which could not have been improved even if the data was split differently. Thus, the 5-fold cross validation method could be implemented with the purpose of tuning the model parameters for each validation fold, and choosing the most optimal dataset split. As a consequence the tuning process would be repeated five times for each dataset. Hence, the approach would require a lot of time and computational power. The splitting issue is therefore a trade off situation between the available time and computational power, and the possible gain in model performance.

As a final note, one should not forget about the test set. A key element for achieving good generalization ability is that the test set represents the dataset in a similar manner as the training set and validation set [77]. Hence, it is important to stratify the splits to make sure that all subsets equally represents the data domain. In addition, this requires extra attention when using external test sets as input to the model. External test sets are not as likely to represent the same data domain as the internal test set, and one should therefore be extra careful when applying the model to external data.

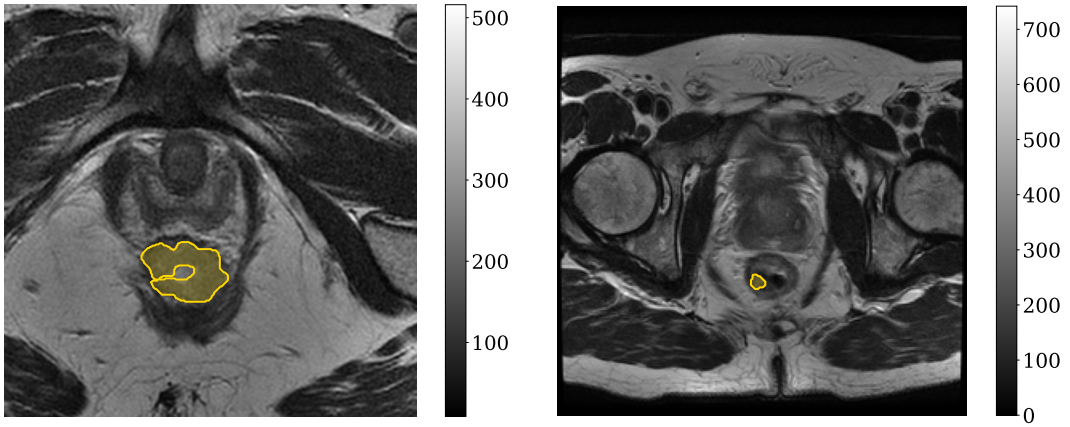
5.2 FINDING THE OPTIMAL MODEL CONFIGURATION

The first part of the thesis focused on finding the optimal model configuration. Thus, the main aim of Section 4.2 was to find the model configuration which gave the overall highest median DSC_P on the validation set. According to Abdi et al. [78] a crucial step of discovering the optimal configuration is to uncover the optimal hyperparameters of the model. Correspondingly, Section 4.2.1 focused on the tuning procedure of the learning rate and loss function for each dataset. Table 4.3 and Figure 4.3 clearly confirm that the model performance depends on the hyperparameter values. The table and figure show how different combinations of learning rates and loss functions result in different median DSC_P on the validation sets. In addition, Figure 4.3 illustrates how the various combinations generate variations in the interpatient spread. The results further demonstrate the importance of implementing the hyperparameter search on a grid structure. In other words, matching all of the selected learning rates with all of the selected loss functions, in or-

der to evaluate the model performance for all possible combinations. If the tuning process had been conducted in a sequential order other hyperparameters would have been chosen. This is confirmed for the Combined dataset in Table 4.3, where the Dice loss function gives the highest performance when combined with a learning rate of $1e-03$. However, when using the Modified Dice loss function together with a learning rate of $1e-03$ the performance decreases. Thus, the Dice loss function together with a learning rate of $1e-03$ would have been chosen as input parameters. Nonetheless, Table 4.3 shows how the Modified Dice loss function combined with a learning rate of $1e-04$ gives the overall best model performance for the Combined dataset when a grid search is used. Hence, a sequential search would be misleading when tuning the two hyperparameters.

Furthermore, Table 4.3 shows the sensitivity in the process of tuning the hyperparameters. For the LARC-RRP dataset a learning rate of $1e-03$ gives the highest median DSC_P when combined with the Dice loss function. However, when the same learning rate is combined with the Modified Dice loss function the median DSC_P on the validation set is equal to zero. As presented in equation (40) the difference between the Dice loss function and the Modified Dice loss function is the removal of the squared operators in the denominator for the Modified Dice. Consequently, the Modified Dice punishes the model prediction less than with the Dice loss function. This adjustment in the loss function makes a great impact on the LARC-RRP model performance. Figure 4.6 presents the predictions made by the model for the same LARC-RRP image slice, when combining different input values of the learning rates and loss functions. The figure further confirms how the prediction fails when the Modified Dice loss function is combined with a learning rate of $1e-03$. For most of the LARC-RRP patients where these values have been used as input parameters the model does not make any predictions at all. In the cases where the model do make a prediction, the prediction consist of small areas as illustrated in Figure 4.6. The rapid decrease in model performance when changing the loss function while using a learning rate of $1e-03$ for the LARC-RRP dataset, indicates that the dataset is extremely sensitive. Oppositely, the OxyTarget dataset are more robust against changes in the input parameters as shown in Table 4.3 and Figure 4.3.

Figure 4.3 further shows how some patients in the LARC-RRP dataset and Combined dataset have a DSC_P equal to zero. When investigating the model performance of the LARC-RRP validation patients into more detail it was found that three patients obtained a median DSC_P equal to zero, or very close to zero, for all combinations of the learning rates and loss functions. These three patients were LARC-RRP 11, LARC-RRP 13 and LARC-RRP 14, and the patients performed equally bad in the Combined dataset. For LARC-RRP 13 and LARC-RRP 14 the tumor sizes were extremely small which makes the segmentation task even more demanding. However, a common factor for all three patients was that they all had image dimensions equal to 256×256. In fact, these were the only patients in the validation set with the smaller im-



(a) 352 × 352 (b) 256 × 256

Figure 5.1: An example of how the FOV differs when comparing image slices of different sizes. The image slice in (a) has been cropped, and as a consequence it does not contain the same information as in (b). The color bar indicates the image intensities.

age dimensions. Hence, the continuous poor performance on these patients indicates that the model cannot handle images with a different size than the majority of the images. It should be noted that the training set consisted of two patients with dimensions equal to 256 × 256, while 61 patients had a size equal to 352 × 352. Thus, the model was mainly trained on images of size 352 × 352, and the weights were adjusted accordingly. One should also recall that the images of size 352 × 352 were cropped, while the images of size 256 × 256 were not cropped in any way. Consequently, the images of size 352 × 352 contained a different FOV than the images of size 256 × 256, as illustrated in Figure 5.1. The different image content makes it extra challenging for the model to correctly recognize the tumor location for the smaller images. In future experiments one should therefore be extra cautious when applying images of different dimensions as input to the model. If different image dimensions are included, one should make sure that the various dimensions are equally represented in each subset.

5.2.1 *Standardization of Input Data*

As previously introduced in Section 2.1.4 one of the main issues with MR images are the non-uniform pixel intensities. The non-uniform intensities in the MR images consist of anatomically irrelevant intensity variation throughout the data [79]. Thus, Section 4.2.2 investigated two different standardization methods of the MR images in order to evaluate if this could further improve the model performance.

Table 4.4 and Figure 4.7 show that the standardization methods have a significant impact on the model performance. This is especially true for the LARC-RRP dataset and Combined dataset. Furthermore, the table and figure show

how the model performance improved for the LARC-RRP dataset and OxyTarget dataset when applying the Z-Score method, while the performance decreased when applying the MH method to the same datasets. Oppositely, the Z-Score method decreased the model performance when applied to the Combined dataset, while the MH method increased the model performance for the same dataset. One might have expected the same tendency to occur for the Combined dataset as for the LARC-RRP dataset and OxyTarget dataset. A possible explanation of the opposing trend could be the use of different MR scanners in the LARC-RRP dataset and OxyTarget dataset. Hence, when performing the Z-Score method on the Combined dataset some information which initially distinguished the two individual datasets has been removed. As illustrated in Table 4.4 the contribution from the OxyTarget dataset, when running the model on the Combined dataset without any standardization, was equal to 0.700. However, when applying the Z-Score method the contribution from the OxyTarget dataset decreased to 0.566. This indicates that some of the features which are important for the model to recognize the tumors in the OxyTarget data have been removed. For the LARC-RRP data on the other hand, the contribution increased when applying the Z-Score method indicating that making the pixel ranges more similar helps when the model tries to recognize the LARC-RRP patients.

The improvement in model performance for the Combined dataset when implementing the MH method can again be explained by the fact that the Combined dataset consisted of two individual datasets with distinct features. As illustrated in Figure 3.6a the two datasets consisted of two very different pixel distributions. This is once more due to the use of different MR scanners for each dataset. Hence, the possible gain of matching the histograms is much larger for the Combined dataset where the pixel distributions are extremely different, than for the LARC-RRP dataset and OxyTarget dataset where the pixel distributions are more similar from the beginning. Still, Figure 4.7 clearly shows that there is a higher degree of interpatient spread when using the MH method on the Combined dataset. The increased size of the interpatient spread when using the MH method can also be recognized for the OxyTarget dataset in Figure 4.7a.

In the Combined dataset one can notice how the contribution from the OxyTarget dataset increases, while the contribution from the LARC-RRP dataset decreases, when applying the MH method in Table 4.4. This could be explained by the fact that the LARC-RRP patients were matched with the histograms from an OxyTarget patient as reference. Consequently, the LARC-RRP images became more similar to the OxyTarget images. Since all of the images in the dataset are more similar to the OxyTarget images the model is more likely to correctly predict the OxyTarget images in the validation set. Thus, one might argue that when implementing the MH method one should use the dataset with the worst image quality as reference. Hopefully, this could further increase the performance for the most challenging dataset.

The combined method MH + Z-Score turned out to be the overall best standardization option, as shown in Table 4.4 and Figure 4.7. The most significant increase in model performance was achieved for the Combined dataset. This is mainly due to the fact that the Combined dataset had a more dissimilar data domain to begin with, since it consisted of two individual datasets. Thus, the room for improvement when standardizing the images is larger than for the LARC-RRP dataset and OxyTarget dataset, which consisted of more similar features from the beginning.

As a final note one can also notice how the OxyTarget dataset seems more robust to the different standardization methods, while the LARC-RRP dataset is affected in a higher degree. This could again be a consequence of the different MR scanners. A newer MR scanner of a different brand was used when imaging the OxyTarget dataset. As a consequence the MR images in the OxyTarget dataset might have a less degree of non-uniform intensities compared to the LARC-RRP images.

5.2.2 *Data Augmentation*

It is a well known concept that bigger datasets result in better DL models [20, 21]. However, in several cases the amount of available data can be extremely limited. This is often the case for medical images due to the time-consuming task of labeling the images accurately. In this thesis the largest dataset consisted of MR images from 199 patients, with a total of 5942 images. In the process of training a DL model this is considered as a relatively small dataset [80, 81]. Thus, data augmentation was investigated as a final step of finding the optimal model configuration.

Two different data augmentation methods were investigated as presented in Section 4.2.3. The Default method consisted of more alternations in the images than the BC method, which did not make any changes in the image brightness or image contrast[3]. For all datasets Table 4.5 and Figure 4.8 show an overall lower model performance on the validation set when the Default method was applied. For the OxyTarget dataset and LARC-RRP dataset this is also the case for the BC method. However, one should recall that the main aim of the data augmentation is to mitigate overfitting and make the model more robust. Hence, the decrease in model performance could indicate that the initial models are not very robust to changes in the datasets.

The idea of data augmentation is that more information can be extracted from the original dataset through augmentations [82]. By transforming existing images while keeping their labels preserved, one can artificially inflate the training dataset size. The transformations of the existing images implemented by the two augmentation methods used in this thesis are considered as geometrical transformations. Geometrical transformations are good solutions for

[3]See Appendix C.1 and C.2 for the exact configurations of the data augmentation methods.

positional biases in the training data [82]. Thus, the decrease in performance when implementing the augmentation methods indicates that there are some positional bias in the two individual datasets. If the tumor is located at the same position in every image, the model is more likely to memorize the tumor position rather than learning how to recognize the tumor pixels in the images. By applying geometrical transformations to the training images the model is forced to locate the tumor at other positions as well. Hence, even though the model performance on the validation set decreases for the Oxy-Target data and LARC-RRP data the models are most likely able to generalize better to new unseen data when data augmentation is implemented.

It should be noted that the BC method on the Combined dataset improved the model performance on the validation set, as shown in Table 4.5 and Figure 4.8c. When examining the contribution from each of the individual datasets within the Combined dataset both of the datasets showed an improvement when using the BC method. Thus, the result suggest that the changes in image brightness and image contrast introduced in the Default method adds some extra noise to the Combined dataset which makes it more difficult to correctly predict the tumors. When removing these changes in the BC method the model performance increased, and one can argue that the BC method is the most optimal augmentation for the Combined dataset.

According to Shorten et al. [82] the biases distancing the training data from the test data are more complex than transitional and positional variances for medical images. Hence, other data augmentation methods should be considered when using medical images. Some suggestions of other data augmentation methods are therefore presented in Section 5.7.

5.2.3 *A Complex Task*

The results in Section 4.2, and the discussion so far, clearly show the complexity of finding the most optimal model configuration. The field of optimizing the model hyperparameters is enormous, and one can spend a great amount of time trying to find the best model configuration. In this thesis a brief tuning of the model parameters have been carried out. As previously stated, several other hyperparameters were already fixed based on recommendations from experienced scientists at the Faculty of Science and Technology at NMBU, as presented in Table 3.8. The ideal situation would be to tune all of the hyperparameters listed in Table 3.8 and Table 3.9 by testing all possible combinations. This would be an extremely cumbersome task to perform manually. Thus, algorithmic approaches for optimizing the hyperparameters have therefore been proposed in the literature such as grid search, random search and sequential search [83]. However, one should keep in mind how large the possible benefits are of tuning the hyperparameters so carefully. The process is definitively a trade off between possible performance gain and time usage. One should therefore consider other aspects of the model such

as the dataset, initialization and architecture, which also could improve the model performance greatly.

Finally, the first part of the thesis with the aim of finding the most optimal model configuration is wrapped up in Section 4.2.4. The final model parameters which gave the overall best performance for all datasets are presented in Table 4.6. Hence, the next part of the thesis consisted of evaluating the developed models using the parameters listed in Table 4.6 as input.

5.3 MODEL PERFORMANCE

In Section 4.3 the OxyTarget model was evaluated against a second radiologist (Radiologist^2_O). Table 4.7 shows how the median DSC_P is quite similar when comparing the model predictions with Radiologist^1_O and Radiologist^2_O. Still, the performance is lower than the interobserver DSC_P between the two radiologists. Figure 4.10 further confirms that the performance when comparing the OxyTarget model with the two radiologists are fairly similar. However, the interpatient spread when comparing the predictions with Radiologist^2_O shows a higher degree of variability compared to Radiologist^1_O. Thus, the model is slightly favoring Radiologist^1_O over Radiologist^2_O. This is as expected since the model was trained on Radiologist^1_O.

According to Table 4.7, the interobserver DSC_P between the two radiologists has a median DSC_P of 0.821. Hence, the radiologists agree most of the time on where the tumors are located in the MR images. Consequently, if the model performs well for one of the radiologists, it will perform well for the second radiologist and vice versa. This is because the two delineations are relatively similar from the beginning. It is therefore not surprising that the OxyTarget model achieved a similar score for both radiologists. Figure 4.11 shows the DSC_P for each patient in the validation set when comparing the model prediction with the two radiologists and calculating the interobserver DSC_P. The figure shows how the interobserver DSC_P is close to 0.800 for all of the patients, except for one patient. Patient ID 122 has an interobserver DSC_P equal to 0.451. Three image slices are presented as example of the dissimilar delineations in Figure 5.2. Radiologist^1_O is marked in yellow while Radiologist^2_O is marked in purple. Figure 5.2a is especially noteworthy since the radiologists do not agree whether or not there is a tumor in the image slice. The dissimilarities in the manual delineations result in a performance gap when comparing the model predictions with the two radiologists, as illustrated for Patient ID 122 in Figure 4.11. The figure shows that the model prediction achieved the highest DSC_P when compared with Radiologist^1_O. Hence, the result further suggest that the model prediction is biased towards the delineation made by Radiologist^1_O.

Another patient which stands out in Figure 4.11 is Patient ID 124. For this patient the interobserver DSC_P is equal to 0.799, while the DSC_P when comparing the prediction to Radiologist^1_O and Radiologist^2_O is equal to 0.333 and

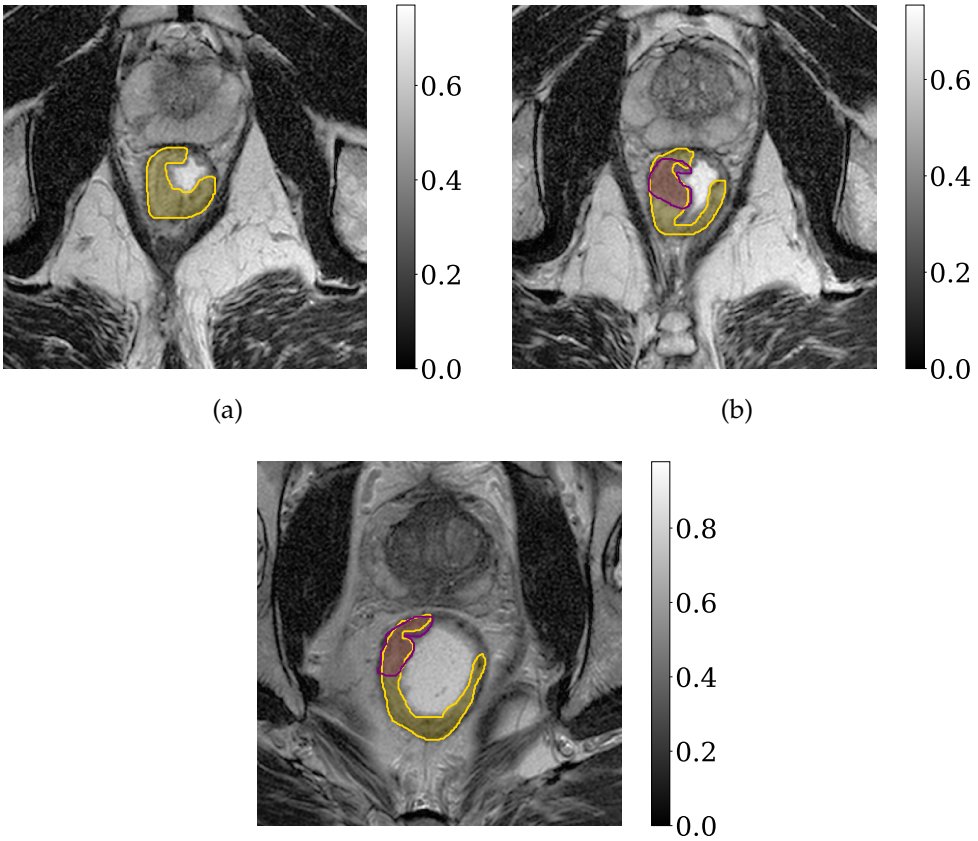


Figure 5.2: Example of three image slices where the manual delineations made by Radiologist^1_O (yellow) and Radiologist^2_O (purple) shows noteworthy dissimilarities. The DSC_P between the two radiologists is equal to 0.451. The color bar indicates the image intensities.

0.274, respectively. A possible explanation of the poor performance on the patient could be the location of the tumor. The tumor is located higher up in the patient, which is not as common for the other patients in the dataset. As a consequence the model might have learned that there are not any tumors in the last slices of the patients. In addition, when the tumor occurs late in the image slices the content in the images are different. If most of the patients do not have any tumor in these images it will be difficult for the model to make a correct prediction. This is another indication that the model could be positional and translational biased, as previously discussed in Section 5.2.2. The result further indicates that the model is memorizing the position of the tumor rather than learning how to recognize the tumor pixels in the images.

The interobserver DSC_P for the 76 patients in the OxyTarget dataset which were delineated by both radiologists was equal to 0.805. Ideally, the model should therefore achieve a median DSC_P of 0.805 or higher in order to perform equally well as the radiologists. However, it should be noted that the score is only an estimate between two radiologists and would therefore most likely decrease when comparing additional radiologists. An article written

by Franco et al. [84] studied the interobserver variations in delineation of rectal cancer for multiple radiation oncologists. In the study 10 radiation oncologists participated in the delineation of rectal cancer in two patients. In order to determine the interobserver variations the study compared the delineations made by the radiation oncologists with a ground truth delineation performed by an experienced radiation oncologist dedicated to rectal cancer treatment. This gave a DSC_P of 0.800 for the first patient, and a DSC_P 0.650 for the second patient. The DSC_S for the radiation oncologists ranged between 0.760-0.860 for the first patient, and between 0.580-0.790 for the second patient. In a second article written by Wang et al. [85] 20 patients with rectal cancer were delineated by two experienced radiologists. The interobserver DSC_P between these two radiologists was equal to 0.710. The lowest interobserver DSC_P achieved on the validation patients in Figure 4.11 is equal to 0.451. One could therefore argue that all predictions made by the model where the DSC_P is above 0.451 is sufficient. However, based on the literature the DSC_P for a patient should be in the range 0.650 to 0.800. It should also be noted that the manual delineations used as ground truth are associated with uncertainty. Manual tumor delineation is an ambiguous task, thus a biased model towards one radiologist should be avoided. A score higher than 0.800 could therefore indicate that the model has learned a delineation pattern which is specific for one radiologist.

5.3.1 *Comparison With Shallow Machine Learning Models*

As introduced in Section 2.4 CNNs have shown to be especially effective in computer vision problems and have become the standard network structure for a wide variety of computer vision tasks [7]. In order to further confirm the usefulness of CNNs Section 4.4 compared the DL models with the SML models obtained from the author's project thesis. Table 4.9 and Figure 4.12 show the median DSC_P and violin plots of the DSC_P, respectively. By examining the table and figure it is clear that the DL models surpass the SML models when the OxyTarget dataset was used as input. For the Combined dataset the median DSC_P is also higher for the DL model than for the SML model. However, Figure 4.12c shows that the interpatient spread of the DL model for the Combined dataset is larger than for the SML model. In a similar manner, the LARC-RRP dataset shows a larger interpatient spread when using the DL model compared with the SML model. In addition, the LARC-RRP dataset obtained a lower median DSC_P when applying the DL model. Hence, the LARC-RRP dataset shows an overall poorer performance with the DL model than with the SML model. In the next paragraphs we will discuss possible reasons why the LARC-RRP dataset performed worse with the DL model than with the SML model, while the OxyTarget dataset and Combined dataset performed better with the DL model than with the SML model.

When comparing the DL models and SML models it should be pointed out that the cropping of the input images were performed differently. As described in

Section 3.4.1 the input images for the DL models were cropped to a standard size of 352×352. This was achieved by cropping the images equally at all edges. For the SML models on the other hand, the images were cropped by creating a 3D bounding box around the tumor while adding a 10 mm margin in all directions, which was restricted by the FOV. Consequently, the SML cropping returned a 3D bounding box where the tumor was located at the same position in all cases, namely 10 mm away from the cropped image edge. Oppositely, the DL cropping resulted in images where the tumor position alternated more. Hence, it would be more challenging for the DL model to segment the tumor correctly. For the SML model on the other hand, the tumor was located approximately at the same place in each image, thus making it easier for the model to segment the tumor. This could explain why the SML model performed better for the LARC-RRP dataset than the DL model. One might argue that a similar cropping method as for the SML model should also be used for the DL model. However, with the SML cropping all of the images would have different sizes which would be challenging to deal with in a DL model. This is mainly due to the fact that the input images to the classification layer in a DL model should have a fixed size, as stated in Section 2.4.1. In addition, the centered tumor position in the SML cropping is not necessarily a positive feature. If the tumor is located at approximately the same position in each image the model is more likely to memorize the position rather than learn the specific features which determines whether or not a pixel in the image is tumor. Thus, by keeping the alternating positions of the tumors the model was forced to learn the corresponding tumor features and the probability of achieving a better generalization performance increases.

Another aspect to consider when analyzing the different cropping approaches is the fact that the SML cropping removes several image slices which do not contain any tumor. The slice thickness in the LARC-RRP dataset was 5 mm. Consequently, a 3D bounding box with 10 mm margin would result in leaving two image slices before the tumor begins and after the tumor ends. Oppositely, the DL cropping does not remove any image slices. Thus, all of the DL models so far have used all image slices as input. The effect of removing the image slices which did not contain any tumor is further discussed in Section 5.3.2.

Finally, it should be noted that the SML models have used post-processing on the initial predictions. The post-processing is described in Section 3.8.1, and consisted of simulating clicks in the initial prediction where the tumor delineated by the radiologist was located. The aim of the post-processing was to remove all predicted regions which were not connected to the tumor region delineated by the radiologist. However, this type of post-processing would require a radiologist to manually go through all of the image slices and clicking within the tumor area. The DL models on the other hand, do not require any manual post-processing which is preferable in a clinical setting.

5.3.2 *Impact of Tumor Slices*

When comparing the LARC-RRP dataset and OxyTarget dataset the number of image slices which contains tumor proved to be significantly different. The percentage of tumor slices in the OxyTarget dataset is equal to 71.2% while for the LARC-RRP dataset the percentage of tumor slices is equal to 29.3%. This is most likely one of the main reasons why the DL model performed better on the OxyTarget dataset than for the LARC-RRP dataset. When the number of tumor slices in the LARC-RRP dataset is small, it will be fewer instances where the model gets the opportunity to learn the tumor features. This can be easier understood in the analogy of a human student. Imagine a student who tries to learn how to recognize and segment tumors in MR images. The student has 10 images available where three of them contains tumor. A teacher is present in order to help the student correctly recognize and segment the tumors. Later, the student is asked to recognize and segment the tumors in 10 completely new images. The new set of images has again a subset of three images which contains tumor. However, the student is not aware of how many images with tumor to expect. In this case the student is likely to make errors due to the limited amount of tumor images available in both the training set and test set. On the other hand, if there were 7 images in the training set the student would be much more likely to correctly learn how to recognize and segment the tumors. Accordingly, the same argument would hold for a DL model.

The suspicion that the balance between tumor slices and non-tumor slices in the datasets had a significant impact on the DL model performances was further confirmed in Section 4.5. Figure 4.13 shows the training performance when all image slices were used as input and when solely using tumor slices as input for the LARC-RRP model. Figure 4.13a clearly shows how the model was not able to learn anything during training when all of the image slices were used as input. Oppositely, Figure 4.13b shows how the model was able to learn when only using tumor slices as input. Table 4.10 and Figure 4.14 show a clear difference in the model performance for all datasets when exclusively using tumor slices as input, compared with using all image slices as input. Figure 4.14 shows how the interpatient spread was reduced in all cases when solely using tumor slices. In addition, the median DSC_P on the validation set increased for all datasets when using tumor slices as input, as presented in Table 4.10. In fact, the performance of the LARC-RRP DL model exceeded the LARC-RRP SML model when only using tumor slices as input. Thus, the DL models outperformed the SML models for all of the datasets when exclusively using tumor slices as input.

The discussion so far demonstrates the importance of having a balanced dataset. In addition, the poor performance when including all image slices as input can be explained by the choice of performance metric. The impact of the performance metric and why it is not suitable when including all image slices are further discussed in Section 5.3.4. However, the DL models showed

a greater performance than the SML models when exclusively using tumor slices as input. It is therefore confirmed that the DL models are preferred in the image segmentation tasks.

5.3.3 *Generalization Ability*

As a final step of evaluating the model performances, the generalization abilities were tested by applying the internal test set for each dataset. The test sets are the ultimate test for evaluating how well the models are able to predict new unseen data [7, 86]. Table 4.11 and Figure 4.15 present the median DSC_P and violin plots of the DSC_P achieved on the test sets. For the OxyTarget dataset and Combined dataset the median DSC_P decreased when the test set was used, compared to the median DSC_P on the validation sets in Table 4.10. However, as previously discussed in Section 5.1, this is a common problem when applying the models to new unseen data [76, 77]. The reduction in the median DSC_P for the OxyTarget dataset and Combined dataset indicates that the models have started to overfit towards the training sets. For the LARC-RRP dataset on the other hand, the model performance on the test set increased slightly compared to the median DSC_P in Table 4.10. It should be noted that the LARC-RRP test set only included one patient with images of size 256×256, while the validation set contained three patients with images of size 256×256. As previously discussed in Section 5.2 the patients with the smaller image size are more difficult to correctly predict. Thus, the LARC-RRP dataset could be considered to have an easier test set compared to the OxyTarget dataset.

Another explanation of why the LARC-RRP dataset generalizes better than the OxyTarget dataset is the higher degree of data variation in the LARC-RRP dataset. When investigating the LARC-RRP dataset in more detail one can notice how the position of the tumor in the images varies more than in the OxyTarget dataset. The OxyTarget dataset seems to have several larger tumors located approximately at the center of the image, while for the LARC-RRP dataset the tumors can also be found in the corners of the images or close to the image edges. In addition, the LARC-RRP dataset consist of images from two different MR scanners. This is another factor which creates more variability in the data domain. Furthermore, the LARC-RRP dataset consist of patients with two different image dimensions. All of these factors generates a dataset with more variability, which forces the DL model to learn the tumor features rather than memorizing them. Consequently, the model trained on the LARC-RRP dataset can be considered as more robust.

Table 4.12, 4.13 and 4.14 present the DSC_P obtained for each patient in the test set for each dataset. Table 4.12 includes the interobserver DSC_P as well, where it is available. For most patients the predicted DSC_P and the interobserver DSC_P are quite similar. The comparison to the interobserver DSC_P is a great tool for evaluating how good the predicted delineation is. Thus, it would be useful to include additional delineations made by Radiologist^2_O. In this way

one could calculate the interobserver DSC_P for all patients in the OxyTarget dataset.

For the LARC-RRP test patients in Figure 4.15 the interpatient spread is larger than for the OxyTarget test patients. Thus, the predictions are more variable even though the median DSC_P increased slightly compared to the validation patients in Figure 4.14b. In Table 4.13 one can notice that LARC-RRP 15 has the lowest DSC_P. This patient has image dimensions equal to 256×256. Thus, the poor performance on the patient further confirms the issue of including images with different dimensions.

Table 4.14 shows how the DSC_P changes slightly for most patients compared to the DSC_P presented in Table 4.12 and 4.13. Hence, the result indicate how the model learns different features during training when combining the two individual datasets. Still, for the Combined dataset the median DSC_P also decreased when applying the test set. However, Table 4.11 shows that the contribution from the OxyTarget dataset and LARC-RRP dataset in the Combined dataset has increased. In other words, the median DSC_P on the test set increased for both the OxyTarget dataset and LARC-RRP dataset when combining the two individual datasets during training. The result therefore confirms that bigger datasets give better DL models. Hence, combining multiple cohorts in order to create larger datasets could therefore be a useful approach when training DL models.

5.3.4 *The Importance of Performance Metrics*

Figure 4.16 presents a selection of image slices from the patient with the highest DSC_P based on Table 4.12, 4.13 and 4.14. The images show how the model is able to make accurate predictions, and how difficult it is to distinguish the manual delineation made by the radiologist in yellow from the model prediction in red. However, in Slice 17 the model makes a mistake and delineates a region of healthy tissue as tumor. Still, the slice has obtained a DSC_S of 0.743 which has previously been considered as a good score in Section 5.3. One could therefore question the reliability of the DSC performance metric when a prediction which falsely delineates healthy tissue is still able to achieve a high DSC.

Performance metrics are essential for assessing the performance of segmentation models in an objective and meaningful manner [87]. Thus, using an appropriate performance metric when evaluating the model performance is crucial in medical image analysis. Reinke et al. [87] therefore investigated the possible pitfalls related to the most frequently used metrics in medical image segmentation tasks. The article points out several weaknesses with the DSC (34) metric. Firstly, the DSC may not be the appropriate metric for segmentation of small structures. The main reason is that a change in one of the predicted image pixels will result in a large impact on the calculated DSC for small structures compared to large structures. This could explain the espe-

cially poor performance for the patients with very small tumor sizes in the LARC-RRP dataset, as previously pointed out. In addition, shape unawareness is another common weakness with the DSC metric. Since the DSC metric simply measures the overlap between objects it is not designed to uncover differences in shape [87]. Thus, predictions with completely different shapes may lead to the same DSC. This is especially problematic in radiotherapy where the shape is an essential feature. Furthermore, the DSC gives a higher score for oversegmented objects compared to undesegmented objects. Hence, the DSC can be misleading and one should be cautious when utilizing it.

In addition, the DSC is not suitable for detection and localization tasks. This was previously confirmed in Section 5.3.2, where it was clear that the model performance increased when solely using tumor slices as input. Hence, the result demonstrated how the DSC is not suited for detection tasks. In general the DSC is strongly biased towards single objects and is therefore not appropriate for detecting multiple object structures [87]. Thus, the DSC should be used to assess segmentations of a single object, and not for detecting multiple objects in the image.

When evaluating the performance of a model one should generally use more than one metric in order to avoid the pitfalls pointed out. Hence, multiple metrics with different properties should be used when evaluating the model. For later experiments another performance metric which evaluates the distance between the ground truth and the prediction should be included. The Mean Surface Distance (MSD) [88] or the Hausdorff Distance (HD) [87] are two possible metrics which analyses the distances between segmented objects.

5.4 DIFFERENT MAGNETIC RESONANCE SEQUENCES

In the end, DWIs were included as additional input in the OxyTarget dataset in Section 4.7. This was done in order to evaluate whether or not the model performance could increase by including additional MR sequences. Table 4.15 presents the median DSC_P on the validation set, while Figure 4.17 compares the DSC_P violin plots when only using T2w images as input and when using DWIs as additional input. The table shows a slightly higher median DSC_P when including the DWIs as input. In addition, the interpatient spread in Figure 4.17 is reduced when including DWIs, compared to solely using T2w images as input. Some image examples of the model predictions when only using T2w images as input and when including DWIs as additional input are presented in Figure 4.18. For the first and last row the predictions made by the models are quite similar when using different MR sequences as input. However, for the middle row there is a significant difference between the predictions when T2w images were used as input and when including DWIs as input. When the model was trained solely on T2w images the model prediction covers the tumor area delineated by the radiologist quite nicely. However, the model predicted an additional area of the image as tumor. This area is quite large, and consists of healthy tissue. Thus, the model prediction would

lead to radiation of healthy tissue. Still, the prediction obtained a DSC_S equal to 0.665. When using DWIs as an additional input a smaller region of the tumor is correctly predicted. However, this model does not predict any regions outside the tumor area marked by the radiologist. Hence, the healthy tissue is not harmed during treatment. Having said that, the prediction made when applying DWIs as additional input obtained a DSC_S equal to 0.510. This is a significantly lower DSC_S than when using the T2w images as input alone. Thus, the prediction with T2w images as input obtained a higher DSC_S even though more healthy tissue was falsely delineated as tumor. The image example clearly demonstrate one of the major weaknesses of using the DSC performance metric alone for evaluating the model performance, as previously discussed in Section 5.3.4.

Nevertheless, Section 4.7 shows promising result for including DWIs as additional input. It should be noted that the DWIs used as input during this thesis had a b-value equal to $500\frac{s}{mm^2}$. Thus, it would be interesting to include DWIs with different b-values in order to investigate whether or not this can increase the model performance further. One should also note that the model performance in Section 4.7 is at the upper limit of what is considered as acceptable when comparing the predictions to manual delineations, as previously discussed in Section 5.3. It could therefore be suspected that the inclusion of DWIs will have higher impact on models with poorer performance. For that reason it would be interesting to include DWIs as additional input for the LARC-RRP model. The model perform worse than the OxyTarget model when only using T2w images. Hence, the possible gain of including additional MR sequences is much larger.

5.5 THE DATASETS

So far, each dataset has been discussed separately, not paying to much attention on why the three datasets perform differently. However, throughout Chapter 4 there is a consistent trend that the OxyTarget model outperform the LARC-RRP model and Combined model. The LARC-RRP model showed the lowest performance in all sections, while the Combined model had a performance which was located between the OxyTarget model and LARC-RRP model. In the following paragraphs an explanation of possible reasons why the datasets perform differently will be given.

One possible explanation of why the LARC-RRP model had a lower model performance is the limited image quality in the dataset. As previously introduced in Section 3.1 and 3.2 the LARC-RRP patients and OxyTarget patients were imaged with different MR scanners. The OxyTarget patients were imaged with the Philips Achieva 1.5-T system (Philips Healthcare, Best, the Netherlands) which gave images with voxel sizes equal to $(0.352, 0.352, 2.75)$ mm. The LARC-RRP patients were either imaged with the 1.5-T GE Signa® LS scanner (GE Healthcare, Milwaukee, WI) or the 1.5T Siemens Espree scanner (Siemens, Erlangen, Germany) which gave images with voxel sizes equal to

$(0.391, 0.391, 5.0)$ mm and $(0.375, 0.375, 5.0)$ mm, respectively. Thus, the OxyTarget data included thinner image slices of the tumor with better resolution compared to the LARC-RRP data. In 2019 Le Hou et al. [89] showed that by increasing the resolution of medical images the segmentation accuracy increased correspondingly. This is mainly due to the fact that with higher image resolution the DL model is able to recognize more fine structure patterns. Thus, the poor performance on the LARC-RRP model compared to the OxyTarget model could be explained by the difference in image quality. One could therefore be tempted to only use images of very high resolution. However, by increasing the image resolution the computational power needed to process the data increases significantly. Hence, there is a trade-off between increased image resolution and computational capacity available.

Another aspect to consider when comparing the two datasets are the number of patients and images in each dataset. The OxyTarget data consisted of 110 patients with a total of 2791 image slices, while the LARC-RRP data consisted of 89 patients with a total of 3133 image slices. Thus, the OxyTarget data had less image slices compared to the LARC-RRP data, even though the OxyTarget data consisted of more patients. However, it should be noted that the OxyTarget data had a significantly higher portion of image slices which contained tumor compared to the LARC-RRP data. When removing all of the image slices without any tumor the LARC-RRP dataset was left with 917 image slices, where 672 were used for training. The OxyTarget dataset on the other hand was left with 1988 image slices, where 1408 were used for training. Hence, the training set of the LARC-RRP model was approximately half the size of the training set used in the OxyTarget model. As already stated, bigger datasets result in better DL models [20, 21]. Consequently, the small portion of training images in the LARC-RRP dataset could explain why the model performed worse compared to the OxyTarget model.

Furthermore, the LARC-RRP dataset consisted of images with two different sizes, namely images of size 352×352 and 256×256. The OxyTarget dataset on the other hand consisted solely of images with the same size, equal to 352×352. For the LARC-RRP dataset there were in total six patients with images of size 256×256; two in the training set, three in the validation set and one in the test set. The consequences of including images of two dimensions have already been discussed in Section 5.2 and will therefore not be discussed in further detail here.

Lastly, it should be pointed out that the tumor delineation on the T2w images were done by two individual radiologists, one for each dataset. Variations between the consistency of the delineations between the two datasets is therefore another factor which can contribute to different model performances.

It is clear that there are some distinctions between the two datasets which resulted in different model performances. Differences in image quality, image dimensions and the number of available training images are important factors to consider when choosing the datasets used as input for a DL model.

5.6 CLINICAL IMPACT

The ultimate goal of developing a DL model for automatic tumor segmentation is to implement it in a clinical setting. The ideal model would be able to take any image as input and correctly delineate the tumor with an accuracy equally well as the manual delineation made by the radiologist. This would save an enormous amount of time for the radiologists or oncologists. A similar model could also delineate the organs at risk, and further decrease the work load of the oncologists. By implementing such a model it would be possible to change treatment plans between radiation fractions, also known as adaptive radiotherapy. This would further optimize the treatment response for the individual patients.

This thesis shows promising results for implementing DL models in a clinical setting. The final OxyTarget model showed a median DSC_P of 0.691 on the test set, which is within the acceptable range of 0.650 to 0.800 as discussed in Section 5.3. This is also the case for the Combined dataset which showed a median DSC_P of 0.673 on the test set. The LARC-RRP test set did unfortunately not have a median DSC_P within the acceptable range. However, some patients still have a satisfying DSC_P as presented in Table 4.13.

A possible way of implementing the DL models in the clinic would be to introduce a threshold value which determines whether or not the patient should be manually re-delineated. Based on the literature presented in Section 5.3 an acceptable threshold could be a DSC of 0.700 [84, 85]. The threshold value could either be applied on a per patient basis or on a per slice basis. However, in order to be completely sure that all image slices of a patient are acceptable one should apply the threshold on each image slice. In this way all of the image slices with a DSC_S above 0.700 should only be visually inspected by the radiologist, while the image slices below the threshold should be manually re-delineated. Even though this is not an optimal solution to the automatic segmentation problem it would still save time for the radiologists and reduce the need of manual delineation for each image slice.

As discussed in Section 5.3 a very high DSC indicates that the model is biased towards the radiologist used as ground truth. Hence, one should also consider implementing an upper threshold value to avoid biased model predictions. Another option is to compare the predicted delineation with the histology of the tumor, which is the ultimate ground truth. Histology is a method that studies the microscopic anatomy of biological tissues, and therefore gives a precise determination of where the tumor cells are located. The uncertainty associated with using manual delineations as ground truth would then be removed. According to the national guidelines, whole-tumor histology should be completed for all patients with rectal cancer [4]. Hence, the tumor removed during operation can be used as ground truth for patients not treated with preoperative radiation and chemotherapy. Unfortunately, it is challenging to preserve the tumor orientation in the removal process. As a consequence it

is though to compare the whole-tumor with the predicted segmentation. Furthermore, only fragments of the tumor will be left after treatment for patients who undergo preoperative radiation and chemotherapy. Hence, it is impossible to compare the predicted tumor segmentation with the whole-tumor histology for these patients. The implementation of histology as ground truth is therefore difficult to accomplish.

Another important aspect to consider are the input images. The models developed in this thesis should solely take image slices containing tumor as input. Hence, in a clinical setting the radiologist should go through all of the image slices while classifying whether or not the slices contain tumor. Another option is to implement an algorithm trained specifically for selecting the image slices with tumor. Either way the predicted tumor segmentation should always be visually inspected by an expert. This is due to the fact that an error in the delineation could lead to an inaccurate treatment plan, which could result in fatal consequences for the patient.

Even though the developed DL models come with certain limitations the models would still save time for the radiologists, and increase the efficiency in the delineation process. In addition, the models could be useful in research areas such as quantitative image biomarkers and radiomics. Radiomics uses data extracted from the segmented tumors in order to explore the tumors into more detail [90, 91]. In this way a more personalized treatment for the cancer patients can be achieved. However, radiomics requires large datasets of delineated tumors. Thus, an automatic segmentation approach would be highly beneficial.

5.7 FURTHER WORK

There are several modifications which can be implemented in order to improve the model performance. One option is to use the 5-fold cross validation method in order to find the most optimal dataset split, as suggested in Section 5.1. In this way one could be more certain that the final model is the most optimal model. There are several other alternatives to how one could increase the model performance further. In the following subsections possible approaches which could be included in future work are presented.

5.7.1 *Model Configuration*

As discussed in Section 5.2 the process of finding the most optimal model configuration is a complex and time-consuming task. In this thesis a selection of different learning rates and loss functions have been investigated. However, a more thorough investigation where multiple hyperparameters are tuned against each other could potentially further increase the model performance. It would therefore be interesting to test different activation functions, optimizers and batch sizes in addition to different learning rates and loss functions.

One could also change the number of layers in the network and the number of neurons in order to optimize the architecture of the network. Since the process of tuning the various hyperparameters is a time-consuming task it would be beneficial to investigate different algorithms for automatic hyperparameter optimization. Bergstra et al. [83] showed that an algorithmic approach for tuning the hyperparameters in a DL network resulted in a better model performance than when humans tuned the hyperparameters manually. They proposed possible algorithms for sequential search and random search which could be interesting to investigate further. Furthermore, Soon et al. [92] investigated the effect of implementing the stochastic method of particle swarm optimization in order to optimize the architecture and hyperparameters of a CNN. The result showed promising result for bridging the gap between hyperparameter optimization and computational efficiency without reducing the model performance.

As stated in Shorten et al. [82] geometrical data augmentations are not always suitable for medical images. This is due to the fact that medical images are more complex and are not necessarily positional biased. Thus, other data augmentation methods should be investigated in order to increase the generalization abilities of the models. Shorten et al. [82] introduced a wide variation of different data augmentation methods. The article highlighted a generative modeling framework as a popular approach for creating more training data in the case of medical images. The aim of a Generative Adversarial Network (GAN) is to create artificial instances from a dataset such that they obtain similar characteristics as the original dataset. In this way the size of the original dataset is increased with images of similar characteristics. GANs are therefore another data augmentation option to consider in future work when using medical images.

Furthermore, the model architecture is another aspect which could be investigated in future work. In 2017, Men et al. [18] showed how a deep dilated CNN-based method can be used to segment the tumor and organs at risk accurately and efficiently. The study used CT images from 278 patients with rectal cancer for evaluation. In addition, the performance of the deep dilated CNN was compared with the performance of U-Net. Consequently, the study found that the deep dilated CNN outperformed the U-Net for all segmentations, and the average DSC of the deep dilated CNN was 3.8% higher than for the U-Net. It should be noted that the study used CT images instead of MR images. However, the result indicates that other model architectures should be considered in future work.

Finally, it should be noted that the model parameters chosen in this thesis was based on the overall performance for all datasets. However, a higher model performance might have been achieved if the parameters were chosen specifically for each dataset. Thus, in future experiments one should consider the option of choosing model parameters individually for each dataset. Several other strategies for increasing the generalization abilities are also possible

to implement, such as dropout regularization and transfer learning. In the following subsection transfer learning is discussed as a possible implementation.

5.7.2 *Transfer Learning*

The idea of transfer learning is to utilize knowledge from one task to solve other related tasks better and faster [93]. Transfer learning was suggested as a way of dealing with limited amount of data available, which is one of the main reasons for poor model performance in the case of DL [20, 21, 47]. Therefore, several articles which performed classification or segmentation on medical images utilized transfer learning. Jonas Wacker et al. [94] used transfer learning for brain tumor segmentation and was able to show that despite the differences between the ImageNet dataset and the MR images used in their work, a considerable performance gain was achieved while stabilizing the training convergence. They were also able to show that a pre-trained network lead to more robust predictions, especially when the test data was different from the training data. Transfer learning has also been implemented for brain tumor classification [95] and for predicting lung cancer [96]. In both cases transfer learning resulted in an increased performance.

Other studies have argued that transfer learning is not always beneficial, and not as efficient as expected. In 2019 Google Brain published an article in collaboration with Cornell University [97] where they found that transfer learning only offers a limited performance gain. They were able to show that the benefits of transfer learning in the small dataset regime are largely due to architecture size. Hence they stated that transfer learning primarily helps the large models, which are designed to be trained with a million examples, while smaller models showed little difference between transfer learning and random initialization. It should be noted that the article was looking at a dataset size of 5000 images which consisted of fundus photographs and X-ray images. In many medical imaging cases the dataset size is less than 5000 images, and can be as small as 1000 images. Hence even though transfer learning did not show as much performance gain for the dataset of 5000 images, it might still be beneficial for even smaller datasets. In addition the images used in the article differs from MR images, and therefore the result might not be as representative for MR images.

Even though it is uncertain how beneficial the transfer learning would be, it is still a very interesting method to investigate further. Several approaches could be used when looking into transfer learning. One approach could be to train a model on the OxyTarget data and utilize the knowledge learned to predict LARC-RRP data. Another interesting approach is to train a model on a different cancer type, and use this knowledge to predict rectal cancer segmentation. A third option would be to use the more traditional approach where the model is trained on a large dataset, such as natural images from the ImageNet dataset, and use this knowledge to predict cancer tumors in

MR images. All of these approaches are interesting options to investigate in future work.

5.7.3 *The Input Images*

Another aspect which should be investigated into more detail is the standardization of the input images. The input images is an important factor which highly influences the output performance. Intensity inhomogeneities in the MR images can negatively impact any analysis done on the images, thus causing a bad model performance. In 1998 Sled et al. [98] proposed an approach for correcting intensity nonuniformity in MR data, known as the Nonparametric Nonuniform Intensity Normalization (N3) algorithm. The method has established itself as the standard approach in the field. However, an improvement of the approach was suggested by Tustison et al. [99] in 2010. The developed algorithm was called the N4ITK algorithm, and has shown promising result for correcting intensity inhomogeneities in MR images. The software is publicly available for use through the Insight Toolkit of the National Institutes of Health [100]. Thus, the approach is easy to implement and should be investigated further.

In addition, the use of DWIs shows promising results. Hence, DWIs should be included as additional input for the LARC-RRP dataset and Combined dataset as well. Furthermore, it would be interesting to test different b-values in order to analyse how the values influence the model performance.

5.7.4 *Additional Performance Metrics*

As pointed out by Reinke et al. [87] the DSC metric is not sufficient for detection tasks. Thus, in later experiments the model should only include tumor slices as input. Further work could therefore consist of creating an algorithm which selects the tumor slices from a patient. In this way one could create a system where the first algorithm chooses the tumor slices while the second algorithm segments the tumor in the selected slices.

Still, another performance metric should be included when evaluating the segmentation made by the model, as suggested by Reinke et al. [87]. The DSC estimates the overlap between two objects, and one should therefore include another performance metric which investigates the distances between the predicted region and the ground truth region. As previously suggested in Section 5.3.4 the MSD [88] and HD [87] metrics are two possible options.

5.7.5 *The Black Box Phenomena*

Finally, one of the main problems when implementing DL models for a clinical setting is the *Black Box* phenomena. The phenomena describes a system where the operations that occur are not visible to the user [47]. Thus, it is

difficult to interpret what exactly happens within the system. Consequently, there is a big issue with trusting the output made by the DL model. In order to mitigate this problem future work could include a comparison study of the manual delineation made by a radiologist and the predicted delineation made by the model. Several radiologists could then be invited to participate, where the aim is to distinguish which delineation was done by the radiologist and which was done by the model. This kind of study could potentially demonstrate the difficulty of separating the two delineations, and consequently mitigate the hesitation caused by the Black Box phenomena.

6 CONCLUSION

The thesis explored whether a deep CNN can be used to automatically segment rectal tumors based on MR images from two independent patient cohorts. T2w images of 89 patients from the LARC-RRP study, and 110 patients from the OxyTarget study were used as input. In addition, DWIs from 109 patients in the OxyTarget dataset, with a b-value equal to $500 \frac{s}{mm^2}$, were used as input. The data was divided into training, validation and test sets, and the manual delineations made by radiologists were used as ground truth. Several DL models were trained and varied in terms of image input, standardization method, loss function, learning rate and data augmentation.

The best performing model was trained on the OxyTarget dataset when solely using T2w images which contained tumor as input. The model used a learning rate of $1e-04$, the Best Combination (BC) as data augmentation method, the z-score normalization combined with matching of histograms (MH + Z-Score) as standardization method and the Modified Dice loss as loss function. The DL model outperformed the SML approach, but performed inferior to the interobserver variation. In addition, the contribution from each individual patient cohort increased when the model was trained on a combination of the LARC-RRP patients and OxyTarget patients.

The results demonstrate the possible benefit of training a DL model by combining two independent patient cohorts as input. Still, the model needs further improvement before it can be fully implemented in a clinical setting. Several modifications can be implemented in an attempt to improve the model performance such as including multiple MR sequences as input, as well as using transfer learning, various standardization methods, data augmentation methods. The thesis did however suggest to implement the model on a per image slice basis, combined with a suitable threshold value used for approving the predicted delineation. This would still increase the efficiency in the delineation process as we know it today. Thus, the DL models with U-Net architecture shows promising results for rectal cancer segmentation, and should be further explored.

BIBLIOGRAPHY

[1] W. H. Organization, *Cancer*, [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/cancer (visited on 05/05/2021).

[2] C. R. UK, *Worldwide cancer statistics | cancer research uk*, [Online]. Available: https://www.cancerresearchuk.org/health-professional/cancer-statistics/worldwide-cancer (visited on 12/05/2020).

[3] Cancer Registry of Norway, "Cancer in Norway 2019 - Cancer incidence, mortality, survival and prevalence in Norway," Norway. Oslo: Cancer Registry of Norway, Tech. Rep., 2020.

[4] Helsedirektoratet, *Nasjonalt handlingsprogram med retningslinjer for diagnostikk, behandling og oppfølging av kreft i tykktarm og endetarm*, 8th. Norway. Oslo: Helsedirektoratet, Dec. 2020.

[5] I. Dregely, D. Prezzi, C. Kelly-Morland, E. Roccia, R. Neji, and V. Goh, "Imaging biomarkers in oncology: Basics and application to MRI," *Journal of Magnetic Resonance Imaging*, vol. 48, no. 1, pp. 13–26, Jul. 2018, DOI: 10.1002/jmri.26058.

[6] C. Njeh, "Tumor delineation: The weakest link in the search for accuracy in radiotherapy," *Journal of Medical Physics*, vol. 33, no. 4, p. 136, Oct. 2008, DOI: 10.4103/0971-6203.44472.

[7] F. Chollet, *Deep Learning with Python*. Manning Publications, 2017.

[8] K. Suzuki, "Overview of deep learning in medical imaging," *Radiological Physics and Technology*, vol. 10, no. 3, pp. 257–273, Sep. 2017, DOI: 10.1007/s12194-017-0406-5.

[9] D. Yan, F. Vicini, J. Wong, and A. Martinez, "Adaptive radiation therapy," *Physics in Medicine and Biology*, vol. 42, no. 1, pp. 123–132, Jan. 1997, DOI: 10.1088/0031-9155/42/1/008.

[10] Z. Akkus, A. Galimzianova, A. Hoogi, D. L. Rubin, and B. J. Erickson, "Deep Learning for Brain MRI Segmentation: State of the Art and Future Directions," *Journal of Digital Imaging*, vol. 30, no. 4, pp. 449–459, Aug. 2017, DOI: 10.1007/s10278-017-9983-4.

[11] J. R. Burt, N. Torosdagli, N. Khosravan, H. RaviPrakash, A. Mortazi, F. Tissavirasingham, S. Hussein, and U. Bagci, "Deep learning beyond cats and dogs: recent advances in diagnosing breast cancer with deep neural networks," *The British Journal of Radiology*, vol. 91, no. 1089, p. 20 170 545, Apr. 2018, DOI: 10.1259/bjr.20170545.

[12] R. K. Samala, H.-P. Chan, L. M. Hadjiiski, M. A. Helvie, K. H. Cha, and C. D. Richter, "Multi-task transfer learning deep convolutional neural network: application to computer-aided diagnosis of breast cancer on mammograms," *Physics in Medicine & Biology*, vol. 62, no. 23, pp. 8894–8908, Nov. 2017, DOI: 10.1088/1361-6560/aa93d4.

[13] J. Lao, Y. Chen, Z.-C. Li, Q. Li, J. Zhang, J. Liu, and G. Zhai, "A Deep Learning-Based Radiomics Model for Prediction of Survival in Glioblastoma Multiforme," *Scientific Reports*, vol. 7, no. 1, p. 10 353, Dec. 2017, DOI: 10.1038/s41598-017-10649-8.

[14] L. Oakden-Rayner, G. Carneiro, T. Bessen, J. C. Nascimento, A. P. Bradley, and L. J. Palmer, "Precision Radiology: Predicting longevity using feature engineering and deep learning methods in a radiomics framework," *Scientific Reports*, vol. 7, no. 1, p. 1648, Dec. 2017, DOI: 10.1038/s41598-017-01931-w.

[15] M. A. Gambacorta, L. Boldrini, C. Valentini, *et al.*, "Automatic segmentation software in locally advanced rectal cancer: READY (REsearch program in Auto Delineation sYstem)-RECTAL 02: prospective study," *Oncotarget*, vol. 7, no. 27, pp. 42 579–42 584, Jul. 2106, DOI: 10.18632/oncotarget.9938.

[16] X. Xia, J. Wang, Y. Li, J. Peng, J. Fan, J. Zhang, J. Wan, Y. Fang, Z. Zhang, and W. Hu, "An Artificial Intelligence-Based Full-Process Solution for Radiotherapy: A Proof of Concept Study on Rectal Cancer," *Frontiers in Oncology*, vol. 10, Feb. 2021, DOI: 10.3389/fonc.2020.616721.

[17] J. Lee, J. E. Oh, M. J. Kim, B. Y. Hur, and D. K. Sohn, "Reducing the Model Variance of a Rectal Cancer Segmentation Network," *IEEE Access*, vol. 7, pp. 182 725–182 733, 2019, DOI: 10.1109/ACCESS.2019.2960371.

[18] K. Men, J. Dai, and Y. Li, "Automatic segmentation of the clinical target volume and organs at risk in the planning CT for rectal cancer using deep dilated convolutional neural networks," *Medical Physics*, vol. 44, no. 12, pp. 6377–6389, Dec. 2017, DOI: 10.1002/mp.12602.

[19] S. Trebeschi, J. J. M. van Griethuysen, D. M. J. Lambregts, M. J. Lahaye, C. Parmar, F. C. H. Bakers, N. H. G. M. Peters, R. G. H. Beets-Tan, and H. J. W. L. Aerts, "Deep Learning for Fully-Automated Localization and Segmentation of Rectal Cancer on Multiparametric MR," *Scientific Reports*, vol. 7, no. 1, p. 5301, Dec. 2017, DOI: 10.1038/s41598-017-05728-9.

[20] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, "Revisiting Unreasonable Effectiveness of Data in Deep Learning Era," in *2017 IEEE International Conference on Computer Vision (ICCV)*, vol. 2017-October, IEEE, Oct. 2017, pp. 843–852, DOI: 10.1109/ICCV.2017.97.

[21] A. Halevy, P. Norvig, and F. Pereira, "The Unreasonable Effectiveness of Data," *IEEE Intelligent Systems*, vol. 24, no. 2, pp. 8–12, Mar. 2009, DOI: 10.1109/MIS.2009.36.

[22] P. E. G. Live Eikenes Neil Peter Jerome, *FY8408 - MR imaging*. Department of Physics, NTNU, 2020.

[23] C. Westbrook and C. Kaut, *MRI in Practice*, 2nd. Wiley-Blackwell, 1998.

[24] R.-J. M. van Geuns, P. A. Wielopolski, H. G. de Bruin, B. J. Rensing, P. M. van Ooijen, M. Hulshoff, M. Oudkerk, and P. J. de Feyter, "Basic principles of magnetic resonance imaging," *Progress in Cardiovascular Diseases*, vol. 42, no. 2, pp. 149–156, Sep. 1999, DOI: 10.1016/S0033-0620(99)70014-9.

[25] D. B. Plewes and W. Kucharczyk, "Physics of MRI: A primer," *Journal of Magnetic Resonance Imaging*, vol. 35, no. 5, pp. 1038–1054, May 2012, DOI: 10.1002/jmri.23642.

[26] P. Hagmann, L. Jonasson, P. Maeder, J.-P. Thiran, V. J. Wedeen, and R. Meuli, "Understanding Diffusion MR Imaging Techniques: From Scalar Diffusion-weighted Imaging to Diffusion Tensor Imaging and Beyond," *RadioGraphics*, vol. 26, no. suppl_1, S205–S223, Oct. 2006, DOI: 10.1148/rg.26si065510.

[27] V. Baliyan, C. J. Das, R. Sharma, and A. K. Gupta, "Diffusion weighted imaging: Technique and applications," *World Journal of Radiology*, vol. 8, no. 9, p. 785, 2016, DOI: 10.4329/wjr.v8.i9.785.

[28] E. O. Stejskal and J. E. Tanner, "Spin Diffusion Measurements: Spin Echoes in the Presence of a Time-Dependent Field Gradient," *The Journal of Chemical Physics*, vol. 42, no. 1, pp. 288–292, Jan. 1965, DOI: 10.1063/1.1695690.

[29] S. Mori and P. B. Barker, "Diffusion magnetic resonance imaging: Its principle and applications," *The Anatomical Record*, vol. 257, no. 3, pp. 102–109, Jun. 1999, DOI: 10.1002/(SICI)1097-0185(19990615)257:3<102::AID-AR7>3.0.CO;2-6.

[30] P. A. Rinck, *Magnetic Resonance in Medicine*, Electronic version 11. European Magnetic Resonance Forum, 2017, [Online]. Available: https://www.magnetic-resonance.org (visited on 12/03/2021).

[31] L. G. Nyúl and J. K. Udupa, "On standardizing the MR image intensity scale," *Magnetic Resonance in Medicine*, vol. 42, no. 6, pp. 1072–1081, Dec. 1999, DOI: 10.1002/(SICI)1522-2594(199912)42:6<1072::AID-MRM11>3.0.CO;2-M.

[32] T. M. Mitchell, *Machine Learning*. McGraw-Hill Science/Engineering/Math, 1997.

[33] F. Balali, J. Nouri, A. Nasiri, and T. Zhao, "Machine Learning Principles," in *Data Intensive Industrial Asset Management*, Cham: Springer International Publishing, 2020, pp. 115–157, DOI: 10.1007/978-3-030-35930-0{_}8.

[34] S.-S. Shai and B.-D. Shai, *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

[35] S. Wang and R. M. Summers, "Machine learning and radiology," *Medical Image Analysis*, vol. 16, no. 5, pp. 933–951, Jul. 2012, DOI: 10.1016/j.media.2012.02.005.

[36] B. Ghojogh and M. Crowley, *Linear and quadratic discriminant analysis: Tutorial*, 2019, arXiv: 1906.02590 [stat.ML], [Online]. Available: https://arxiv.org/abs/1906.02590.

[37] W. S. Noble, "What is a support vector machine?" *Nature Biotechnology*, vol. 24, no. 12, pp. 1565–1567, Dec. 2006, DOI: 10.1038/nbt1206-1565.

[38] A. Widodo and B.-S. Yang, "Support vector machine in machine condition monitoring and fault diagnosis," *Mechanical Systems and Signal Processing*, vol. 21, no. 6, pp. 2560–2574, Aug. 2007, DOI: 10.1016/j.ymssp.2006.12.007.

[39] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, [Online]. Available: http://www.deeplearningbook.org.

[40] A. Shrestha and A. Mahmood, "Review of Deep Learning Algorithms and Architectures," *IEEE Access*, vol. 7, pp. 53 040–53 065, 2019, DOI: 10.1109/ACCESS.2019.2912200.

[41] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep Learning for Computer Vision: A Brief Review," *Computational Intelligence and Neuroscience*, vol. 2018, pp. 1–13, 2018, DOI: 10.1155/2018/7068349.

[42] S. Sharma, S. Sharma, and A. Athaiya, "ACTIVATION FUNCTIONS IN NEURAL NETWORKS," *International Journal of Engineering Applied Sciences and Technology*, vol. 4, no. 12, pp. 310–316, 2020, [Online]. Available: http://www.ijeast.com.

[43] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference and Prediction*, 2nd. Springer New York, 2009.

[44] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation," in *2016 Fourth International Conference on 3D Vision (3DV)*, IEEE, Oct. 2016, pp. 565–571, DOI: 10.1109/3DV.2016.79.

[45] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, 2017, arXiv: 1412.6980 [cs.LG], [Online]. Available: https://arxiv.org/abs/1412.6980.

[46] A. Jain, K. Nandakumar, and A. Ross, "Score normalization in multimodal biometric systems," *Pattern Recognition*, vol. 38, no. 12, pp. 2270–2285, Dec. 2005, DOI: 10.1016/j.patcog.2005.01.012.

[47] I. Castiglioni, L. Rundo, M. Codari, G. Di Leo, C. Salvatore, M. Interlenghi, F. Gallivanone, A. Cozzi, N. C. D'Amico, and F. Sardanelli, "AI applications to medical images: From machine learning to deep learning," *Physica Medica*, vol. 83, pp. 9–24, Mar. 2021, DOI: 10.1016/j.ejmp.2021.02.006.

[48] A. Kulkarni, D. Chong, and F. A. Batarseh, "Foundations of data imbalance and solutions for a data democracy," in *Data Democracy*, Elsevier, Jan. 2020, pp. 83–106, DOI: 10.1016/B978-0-12-818366-3.00005-8.

[49] S. Raschka and V. Mirjalili, *Python Machine Learning*, 2nd. Packt Publishing, 2015.

[50] M.-P. Hosseini, M. R. Nazem-Zadeh, D. Pompili, and H. Soltanian-Zadeh, "Statistical validation of automatic methods for hippocampus segmentation in MR images of epileptic patients," in *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, IEEE, Aug. 2014, pp. 4707–4710, DOI: 10.1109/EMBC.2014.6944675.

[51] K. H. Zou, S. K. Warfield, A. Bharatha, C. M. Tempany, M. R. Kaus, S. J. Haker, W. M. Wells, F. A. Jolesz, and R. Kikinis, "Statistical validation of image segmentation quality based on a spatial overlap index1," *Academic Radiology*, vol. 11, no. 2, pp. 178–189, Feb. 2004, DOI: 10.1016/S1076-6332(03)00671-8.

[52] A. Tharwat, "Classification assessment methods," *Applied Computing and Informatics*, vol. 17, no. 1, pp. 168–192, Jan. 2021, DOI: 10.1016/j.aci.2018.08.003.

[53] 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, May 2019, DOI: 10.1016/j.zemedi.2018.11.002.

[54] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*, vol. 2018-Janua, IEEE, Aug. 2017, pp. 1–6, DOI: 10.1109/ICEngTechnol.2017.8308186.

[55] W. Weng and X. Zhu, "INet: Convolutional Networks for Biomedical Image Segmentation," *IEEE Access*, vol. 9, pp. 16 591–16 603, 2021, DOI: 10.1109/ACCESS.2021.3053408.

[56] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9351, Springer Verlag, 2015, pp. 234–241, DOI: 10.1007/978-3-319-24574-4{_}28.

[57] *The LARC-RRP study - Acredit*, [Online]. Available: https://www.acredit.no/the-larc_rrp-study/ (visited on 10/25/2020).

[58] T Seierstad, K. H. Hole, K. K. Grøholt, S Dueland, A. H. Ree, K Flatmark, and K. R. Redalen, "MRI volumetry for prediction of tumour response to neoadjuvant chemotherapy followed by chemoradiotherapy in locally advanced rectal cancer," *The British Journal of Radiology*, vol. 88, no. 1051, p. 20 150 097, Jul. 2015, DOI: 10.1259/bjr.20150097.

[59] *The OxyTarget study – Acredit*, [Online]. Available: https://www.acredit.no/the-oxytarget-study/ (visited on 10/26/2020).

[60] K. M. Bakke, S. Meltzer, E. Grøvik, A. Negård, S. H. Holmedal, K.-I. Gjesdal, A. Bjørnerud, A. H. Ree, and K. R. Redalen, "Sex Differences and Tumor Blood Flow from Dynamic Susceptibility Contrast MRI Are Associated with Treatment Response after Chemoradiation and Long-term Survival in Rectal Cancer," *Radiology*, vol. 297, no. 2, pp. 352–360, Nov. 2020, DOI: 10.1148/radiol.2020200287.

[61] C. E. Kahn, J. A. Carrino, M. J. Flynn, D. J. Peck, and S. C. Horii, "DICOM and Radiology: Past, Present, and Future," *Journal of the American College of Radiology*, vol. 4, no. 9, pp. 652–657, Sep. 2007, DOI: 10.1016/j.jacr.2007.06.004.

[62] *How to use DicomCleaner™*, [Online]. Available: http://www.dclunie.com/pixelmed/software/webstart/DicomCleanerUsage.html (visited on 11/22/2020).

[63] Ingvild Askim Adde, *Automatic tumor segmentation of rectal cancer in MR images*, 2020.

[64] M. Larobina and L. Murino, "Medical Image File Formats," *Journal of Digital Imaging*, vol. 27, no. 2, pp. 200–206, Apr. 2014, DOI: 10.1007/s10278-013-9657-9.

[65] *Colorectal Cancer: Stages*, [Online]. Available: https://www.cancer.net/cancer-types/colorectal-cancer/stages (visited on 02/22/2021).

[66] A. Collette, *Python and HDF5: Unlocking Scientific Data*. O'Reilly Media, 2013.

[67] Eline Furu Skjelbred, *Tumor segmentation by deep learning*, 2020.

[68] *deoxys — deoxys 0.0.8 documentation*, [Online]. Available: https://deoxys.readthedocs.io/en/latest/readme.html#features (visited on 03/29/2021).

[69] *NMBU Orion Compute Cluster — NMBU Orion user support documentation*, [Online]. Available: https://nmbu-orion-support.readthedocs.io/en/latest/index.html (visited on 03/29/2021).

[70] B. W. Silverman, *Density Estimation for Statistics and Data Analysis Chapter 1 and 2*. Chapman and Hall, 1986.

[71] J. L. Hintze and R. D. Nelson, "Violin Plots: A Box Plot-Density Trace Synergism," *The American Statistician*, vol. 52, no. 2, pp. 181–184, 1998, DOI: 10.2307/2685478.

[72] M. Själander, M. Jahre, G. Tufte, and N. Reissmann, *EPIC: An energy-efficient, high-performance GPGPU computing research infrastructure*, 2019, arXiv: 1912.05848 [cs.DC], [Online]. Available: http://arxiv.org/abs/1912.05848.

[73] H. Ng, S. Huang, S. Ong, K. Foong, P. Goh, and W. Nowinski, "Medical image segmentation using watershed segmentation with texture-based region merging," in *2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, IEEE, Aug. 2008, pp. 4039–4042, DOI: 10.1109/IEMBS.2008.4650096.

[74] Y. Song, Y.-D. Zhang, X. Yan, H. Liu, M. Zhou, B. Hu, and G. Yang, "Computer-aided diagnosis of prostate cancer using a deep convolutional neural network from multiparametric MRI," *Journal of Magnetic Resonance Imaging*, vol. 48, no. 6, pp. 1570–1577, Dec. 2018, DOI: 10.1002/jmri.26047.

[75] Y. Chen, L. Xing, L. Yu, H. P. Bagshaw, M. K. Buyyounouski, and B. Han, "Automatic intraprostatic lesion segmentation in multiparametric magnetic resonance images with proposed multiple branch UNet," *Medical Physics*, vol. 47, no. 12, pp. 6421–6429, Dec. 2020, DOI: 10.1002/mp.14517.

[76] B. LeBaron and A. S. Weigend, "A bootstrap evaluation of the effect of data splitting on financial time series," *IEEE Transactions on Neural Networks*, vol. 9, no. 1, pp. 213–220, 1998.

[77] R. May, H. Maier, and G. Dandy, "Data splitting for artificial neural networks using SOM-based stratified sampling," *Neural Networks*, vol. 23, no. 2, pp. 283–294, Mar. 2010, DOI: 10.1016/j.neunet.2009.11.009.

[78] A. H. Abdi, C. Luong, T. Tsang, *et al.*, "Automatic Quality Assessment of Echocardiograms Using Convolutional Neural Networks: Feasibility on the Apical Four-Chamber View," *IEEE Transactions on Medical Imaging*, vol. 36, no. 6, pp. 1221–1230, Jun. 2017, DOI: 10.1109/TMI.2017.2690836.

[79] B. Belaroussi, J. Milles, S. Carme, Y. M. Zhu, and H. Benoit-Cattin, "Intensity non-uniformity correction in MRI: Existing methods and their validation," *Medical Image Analysis*, vol. 10, no. 2, pp. 234–246, Apr. 2006, DOI: 10.1016/j.media.2005.09.004.

[80] 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, Dec. 2017, DOI: 10.1016/j.media.2017.07.005.

[81] M. I. Razzak, S. Naz, and A. Zaib, "Deep Learning for Medical Image Processing: Overview, Challenges and the Future," in *Lecture Notes in Computational Vision and Biomechanics*, vol. 26, Springer Netherlands, 2018, pp. 323–350, DOI: 10.1007/978-3-319-65981-7{_}12.

[82] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," *Journal of Big Data*, vol. 6, no. 1, p. 60, Dec. 2019, DOI: 10.1186/s40537-019-0197-0.

[83] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," in *25th annual conference on neural information processing systems (NIPS 2011)*, Neural Information Processing Systems Foundation, vol. 24, 2011.

[84] P. Franco, F. Arcadipane, E. Trino, E. Gallio, S. Martini, G. C. Iorio, C. Piva, F. Moretto, M. G. R. Redda, R. Verna, *et al.*, "Variability of clinical target volume delineation for rectal cancer patients planned for neoadjuvant radiotherapy with the aid of the platform anatom-e," *Clinical and translational radiation oncology*, vol. 11, pp. 33–39, 2018.

[85] J. Wang, J. Lu, G. Qin, L. Shen, Y. Sun, H. Ying, Z. Zhang, and W. Hu, "Technical Note: A deep learning-based autosegmentation of rectal tumors in MR images," *Medical Physics*, vol. 45, no. 6, pp. 2560–2564, Jun. 2018, DOI: 10.1002/mp.12918.

[86] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, DOI: 10.1038/nature14539.

[87] A. Reinke, M. Eisenmann, M. D. Tizabi, *et al.*, *Common Limitations of Image Processing Metrics: A Picture Story*, Apr. 2021, arXiv: 2104.05642 [cs.DC], [Online]. Available: http://arxiv.org/abs/2104.05642.

[88] *Surface Distance - an overview | ScienceDirect Topics*, [Online]. Available: https://www.sciencedirect.com/topics/engineering/surface-distance (visited on 06/05/2021).

[89] L. Hou, Y. Cheng, N. Shazeer, N. Parmar, Y. Li, P. Korfiatis, T. M. Drucker, D. J. Blezek, and X. Song, *High Resolution Medical Image Analysis with Spatial Partitioning*, Sep. 2019, arXiv: 1909.03108v3 [eess.IV], [Online]. Available: https://arxiv.org/abs/1909.03108.

[90] V. Kumar, Y. Gu, S. Basu, *et al.*, "Radiomics: the process and the challenges," *Magnetic Resonance Imaging*, vol. 30, no. 9, pp. 1234–1248, Nov. 2012, DOI: 10.1016/j.mri.2012.06.010.

[91] S. Rizzo, F. Botta, S. Raimondi, D. Origgi, C. Fanciullo, A. G. Morganti, and M. Bellomi, "Radiomics: the facts and the challenges of image analysis," *European Radiology Experimental*, vol. 2, no. 1, p. 36, Dec. 2018, DOI: 10.1186/s41747-018-0068-z.

[92] F. C. Soon, H. Y. Khaw, J. H. Chuah, and J. Kanesan, "Hyper-parameters optimisation of deep CNN architecture for vehicle logo recognition," *IET Intelligent Transport Systems*, vol. 12, no. 8, pp. 939–946, Oct. 2018, DOI: 10.1049/iet-its.2018.5127.

[93] J. Lu, V. Behbood, P. Hao, H. Zuo, S. Xue, and G. Zhang, "Transfer learning using computational intelligence: A survey," *Knowledge-Based Systems*, vol. 80, pp. 14–23, May 2015, DOI: 10.1016/j.knosys.2015.01.010.

[94] J. Wacker, M. Ladeira, and J. E. V. Nascimento, "Transfer Learning for Brain Tumor Segmentation," in, Springer, Cham, Oct. 2021, pp. 241–251, DOI: 10.1007/978-3-030-72084-1{_}22.

[95] R. Chelghoum, A. Ikhlef, A. Hameurlaine, and S. Jacquir, "Transfer Learning Using Convolutional Neural Network Architectures for Brain Tumor Classification from MRI Images," in *IFIP Advances in Information and Communication Technology*, vol. 583 IFIP, Springer, Jun. 2020, pp. 189–200, DOI: 10.1007/978-3-030-49161-1{_}17.

[96] Y. Xu, A. Hosny, R. Zeleznik, C. Parmar, T. Coroller, I. Franco, R. H. Mak, and H. J. Aerts, "Deep Learning Predicts Lung Cancer Treatment Response from Serial Medical Imaging," *Clinical Cancer Research*, vol. 25, no. 11, pp. 3266–3275, Jun. 2019, DOI: 10.1158/1078-0432.CCR-18-2495.

[97] M. Raghu, C. Zhang, J. Kleinberg, and S. Bengio, "Transfusion: Understanding Transfer Learning for Medical Imaging," no. NeurIPS, Feb. 2019, [Online]. Available: http://arxiv.org/abs/1902.07208.

[98] J. Sled, A. Zijdenbos, and A. Evans, "A nonparametric method for automatic correction of intensity nonuniformity in MRI data," *IEEE Transactions on Medical Imaging*, vol. 17, no. 1, pp. 87–97, 1998, DOI: 10.1109/42.668698.

[99] N. J. Tustison, B. B. Avants, P. A. Cook, Yuanjie Zheng, A. Egan, P. A. Yushkevich, and J. C. Gee, "N4ITK: Improved N3 Bias Correction," *IEEE Transactions on Medical Imaging*, vol. 29, no. 6, pp. 1310–1320, Jun. 2010, DOI: 10.1109/TMI.2010.2046908.

[100] L. Ibáñez, I. Ibá, I. Ibáñez, W. Schroeder, L. Ng, and J. Cates, *The ITK Software Guide Second Edition Updated for ITK version 2.4*, 2005, [Online]. Available: http://www.itk.org.

A SPLITTING OF DATASETS

A.1 TRADITIONAL SPLIT OF OXYTARGET DATA

	PERCENTAGE OF PATIENTS	NUMBER OF PATIENTS	NUMBER OF IMAGE SLICES
Train	70 %	77	1973
Validation	15 %	16	397
Test	15 %	17	439
Total	100 %	110	2809

A.2 TRADITIONAL SPLIT OF LARC-RRP DATA

	PERCENTAGE OF PATIENTS	NUMBER OF PATIENTS	NUMBER OF IMAGE SLICES
Train	70 %	63	2262
Validation	15 %	13	423
Test	15 %	13	448
Total	100 %	89	3133

A.3 5-FOLD CROSS VALIDATION SPLIT OF OXYTARGET DATA

	PERCENTAGE OF PATIENTS	NUMBER OF PATIENTS	NUMBER OF IMAGE SLICES
Fold 1	20 %	19	485
Fold 2	20 %	19	507
Fold 3	20 %	19	477
Fold 4	20 %	18	456
Fold 5	20 %	18	445
Total	100 %	93	2370

A.4 5-FOLD CROSS VALIDATION SPLIT OF LARC-RRP DATA

	PERCENTAGE OF PATIENTS	NUMBER OF PATIENTS	NUMBER OF IMAGE SLICES
Fold 1	20 %	15	503
Fold 2	20 %	15	559
Fold 3	20 %	15	548
Fold 4	20 %	15	499
Fold 5	20 %	16	576
Total	100 %	76	2685

HDF5 FILE NUMBER	DATASET	IMAGE MODALITY	IMAGE SLICES	SPLIT METHOD	STANDARDIZATION
1	OxyTarget	T2w	All	K-Fold	No
2	OxyTarget	T2w	All	Traditional	No
3	OxyTarget	T2w	All	Traditional	Z-Score
4	OxyTarget	T2w	All	Traditional	MH
5	OxyTarget	T2w	All	Traditional	MH + Z-Score
6	OxyTarget	T2w	Tumor	Traditional	No
7	OxyTarget	T2w	Tumor	Traditional	Z-Score
8	OxyTarget	T2w	Tumor	Traditional	MH
9	OxyTarget	T2w	Tumor	Traditional	MH + Z-Score
10	OxyTarget	T2w + DWI	Tumor	Traditional	No
11	OxyTarget	T2w + DWI	Tumor	Traditional	Z-Score
12	OxyTarget	T2w + DWI	Tumor	Traditional	MH
13	OxyTarget	T2w + DWI	Tumor	Traditional	MH + Z-Score
14	LARC-RRP	T2w	All	K-Fold	No
15	LARC-RRP	T2w	All	Traditional	No
16	LARC-RRP	T2w	All	Traditional	Z-Score
17	LARC-RRP	T2w	All	Traditional	MH
18	LARC-RRP	T2w	All	Traditional	MH + Z-Score

HDF5 FILES

HDF5 FILE NUMBER NUMBER	DATASET	IMAGE MODALITY	IMAGE SLICES	SPLIT METHOD	STANDARDIZATION
19	LARC-RRP	T2W	Tumor	Traditional	No
20	LARC-RRP	T2W	Tumor	Traditional	Z-Score
21	LARC-RRP	T2W	Tumor	Traditional	MH
22	LARC-RRP	T2W	Tumor	Traditional	MH + Z-Score
23	Combined	T2W	All	K-Fold	No
24	Combined	T2W	All	Traditional	No
25	Combined	T2W	All	Traditional	Z-Score
26	Combined	T2W	All	Traditional	MH
27	Combined	T2W	All	Traditional	MH + Z-Score
28	Combined	T2W	Tumor	Traditional	No
29	Combined	T2W	Tumor	Traditional	Z-Score
30	Combined	T2W	Tumor	Traditional	MH
31	Combined	T2W	Tumor	Traditional	MH + Z-Score

C CODE

C.1 DEFAULT AUGMENTATION CONFIGURATION

Developed by Ngoc Huynh Bao[2]

```
"augmentations": [{
                "class_name": "ImageAugmentation2D",
                "config": {
                  "rotation_range": 90,
                  "zoom_range": [
                    0.8,
                    1.2
                  ],
                  "shift_range": [
                    10,
                    10
                  ],
                  "flip_axis": 0,
                  "brightness_range": [
                    0.8,
                    1.2
                  ],
                  "contrast_range": [
                    0.7,
                    1.3
                  ],
                  "noise_variance": 0.05,
                  "blur_range": [
                    0.5,
                    1.5
                  ]
                }
              }]
```

C.2 BEST COMBINATION AUGMENTATION CONFIGURATION

Developed by Maria Ødegaard[3]

```
"augmentations": [{
                "class_name": "ImageAugmentation2D",
                "config": {
                  "rotation_range": 90,
                  "zoom_range": [
                    0.5,
                    1.5
                  ],
                  "shift_range": [
                    10,
                    10
                  ],
                  "flip_axis": 0,
                  "brightness_range": 1,
                  "contrast_range": 1,
                  "noise_variance": 0.05,
                  "blur_range": [
                    0.5,
                    1.5
                  ]
                }
              }]
```

D.1 ABSTRACT

Automatic tumor segmentation in rectal cancer by machine learning of MR images

Ingvild Askim Adde[1], Franziska Knuth[1], Aurora R. Grøndahl[2], Anne Negård[3], Sebastian Meltzer[4], Cecilia M. Futsæther[2], Kathrine Røe Redalen[1]

[1]Department of Physics, Norwegian University of Science and Technology
[2]Faculty of Science and Technology, Norwegian University of Life Sciences
[3]Department of Radiology, Akershus University Hospital
[4]Department of Oncology, Akershus University Hospital

Purpose: Manual tumor delineation is required for several purposes, such as calculation of quantitative image biomarkers and for target delineation in radiotherapy. The delineation process is a time-consuming task which is subject to intra- and interobserver variations. It would therefore be beneficial to develop a method which automatically segments the tumor and reduces the intra- and interobserver variations, as well as saves time for the radiologists and oncologists. The aim of this project was to evaluate several shallow machine learning models for their ability to perform automatic tumor segmentation of rectal cancer based on T2-weighted magnetic resonance (MR) images.

Materials: Two datasets with MRI of rectal cancer were used for training and testing of the machine learning models. Dataset 1 consisted of 89 patients, and dataset 2 of 110 patients. Manual delineations of the tumor volumes were made by experienced radiologists and used as ground truth.

Methods: In order to separate the tumor and the normal tissue in the MR images three different shallow machine learning models were evaluated as classification methods; linear discriminant analysis (LDA), quadratic discriminant analysis (QDA) and support vector machine (SVM). The properties that a voxel in the MR images represented was determined by using three different unfolding methods. First, each voxel was represented by its own intensity value (1D). Second, the intensity information of the closest neighbors in two dimensions (2D) was included. Third, the intensity information of the closest neighbors in three dimensions (3D) was included. The models were evaluated by combining these classification and unfolding methods. Dataset 1, dataset 2 and a combination of the dataset 1 and 2 were used as input for the models. The results were evaluated using the dice similarity coefficient (DSC) and the mean surface distance (MSD). The DSC was used to evaluate the overlap between the manual and automatic delineation, while the MSD was used to estimate the surface distance between the manual and automatic delineation.

Results: The best model performance was achieved with dataset 2 alone when QDA was used as classification method and 3D was used as unfolding method. Figure 1 shows an example of a tumor segmented with this approach. This combination gave a median DSC of 0.612 and a median MSD of 3.46 mm. The results for all models run with QDA are shown in Figure 2.

Conclusion: We evaluated whether shallow machine learning models can be used to automatically segment rectal tumors based on T2-weighted MR images. The performance of the models were not good enough to be considered for clinical use. A more advanced deep learning model of the MR images is currently being implemented and results from these analyses will be presented at the meeting.

E.1 ABSTRACT

MRI-based automatic segmentation of rectal cancer using 2D U-Net on two independent cohorts

Knuth F (1)*, Adde IA (1)*, Huynh BN (2), Grøndahl AR (2), Winter RM (1), Negård A (3), Holmedal SH (3), Meltzer S (4), Ree AH (4), Flatmark K (5), Dueland S (6), Hole KH (7), Seierstad T (7), Redalen KR (1), Futsæther CM (2); (*: equal contribution)

(1) Dept. of Physics, NTNU, Trondheim, Norway;
(2) Faculty of Science and Technology, NMBU, Ås, Norway;
(3) Dept. of Radiology, Akershus University Hospital, Lørenskog, Norway;
(4) Dept. of Oncology, Akershus University Hospital, Lørenskog, Norway;
(5) Dept. of Gastroenterological Surgery, Oslo University Hospital, Oslo, Norway;
(6) Dept. of Oncology, Oslo University Hospital, Oslo, Norway;
(7) Division of Radiology and Nuclear Medicine, Oslo University Hospital, Oslo, Norway;

Introduction

Tumor delineation is time- and labor-intensive and prone to inter- and intra-observer variations. Magnetic resonance imaging (MRI) provides good soft tissue contrast, and functional MRI captures tissue properties that may be valuable for tumor delineation. We explored MRI-based automatic segmentation of rectal cancer using a deep learning (DL) approach. We first investigated potential improvements when including both anatomical T2-weighted (T2w) MRI and diffusion-weighted MRI (DWI). Secondly, we investigated generalizability by including two independent cohorts.

Materials and methods

Two patient cohorts (A and B) from different hospitals with 110 and 89 patients, respectively, were subject to 1.5T MRI at baseline. T2w MRI was acquired from both cohorts and DWI from 109 patients in cohort A. Tumors were manually delineated by three radiologists (two in cohort A, one in cohort B). Both cohorts were split into train (70%), validation (15%), and test (15%) sets. A 2D U-net was trained on T2w and T2w+DWI (b-value of 500 s/mm^2) on individual (A, B) and combined (A+B) cohorts. Optimal parameters for image preprocessing and training were identified before the optimized models were evaluated on the validation sets of both cohorts. Median per patient Dice similarity coefficient (mDSC) was used as performance measure.

Results

For cohort A, the T2w model resulted in a mDSC of 0.80. Inclusion of DWI did not further improve the performance (mDSC: 0.80). The T2w MR-based model trained on A and tested on B achieved a mDSC of 0.53. This performance was lower compared to the model trained and validated on B (mDSC: 0.60). Training on the combined dataset resulted in an overall mDSC of 0.74 where patients from cohort A and B contributed 0.82 and 0.56, respectively.

Conclusion

T2w MR-based DL models demonstrated high performance for automatic tumor segmentation, at the same level as published data on interobserver variation. DWI did not improve results further. Using DL models on unseen cohorts require caution, and one cannot expect the same performance.

Ingvild Askim Adde

Master's Thesis

NTNU
Norwegian University of
Science and Technology