Eline Furu Skjelbred

Tumor segmentation by deep learning

Master's thesis in Applied Physics and Mathematics

Supervisor: Kathrine Røe Redalen

June 2020

Master's thesis

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

NTNU
Norwegian University of
Science and Technology

Eline Furu Skjelbred

Tumor segmentation by deep learning

Master's thesis in Applied Physics and Mathematics
Supervisor: Kathrine Røe Redalen
June 2020

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

NTNU
Norwegian University of
Science and Technology

Acknowledgments

First of all, I would like to thank my supervisor, Associate Professor Kathrine Røe Redalen, who has offered excellent guidance and feedback through this whole process. I am extremely grateful for the continued follow-up she has given me through online meetings and emails in the challenging time after the covid-19 outbreak. Without her support, this thesis would have been difficult to finish.

I also wish to thank Ph.D. student Franziska Knuth for being available for discussions and providing answers to my questions. In addition to this, she deserves a huge thanks or her work with the pre-processing of the image data.

The outbreak of covid-19 resulted in drastic changes for all of us, and as a consequence, the original plan for this master thesis had to be adjusted. I was no longer able to visit the Norwegian University of Life Sciences (NMBU) to train the deep learning models. Hence, I want to express my gratitude to Yngve Mardal Moe, head engineer at NMBU, for doing a tremendous job with the training of the models and providing me with the results. Without him, this would not have been possible.

Further, I would like to thank Professor Cecilia Marie Futsæther for making the collaboration with NMBU possible, and for sharing her knowledge. I must also thank postdoc. René Winter. His insightful input to discussions and great interest in the project has been appreciated.

Many thanks to my closest friends and fellow students. They have provided great support and given me the motivation to complete this thesis. It would have been difficult to get through this semester without their smiles and laughter.

Finally, I would like to thank my family for all the love and support they have given me.

Eline Furu Skjelbred
Trondheim, June 2020

Abstract

An important step in both quantitative image analysis and radiotherapy treatment planning is the delineation of the tumor volume. This is a time-consuming task, and it is also the greatest source of uncertainty due to interobserver variability. The purpose of this thesis was to explore a deep learning approach with convolutional neural networks for automatic segmentation of tumor volume based on MR images from patients with rectal cancer. This could potentially save time for the radiologists/oncologists and contribute to a more consistent delineation.

T2 weighted and diffusion weighted images with seven different b-values between $0\,\mathrm{s/mm^2}$ and $1300\,\mathrm{s/mm^2}$ from 81 patients from the OxyTarget study with rectal cancer patients were used. The image data was split into a training set (51 patients), a validation set (10 patients), and a test set (20 patients), stratified by gender and the tumor stage. A total of nine models with a U-net architecture were created and varied in terms of which image types that were used as input and which loss function that was used. The different loss functions that were tested were the cross entropy loss, the Dice loss, and a modified version of the Dice loss. Two radiologists had performed manual tumor delineations on the images, and the union of these two was used as the ground truth for the models. The models were evaluated based on the average Dice similarity coefficient (DSC) per patient in the validation set. The best U-net model was then compared to the results from a shallow machine learning approach based on the linear support vector classifier.

The performance for the different U-net models ranged from a DSC of 0.58 to a DSC of 0.67, and the best model took T2 weighted images as input and used the modified Dice loss function. Compared to the model with the linear support vector classifier, which resulted in a DSC equal to 0.48, the U-net models were superior. The DSC between the two manual delineations was calculated to 0.78, which indicates that the U-net model needs to be improved before it can be of clinical use.

However, the U-net model shows promising results for the automatic segmentation of the tumor volume. To improve the model performance, the effect of having input images with high resolution, and adding data augmentation and image cropping should be explored.

Sammendrag

Et viktig steg i både kvantitativ bildeanalyse og strålebehandling er inntegningen av kreftsvulstvolumet. Dette er en tidkrevende oppgave, og det er den største kilden til usikkerhet grunnet interobservatørvarabilitet. Hensikten med denne masteroppgaven var å undersøke om kunstig intelligens i form av dyp læring med konvolusjonelle nevrale nettverk kan benyttes for automatisk segmentering av kreftsvulsten basert på MR-bilder fra pasienter med endetarmskreft. Dette kan potensielt spare tid for radiologene/onkologene og bidra til en mer konsekvent inntegning.

T2-vektede og diffusjonsvektede bilder med syv ulike b-verdier mellom $0\,\mathrm{s/mm^2}$ and $1300\,\mathrm{s/mm^2}$ fra 81 pasienter fra OxyTarget-studien med pasienter med endetarmskreft ble brukt. Bildedataene ble splittet i et treningssett (51 pasienter), et valideringssett (10 pasienter) og et testsett (20 pasienter), jevnt fordelt med hensyn på kjønn og kreftstadier. Totalt ble det utviklet ni modeller med en U-net arkitektur der type bilder og tapsfunksjonen varierte. De ulike tapsfunksjonene som ble testet var cross entropy tap, Dice tap og en modifisert versjon av Dice tapet. To radiologer hadde tegnet inn omrissene av kreftsvulstene manuelt på bildene, og unionen av disse ble definert som fasit for modellene. Modellene ble evaluert basert på den gjennomsnittlige Dice likhetskoeffisient (DSC) per pasient for pasientene i valideringssettet. Den beste U-net modellen ble så sammenlignet med resultatene fra en grunn maskinlæringsmodell basert på den lineære støttevektorklassifikatoren.

Resultatet for de ulike U-net modellene varierte fra en DSC lik 0.58 til en DSC lik 0.67, og den beste modellen brukte T2-vektede bilder og den modifiserte Dice tapsfunksjonen. Sammenlignet med modellen med den lineære støttevektorklassifikatoren, som resulterte i en DSC lik 0.48, var U-net modellene overlegne. DSC mellom de to manuelle inntegningene ble kalkulert til å være 0.78, og dette indikerer at U-net modellen må forbedres før den kan brukes klinisk.

U-net modellen viser uansett lovende resultater for automatisk segmentering av kreftsvulster. For å utvikle en forbedret modell bør effekten av å bruke bilder med høyere oppløsning, samt å legge til data augmentasjon og bilde beskjæring, undersøkes.

Contents

List of figures

List of tables

Abbreviations

ACC:	Accuracy
ADC:	Apparent diffusion coefficient
CNN:	Convolutional neural network
DICOM:	Digital Imaging and Communications in Medicine
DSC:	Dice similarity coefficient
DW:	Diffusion weighted
DWI:	Diffusion weighted image
ERR:	Error
FCN:	Fully convolutional network
FN:	False negative
FOV:	Field of view
FP:	False positive
FPR:	False positive rate
HDF5:	Hierarchical Data Format version 5
JSON:	JavaScript Object Notation
MR:	Magnetic resonance
MRI:	Magnetic resonance imaging
NIfTI:	Neuroimaging Informatics Technology Initiative
PRE:	Precision
RAM:	Random access memory
ReLU:	Rectified Linear Unit
RF:	Radio frequency
SVC:	Support vector classifier
TE:	Echo time
TN:	True negative
TP:	True positive
TPR:	True positive rate
TR:	Repetition time

Chapter 1

Introduction

Cancer is a group of diseases characterized by uncontrolled cell division. The cancer cells also possess the ability to invade neighboring tissues and spread to other parts of the body through for example the bloodstream or the lymphatic system. It is estimated that cancer caused 9.6 million deaths worldwide in 2018 [1, 2].

In 2018 there were 34190 new cancer incidents in Norway [3]. Cancers in the rectum and rectosigmoid had 1360 new incidents, which corresponds to 4% of the total number of cancer incidents. This makes rectal cancer the seventh most frequent type of cancer in Norway. The relative survival with this cancer type after five years is 69.8% [3].

National guidelines state that patients diagnosed with rectal cancer should undergo a preoperative magnetic resonance imaging (MRI) examination to determine the stage of the disease [4]. From the image-based staging, the optimal treatment of the patient can be decided. For patients with locally advanced rectal cancer, the tumor has grown into the bowel wall and/or invaded nearby organs. These patients will receive preoperative chemoradiotherapy to reduce the size of the tumor, thereby enabling a better outcome of the subsequent surgery. Other patients with localized disease will be directly referred to surgery only [5, 6].

The delineation of the tumor volume is needed to make plans for radiation treatment. Today this delineation is done manually by radiologists or oncologists, and there is a significant interobserver variation which creates uncertainties [7]. Accuracy in the delineation is crucial because it is one of the first steps in the planning process, and an error in the target volume will generate a systematic error in the resulting treatment plan [8, 9]. This can impact the cure rate and toxicity of the treatment since the goal is to give a high dose to the tumor while limiting the dose to organs at risk and normal tissue. Another drawback of manual delineation is that it is a very time-consuming task. The time it takes to perform the delineation for one tumor can range from one minute to approximately 20 minutes [10].

In the last few years, there have been great progress and interest in the field of artificial intelligence and deep learning [11]. This is partly due to the rapid improvements in computational power, fast data storage, and parallelization, which makes it possible to analyze large amounts of data [12]. Deep learning approaches

based on convolutional neural networks have shown promising results for image segmentation with biomedical images [13, 14]. It might be possible to create a model that automatically segments the tumor volume, and in that way provide a standardized method for delineation. This would eliminate the interobserver variations and be time-saving for the radiologists if the results from the model are sufficiently accurate.

MR images were traditionally only evaluated visually by radiologists to determine the stage and size of the tumor. In recent years, MR images are also used to identify cancer biomarkers. A biomarker is a characteristic that can be measured objectively and act as an indicator of biology processes, pathological changes, or response to an intervention [15]. Such markers can give more information about the aggressiveness of the disease and be used to evaluate treatment response and predict the survival of the patient if he/she receives a given treatment. Delineations of tumor volumes are needed to calculate tumor biomarkers. Radiomics is a growing field that seeks to identify biomarkers by analyzing a large amount of image feature data. To make the results obtained from radiomics reliable and reproducible, a standardized method for delineation would be beneficial [16].

The aim of this thesis was to train a deep convolutional neural network with MR images from patients with rectal cancer in order to create a model for automatic segmentation of the tumor volume. The accuracy of the model was evaluated and compared with results obtained from a shallow machine learning approach where classification was done based on voxel intensities and with manual delineations by two radiologists.

Chapter 2

Theory

2.1 Magnetic resonance imaging

Magnetic resonance imaging (MRI) is an imaging technique used to form images of the anatomy and functions of the body. This section is taken from the author's project thesis, written during the fall semester of 2019, with minor adjustments, and it is based on the book *MRI in Practice* [17] unless other is stated.

MRI is an imaging technique used to form images of the anatomy and functions of the body. The technique is based on the spin and magnetic moment of nuclei. An MR active nucleus has an odd mass number and therefore a net spin. Nuclei with a net charge and spin will have a magnetic moment, the same way as a current moving through a coil induces a magnetic field. In this case, the nuclei then act as a small magnet. In human applications hydrogen (^1H) is the most used nuclei because of its relatively large magnetic moment, and the fact that a large amount of the body consists of water which means that it is a lot of hydrogen available. The spins are randomly oriented, but when an external magnetic field is applied, the nuclei tend to align their axis of rotation to the magnetic field. They can align parallel or anti-parallel to the field, and there is a slight preference for parallel because this corresponds to a lower energy state. This leads to a net magnetization in the direction of the magnetic field. The spins will precess around the magnetic field, B_0, with a frequency, called the Larmor frequency, w_0.

$$w_0 = \gamma B_0, \tag{2.1}$$

where γ is the gyromagnetic ratio which expresses the relationship between the magnetic moment and the angular momentum. This is a constant specific to the nuclei type.

A radio frequency pulse can be applied at the Larmor frequency to excite the spins. By exciting the spins, the net magnetization vector can be moved away from alignment with B_0. The flip angle is referred to as the angle the net magnetization vector is moved out of alignment, and this angle is often 90°. That will say that the net magnetization is moved from the longitudinal plane to the transverse plane. The nuclei will then precess in the transverse plane and produce magnetic field fluctuations inside a receiver coil. This induces an electrical voltage, and this is

the MR signal. The net magnetization vector will try to realign with the B_0 field, and in this process, the nuclei transfer energy to the surroundings. A decrease in the magnetization in the transverse plane and recovery of the magnetization in the longitudinal plane will then occur. This is called T1 relaxation. It is an exponential process, and the time it takes for 63% of the longitudinal magnetization to recover is called T1.

The spins in the transverse plane will start in phase after the excitation pulse and then dephase. This dephasing is due to spin-spin interactions, T2, and inhomogeneities in the magnetic field, T2'. The T2' dephasing is a systematic effect that can be reversed, while T2 is a random effect and varies with the nuclei type. The total dephasing is referred to as T2* decay, and the relationship between T2, T2' and T2* is given by the following equation.

$$\frac{1}{T2^*} = \frac{1}{T2} + \frac{1}{T2'} \tag{2.2}$$

In MR sequences the repetition time, TR, is the time from the application of the excitation pulse to the application of the next excitation pulse. This time determines the amount of T1 relaxation that is allowed to occur before the signal readout. The echo time, TE, is the time from the application of the excitation pulse to the peak of the signal that is induced in the receiver coil. This determines how much T2 relaxation that is allowed to happen before the readout.

A spin-echo sequence is one of the most used pulse sequences in MRI. In this sequence, a 90° excitation pulse is applied to flip the net magnetization to the transverse plane. A free induction decay signal will occur, and after a time TE/2 a 180° pulse is applied to rephase the spins and we get a spin echo signal at TE. Figure 2.1 shows a vector representation of the dephasing and rephasing of the spins.

To form an image it is important to determine the spatial location of the signal. This is done with the use of magnetic field gradients. The Larmor frequency is dependent on the magnetic field strength, and a nucleus experiencing a high magnetic field strength will have a lager Larmor frequency than a nucleus experiencing a lower field strength. To select a slice in the z-direction, often the direction from feet to head of a patient, a gradient is applied in the B_0 direction. The Larmor frequency of the spins will now vary along the z-direction. The excitation pulse with a band of frequencies equal to the Larmor frequencies of the spins in the wanted slice is applied, and only spins in this slice will get excited. The slice thickness is dependent on the bandwidth of the pulse and the steepness of the gradient. In a spin-echo sequence, the slice selection gradient is on during the 90° and 180° radio frequency pulse. The two remaining directions are called the frequency encoding direction and the phase encoding direction. A gradient in the frequency encoding direction is switched on during the readout of the signal. Signals from different locations along this gradient will have different frequencies. In the phase encoding direction, a gradient is applied after the excitation pulse. This gradient is only on for a given amount of time and induces a phase shift between spins along the phase encoding gradient. The resulting pulse sequence with all the gradients is shown in figure 2.2.

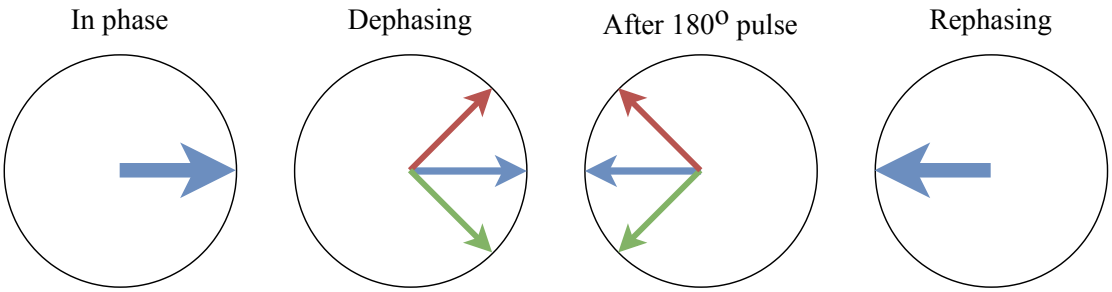


Figure 2.1: Vector representation of the spin dephasing and rephasing in a spin-echo sequence. The blue arrow represents the spins that rotate at the Larmor frequency, the green arrow represents the spins that rotates a bit faster and the red arrow represents the spins that rotate a bit slower.

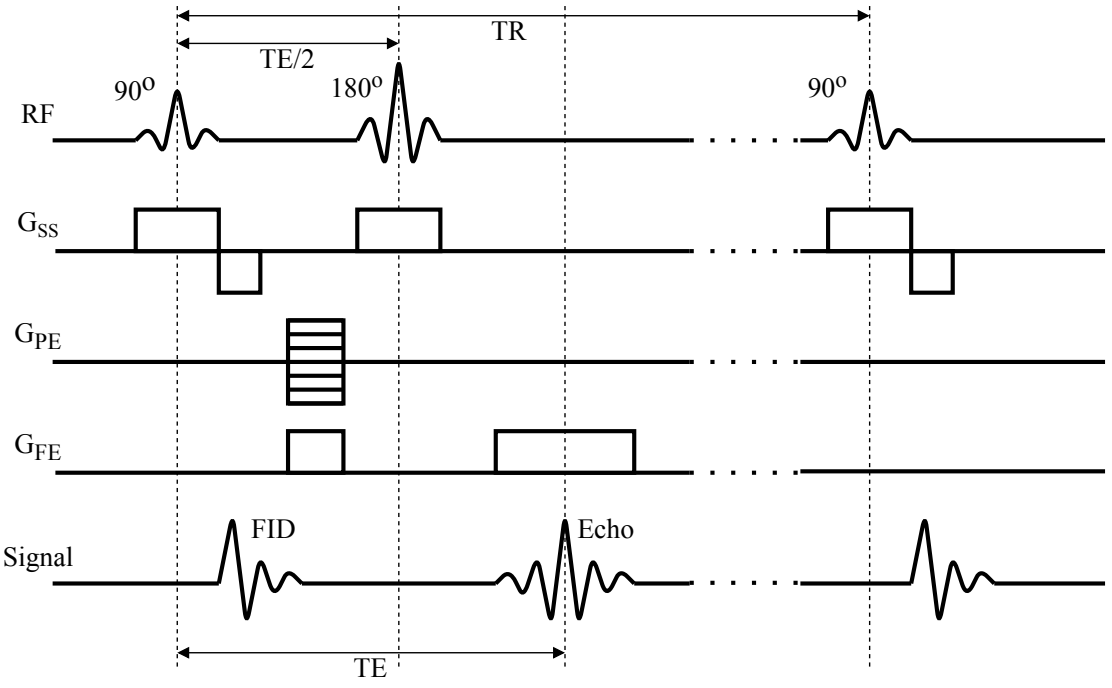


Figure 2.2: Spin echo pulse sequence and spatial encoding gradients. RF is the radio frequency pulses, G_{SS} is the slice selecting gradient, G_{PE} is the phase encoding gradient and G_{FE} is the frequency encoding gradient.

The recorded signal is mapped to a spatial frequency domain, the so-called k-space. The horizontal lines correspond to the frequency encoding while the vertical lines correspond to the phase encoding. A 2D Fourier Transform is applied to reconstruct the image from k-space.

By applying different pulse sequences, different contrasts can be obtained in the images. It is possible to have sequences that highlight the anatomy but also sequences that highlight functional properties like diffusion.

2.1.1 T2 weighted images

In T2 weighted images, water/fluid will appear bright, fat will appear intermediate-bright, while air and muscle will appear dark. This can be seen in the T2 weighted image in figure 2.3. The image contrast is a result of the fact that different tissues have different T2. Fat molecules can easily absorb energy into its lattice from the hydrogen nuclei due to low inherent energy. From this, it follows that the longitudinal magnetization is able to recover quickly in fat, and fat has a short T1. Water, on the other hand, has high inherent energy and does not absorb energy into its lattice easily. Because of this, it takes water a longer time to recover the longitudinal magnetization and it has a long T1. The fat molecules are packed closely together, and spin-spin interactions are likely to occur. The spins in fat will dephase quickly, which leads to a short T2. The spin-spin interactions are less likely to occur in water because there is more space between the molecules, and water has a long T2.

To get a T2 weighted image the difference in T2 for water and fat needs to be enhanced, and the difference in T1 needs to be diminished. This can be controlled by adjusting TE and TR. The TE must be long enough so that both fat and water have time to decay. Since water has the longest T2, it will be most signal left from water. The TR must be so long that both water and fat get time to fully recover their longitudinal magnetization, and therefore the difference in T1 will not create contrast in the image.

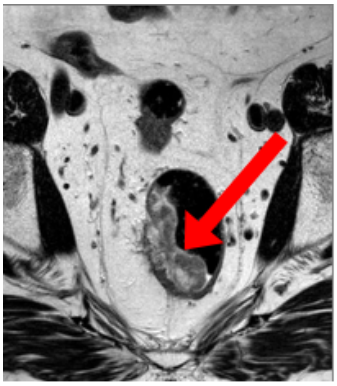


Figure 2.3: A T2 weighted image of a patient with rectal cancer. The red arrow points at the tumor.

2.1.2 Diffusion weighted images

Diffusion is referred to as the random Brownian motion of molecules driven by thermal energy [18]. In a perfectly homogeneous medium, the probability for motion will be the same in all directions, and there will be free diffusion. This is not the case in a complex environment like the human body. In the body, there are intracellular and extracellular compartments. In the extracellular regions, the water molecules will experience a relatively free diffusion while there will be a more restricted diffusion in the intracellular regions. Different tissues have different proportions of intra- and extracellular compartments and characteristic cellular architecture. This means that the diffusion properties vary with the tissue. In tumors, there is a higher cell density than in healthy tissue, and this results in a more restricted diffusion.

In a diffusion weighted image (DWI) the contrast is determined by the diffusion of water molecules [19]. The presence of a magnetic field gradient will cause a phase shift in the spins, and the cumulative phase shift, ϕ, for a single static spin is given by

$$\phi(t) = \gamma B_0 t + \gamma \int_0^t \mathbf{G}(t') \cdot \mathbf{x}(t') dt'. \tag{2.3}$$

In equation (2.3) the first term is due to the static B_0-field and the second term is due to a magnetic field gradient. $\mathbf{G}$ is the strength of the gradient, $\mathbf{x}$ is the spatial location of the spin and t is the duration of the gradient.

A normal pulse sequence in DW imaging consists of a T2-weighted spin-echo sequence and two equal gradient pulses applied before and after the 180° refocusing pulse. This is called a Stejskal-Tanner sequence [20], and it is shown in figure 2.4.

The phase shift due to the applied gradient will for an individual spin be proportional to the displacement of the spin along the direction of the gradient [19]. At the echo time, TE, the total phase shift for a particular spin is equal to

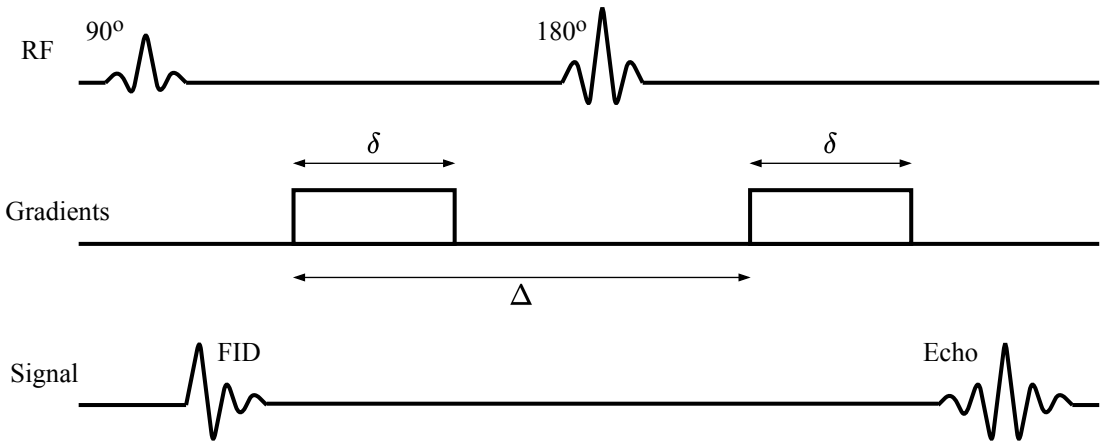


Figure 2.4: Stejskal-Tanner sequence consisting of a spin echo pulse sequence together with the diffusion gradients used in diffusion weighted imaging.

$$\phi(TE) = \gamma \int_{t_1}^{t_1+\delta} \mathbf{G}(t') \cdot \mathbf{x}(t')dt' - \gamma \int_{t_1+\Delta}^{t_1+\Delta+\delta} \mathbf{G}(t') \cdot \mathbf{x}(t')dt'. \tag{2.4}$$

Here δ is the time the gradient is applied for and Δ is the time between the first and the second gradient. From equation (2.4) it is clear that if there is no displacement along the gradient, the two terms will cancel. That results in no net phase shift. With diffusion, each spin acquires a random displacement and the phase shift for the individual spins will vary. Only the spins with no moment will be refocused perfectly and diffusion leads to a reduction of the signal. Regions with strongly restricted diffusion, like tumors, will therefore appear bright in the images while regions with relatively free diffusion will appear dark. This can be seen in figure 2.5a.

It can be shown that the diffusion results in an echo attenuation given by

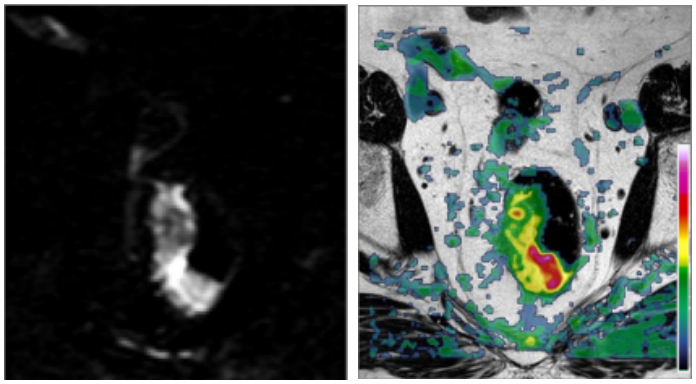
$$S(b, TE)_{SE} = S_0 \exp\left(-\frac{TE}{T_2}\right) \exp\left(-b \cdot ADC\right), \tag{2.5}$$

where b refers to the diffusion-sensitizing factor, also called b-value, and ADC is the apparent diffusion coefficient. The b-value determines the amount of diffusion weighting in the image, and it can be calculated as follows

$$b = \gamma^2 G^2 \delta^2 (\Delta - \frac{\delta}{3}). \tag{2.6}$$

A b-value equal to zero will correspond to a T2 weighted image with no diffusion weighting. Figure 2.5b shows an ADC map, and it reflects the degree of restricted diffusion. The ADC can be calculated from equation (2.5) by using at least two different b-values, and one gets

$$ADC = -\frac{1}{b_1 - b_0} \ln\left(\frac{S(b_1)}{S(b_0)}\right). \tag{2.7}$$



(a) DWI **(b)** ADC map

Figure 2.5: A diffusion weighted image (a) and an ADC map (b) for a patient diagnosed with rectal cancer.

2.2 Machine learning

Machine learning is a field closely related to artificial intelligence, pattern recognition, statistics, optimization, and computer science [21]. It can be defined as the use of computer algorithms that improves the performance of a given task by learning from experience [22]. Different statistical learning methods are used by the algorithms to create decision boundaries, and these can be used to make predictions on new data.

Machine learning can be divided into two main categories, supervised and unsupervised learning. With supervised learning, both the data and the corresponding labels are used to train the model, and the performance of the model can be determined from how well predicted labels correspond to the real labels. With unsupervised learning, on the other hand, the data has no labels and it is up to the model to find patterns within the data. A typical example of this is clustering which seeks to separate the data into distinct subsets. There is no straight forward method to determine the model performance for models with unsupervised learning. In addition to supervised and unsupervised learning, there also exists a third category, reinforcement learning, which is often applied when teaching a machine to play games. In this case, the model gets feedback based on the outcome of the game [23].

The predictions made by a machine learning model can either be quantitative or qualitative. A quantitative variable will have a numerical value, and can, for example, be life expectancy, while a qualitative value will be set to one of N different categories. An example of this could be a person's gender (male or female) or the result of a medical test (positive or negative). A regression problem is a case where the model should output quantitative values, while a classification problem would refer to a model that outputs qualitative values [24].

Deep learning is a subfield of machine learning where the data is processed in several hierarchical layers in order to understand more complex features and representations of the data [23]. Shallow machine learning approaches mainly looks at one representation of the input data, and will therefore only be able to make accurate predictions if this representation contains features that are clearly related to the expected output. It, therefore, lacks the level of abstraction found in deep learning [23]. Deep learning has the advantage that one can input raw data and the model will learn to automatically extract the features that are relevant for the predictions. Most deep learning models are neural networks with several hidden layers. The following pages will give a short introduction to how neural networks work, and how they can be used for image recognition.

2.2.1 Neural networks

A neural network is a model that consists of several layers of processing units referred to as neurons, and an illustration of this is shown in figure 2.6. Each neuron takes an input and process the data before it is sent to neurons in the next layer. The connections between the different neurons can vary in strength, and the strength of these connections will determine how the data is processed [23].

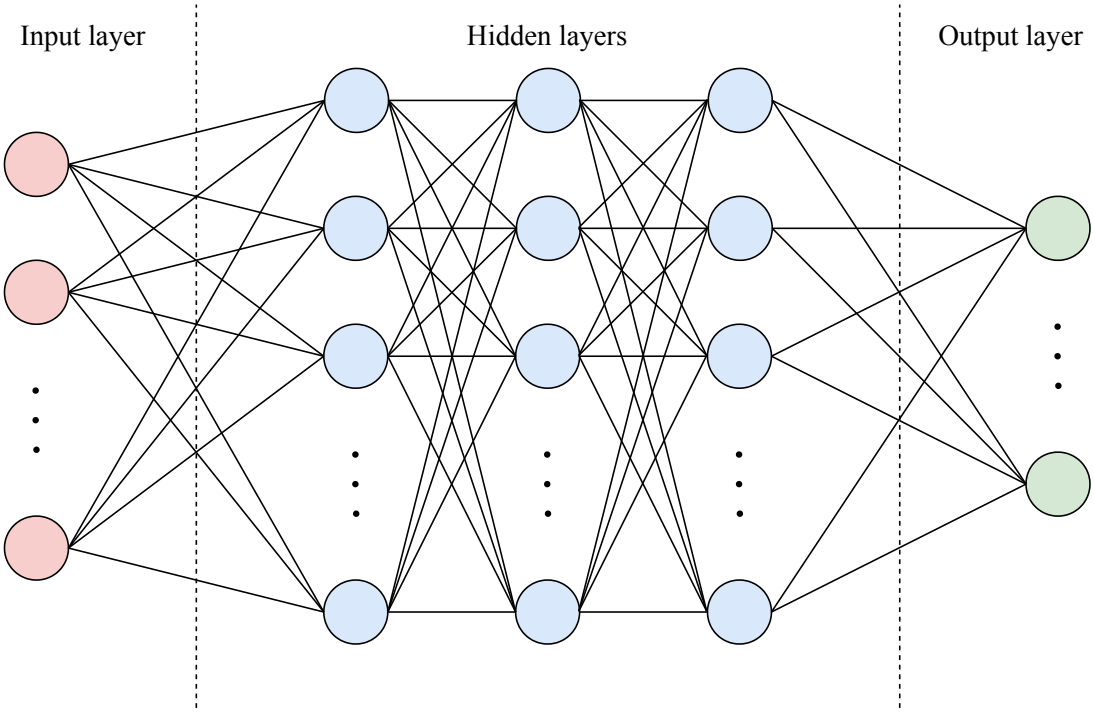


Figure 2.6: Illustration of a possible structure of neurons in a neural network with three hidden layers. The circles correspond to the neurons, and the solid lines represent the connections between the neurons.

The neurons will have an internal state depending on the input, which is a sum of outputs from neurons in the previous layer. A simple function for this internal state value, or activation value, is given by the following equation.

$$a(x) = \sum_i w_i x_i, \tag{2.8}$$

where x_i is the input originating from neuron number i in the previous layer, and w_i is the strength of the connection between the two neurons. If one consider $\mathbf{x}$ and $\mathbf{w}$ as vectors, the activation value will be the dot product between these two [23]. $\mathbf{w} \cdot \mathbf{x} = 0$ will define a hyperplane in $\mathbf{R}^d$, where d is the dimension of $\mathbf{x}$. A vector $\mathbf{x_1}$, which gives $\mathbf{w} \cdot \mathbf{x_1} > 0$, is a vector that lies on one side of the hyperplane while a vector $\mathbf{x_2}$ such that $\mathbf{w} \cdot \mathbf{x_2} < 0$ lies on the other side. Each neuron can therefore act as a classifier. It is possible to include a bias which will shift the hyperplane away from the origin, and that will result in the following function for the activation value.

$$a(x) = \sum_i w_i x_i - b \tag{2.9}$$

The output from the neuron is determined by an activation function that takes the activation value as input, and this data processing that takes place in the neuron is illustrated in figure 2.7. One of the simplest activation functions is the identity function, $f(a) = a$. This is a linear function where the output of the

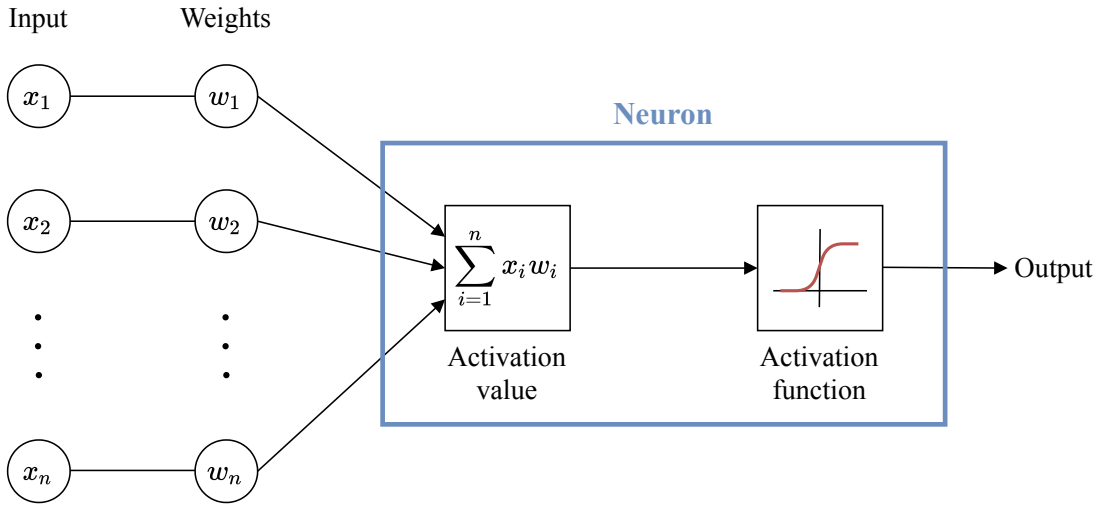


Figure 2.7: Illustration of the neuron composition. The input values, x_i, are multiplied with their corresponding weight, w_i, and summed up to the activation value. This activation value is sent to the activation function which determine the output from the neuron.

neuron equals the activation value. An example of a non-linear activation function is the threshold function in equation (2.10). It results in activation of the neuron (output equal to 1) if the activation value is above a certain threshold value and an output of zero if it is below.

$$f(a) = \begin{cases} 1 & \text{if } a \geq 0 \\ 0 & \text{if } a < 0 \end{cases} \tag{2.10}$$

A combination of the identity and the threshold function yields the Rectified Linear Unit function, ReLU, which is shown in the following equation.

$$f(a) = \begin{cases} a & \text{if } a \geq 0 \\ 0 & \text{if } a < 0 \end{cases} \tag{2.11}$$

Another activation function that is commonly used is the logistic sigmoid function,

$$f(a) = \frac{1}{1 + \exp(-a)}. \tag{2.12}$$

The output from this function is bound between 0 and 1, and it can be interpreted as the probability for the neuron to activate. In a neural network, all neurons in the same layer tend to have the same activation function, but neurons in different layers can have different activation functions. The choice of the activation function is related to the underlying problem [23].

When training a neural network, the strength of the connections between different neurons, the weights, are first initialized as small random numbers. The goal is then to optimize these weights so that the error in the predictions made by the network is minimized. The error is calculated with a loss function, $J(\mathbf{w})$,

and it represents the difference between the predicted values and the true values. For regression problems, the squared error loss function is commonly used, and it is given as

$$J(\mathbf{w}) = \sum_i (y^i - t^i)^2, \tag{2.13}$$

where y^i is the predicted value and t^i is the target value or true value for sample i. The cross entropy loss function is often used for classification problems, and for classification with two distinct classes (1 and 0) it is defined as

$$J(\mathbf{w}) = -\frac{1}{n}\sum_{i=1}^{n} t^i \ln(\sigma(a^i)) + (1 - t^i)\ln(1 - \sigma(a^i)), \tag{2.14}$$

where $\sigma(a^i)$ is the probability that sample i belongs to class 1 with the given weights. To use the cross entropy loss function it is important to use an activation function that outputs a value between 0 and 1 that can be interpreted as a probability function, like the logistic sigmoid function [23].

To minimize the loss function, and thus optimize the weights, the weights are updated iteratively. One widely used method for deciding how the weights should be updated is the gradient descent. The loss function is a function of the weights in the network, and by calculating the gradient of the loss function, one finds the direction with the steepest slope at given points. The weights can then be updated in the opposite direction of the gradient, and as a result, the next iteration will yield a lower loss. The weight update is given by the following equation.

$$\mathbf{w}^{(i+1)} = \mathbf{w}^{(i)} - \lambda \nabla J(\mathbf{w}^{(i)}), \tag{2.15}$$

where λ is the magnitude of the update, the learning rate, and $\nabla(J(\mathbf{w}^{(i)}))$ is the gradient of the loss function. It is important to choose a reasonable value for the learning rate. A too small learning rate will lead to unnecessary many iterations and one can get stuck in local minima. On the other hand, a too large learning rate might not lead to a minimum at all, only a random location on the curve [25].

There are several challenges with the gradient descent method. The convergence speed can be low due to oscillations around the minimum, and it is possible to get stuck in local minima. Momentum gradient descent is a method that was developed to address these two issues [26]. With this method, the weights are not only updated based on the current gradient but also the previous. A useful analogy can be to think of a ball rolling down the loss function. The movement of the ball will not only depend on the current acceleration but also the velocity resulting from previous acceleration. If the ball has enough momentum, it will get past the local minima and end up in the global minimum. The adjustment to the weights, $\Delta\mathbf{w}^{(i)}$, with the momentum gradient descent is defined as

$$\begin{aligned} \Delta\mathbf{w}^{(i)} &= \gamma\Delta\mathbf{w}^{(i-1)} - \lambda\nabla J(\mathbf{w}^{(i)}) \\ \mathbf{w}^{(i+1)} &= \mathbf{w}^{(i)} + \Delta\mathbf{w}^{(i)}, \end{aligned} \tag{2.16}$$

where γ is a parameter that controls how much the previous iteration should be weighted.

An adaptive learning rate optimization algorithm, *Adam*, was proposed by Kingma and Ba [27]. It is a versatile optimization algorithm that can be used for large-scale high-dimensional machine learning problems and has, therefore, become a popular algorithm to use for neural networks. Adam does not only include the momentum, but it also modifies the learning rate for each weight. If one weight gets a very large update in the previous iteration, this indicates numerical instabilities, and the learning rate is decreased. Similar, if the weight had a small update in the last iteration, it might be on a plateau, and increasing the learning rate could lead to faster convergence. The algorithm updates the moving average of the gradient, $\mathbf{m}^{(i+1)}$, and the moving average of the squared gradient, $\mathbf{v}^{(i+1)}$, in the following manner.

$$\begin{aligned} \mathbf{m}^{(i+1)} &= (1-\beta_1)\nabla J(\mathbf{w}^{(i)}) + \beta_1 \mathbf{m}^{(i)} \\ \mathbf{v}^{(i+1)} &= (1-\beta_2)(\nabla J(\mathbf{w}^{(i)}))^2 + \beta_2 \mathbf{v}^{(i)}, \end{aligned} \tag{2.17}$$

where β_1 and β_2 are hyper-parameters between 0 and 1 that control the exponential decay rates of $\mathbf{m}^{(i+1)}$ and $\mathbf{v}^{(i+1)}$. $\mathbf{m}^{(i+1)}$ and $\mathbf{v}^{(i+1)}$ can be seen as an estimate of the first moment (the mean) and the second moment (the uncentered variance) of the gradient respectively. Both the moments are initialised as zero, and thus introduce a bias towards zero in the estimates. A bias correction is therefore applied, and the bias-corrected moments are defined as

$$\begin{aligned} \hat{\mathbf{m}}^{(i+1)} &= \frac{\mathbf{m}^{(i+1)}}{1-\beta_1^i} \\ \hat{\mathbf{v}}^{(i+1)} &= \frac{\mathbf{v}^{(i+1)}}{1-\beta_2^i}. \end{aligned} \tag{2.18}$$

The Adam algorithm then defines the weight update as given by the following equation.

$$\Delta \mathbf{w}^{(i)} = -\lambda^{(i)} \frac{\hat{\mathbf{m}}^{(i+1)}}{\sqrt{\hat{\mathbf{v}}^{(i+1)}} + \epsilon} \tag{2.19}$$

Here, ϵ is a small number included to ensure numerical stability.

When one has a neural network with only one layer, the weight optimization can easily be understood. With several hidden layers, the method is not so straight forward, and a method called back-propagation is used. The idea behind this method is that the error in the last hidden layer is calculated and then an estimate of the error in the previous layer is made. The error is propagated backward from the last layer to the first layer [11]. A complete mathematical description of the back-propagation algorithm is beyond the scope of this thesis, but it is mainly use of the chain rule.

2.2.2 Overfitting

A problem with complex models like neural networks is overfitting. An overfitted model has learned the noise in the data used for training and will produce predictions with very high accuracy for this data, but it will not perform well on new unseen data [23]. At the beginning of the training, the model will improve its performance on both training data and unseen data with a better optimization based on the training data. At this stage, it is still relevant features for the model to learn, and the model is said to be underfit. At one point the model becomes overfitted, and it has then learned features that are specific to the training data but that are irrelevant or misleading when it comes to new data [11]. There is a compromise when it comes to optimization and generalization, but a model trained on a larger amount of data will generalize better.

Training on more data is not always possible, but there are other ways to avoid overfitting. One can regulate the quantity of information the model is allowed to store or add a constraint on the information that can be stored. In this way, the model is forced to focus on the most dominant patterns, and this approach is referred to as regularization [11]. The simplest way one can do this is to reduce the network size. The number of parameters that the model can learn depends on the number of layers together with the number of units within each layer, and this is called the capacity of the network. The adjustment of the capacity of the model will be a compromise between an overfitted and an underfitted model, in other words too much capacity or not enough capacity.

Another option is to add weight regularization. The weights can be forced to be small by adding a cost for having large weights to the loss function. This will result in a less complex model because the weights will have a more regular distribution [11]. There are two common ways to implement weight regularization, L1 and L2 regularization. With L1 regularization a cost proportional to the absolute value of the weight coefficients is added, while with L2 regularization the added cost is proportional to the square of the value of the weight coefficients.

One of the most effective and most commonly used methods to avoid overfitting is dropout. With this method, some output features of the layer are randomly selected and set to zero (dropped out) during the training of the model [11]. The model is thus forced to learn a more robust representation of the data, and the predictions can not only depend on a few specific features. The fraction of the features that are set to zero is the dropout rate, and this is usually between 0.2 and 0.5. The dropout is only done during the training of the model, and when it is run on test data the output values are scaled with a factor equal to the dropout rate. This is done to compensate for the fact that there are more active units than during the training.

2.2.3 Image recognition with convolutional neural networks

When applying neural networks for image recognition, a type of layers called convolution layers are almost always used [23]. Instead of getting input from each of the neurons in the previous layer like fully connected layers does, a neuron in a convolution layer only receives input from a small sub-region of neighboring neurons in the previous layer. These neighboring neurons will correspond to neighboring pixels in the image. In this way, a neural network with convolution layers, convolutional neural networks, will reduce the number of parameters needed and consequently help avoid overfitting [23].

A convolution layer can be seen as an image filter that highlights certain features [23]. The filter is a two-dimensional matrix, usually 3×3 or 5×5, containing weights that are moved across the input image as shown in figure 2.8. The filter is usually moved one pixel at the time, and the number of pixels moved corresponds to the stride. The result obtained from each move corresponds to the activation value of a neuron in the convolution layer. One convolution layer can contain several filters and the output will then be a set of images with different features or characteristics highlighted. This set of images will be referred to as feature maps, and the depth of the feature map, also called number of channels, corresponds to the number of images. The depth of the output feature map will be the depth of the input feature map multiplied with the number of different filters. A convolution layer with an input feature map with a depth equal to two and two different filters is shown in figure 2.9.

One large advantage with convolutional neural networks is that the key patterns that they learn are translation invariant [11]. A pattern that occurs at one location and is learned by the network can be recognized by the network even if it appears at another location in the image. With a fully connected network, the pattern would have to be learned again if it were to appear at a different location. Due to this, a convolutional neural network needs fewer samples to learn representations that can be generalized compared to a fully connected network. Convolutional neural networks can also learn hierarchies of patterns [11]. The first convolution layer will be able to learn small local patterns, like for example edges, while a second convolution layer will learn larger and more complex patterns from the features in the first layers.

When convolution is applied to an image, the size of the output image becomes smaller than the input image. A 3×3 filter can be centered around every pixel in the image except the ones around the edge. This will lead to an output image with two fewer pixels in each dimension compared to the input image, as seen in figure 2.8. To get an output image with the same size as the input image, padding can be applied. This is done by adding zero value pixels around the original image [23], and the effect is shown in figure 2.10. If the filter has larger dimensions or the stride is larger, one would need more padding to achieve the same output image size. It is also possible to use padding to increase the size of the output image.

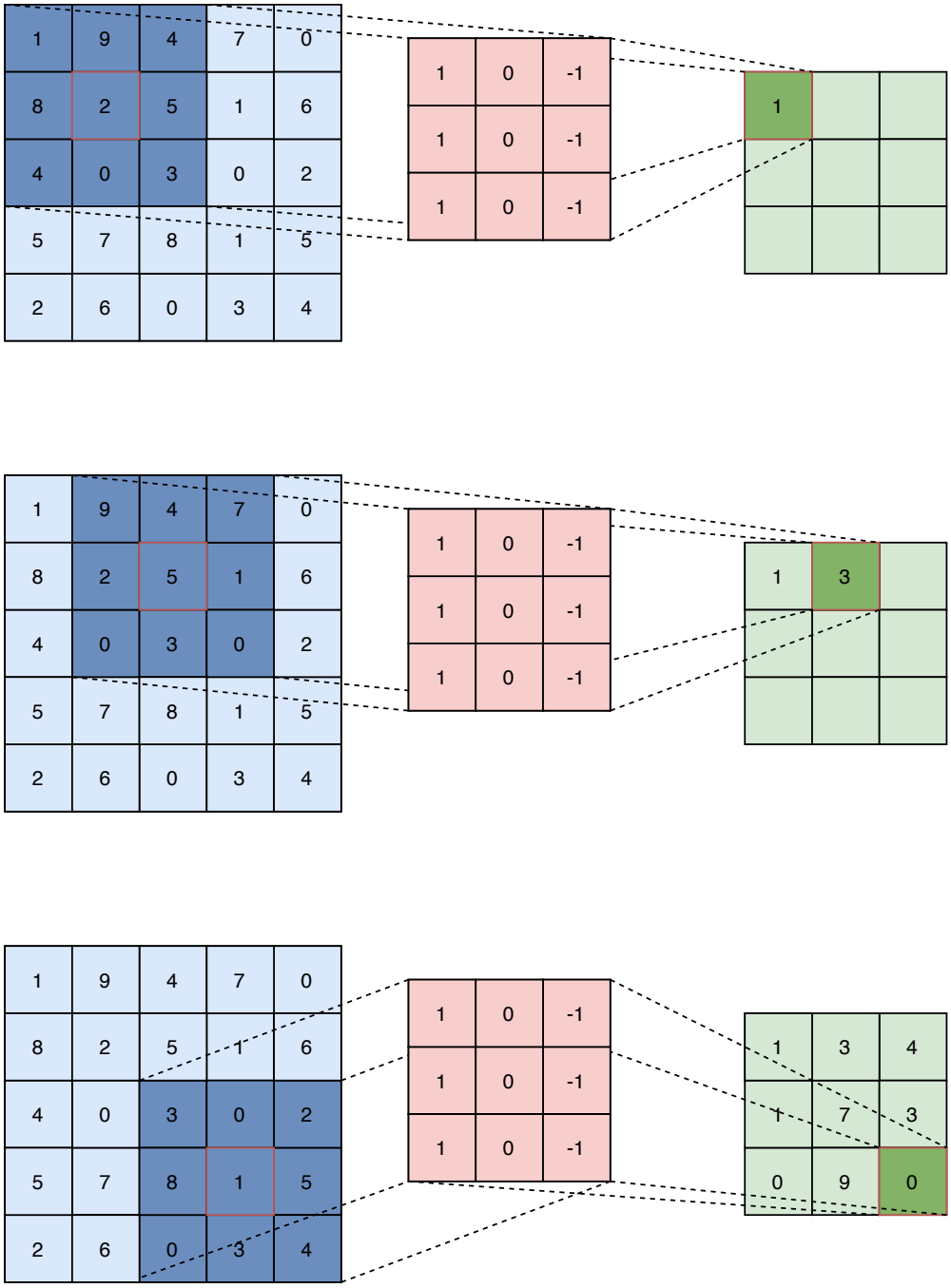


Figure 2.8: Illustration of a 2D convolution. The 3×3 convolution kernel, or filter, (pink) is moved across the 5×5 input (blue) with a stride equal to 1 and produces the 3×3 output (green). The output value in the top left corner is computed in the following way; $(1 \cdot 1) + (8 \cdot 1) + (4 \cdot 1) + (9 \cdot 0) + (2 \cdot 0) + (0 \cdot 0) + (4 \cdot (-1)) + (5 \cdot (-1)) + (3 \cdot (-1)) = 1$.

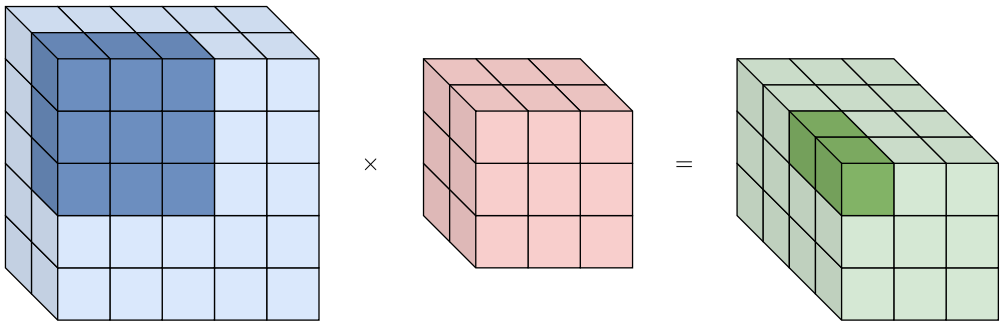


Figure 2.9: Convolution with two filters (pink) performed on an input feature map with two channels (blue). The resulting output feature map has four channels and is shown in green.

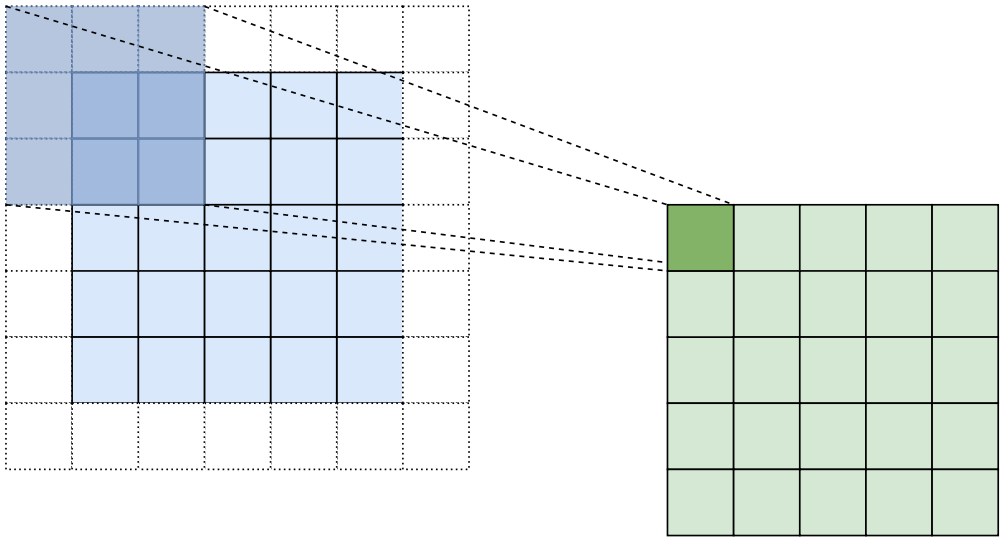


Figure 2.10: Illustration of the effect of adding same padding to a convolution with a 3×3 filter and stride equal to one. Zeros are added around the input image (blue) increasing the size with two in each dimension. The filter can then be centered around every pixel in the input which results in an output image with the same size as the input image.

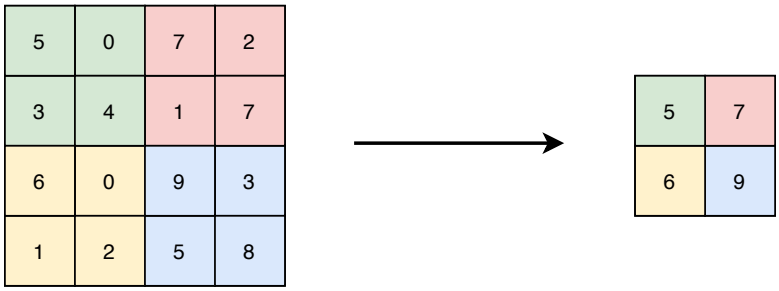


Figure 2.11: The max pooling operation with window size 2×2 and stride equal to two. The input is divided into a grid and the output consists of the maximum value from each of the windows. The output is consequently a 75% downsampling of the input.

Pooling layers are used in neural networks to reduce the number of feature map coefficients that need to be processed [11]. With max pooling, one creates grids, usually with 2×2 windows, on each image and keep the pixel with the maximum value within each window, as illustrated in figure 2.11. This operation will discard 75% of the neurons, and only the neurons that contribute the most will be kept [23]. It is also possible to use mean pooling where the average of the pixel values in the window used instead of the max, but max pooling tends to work better [11]. In addition to reduce the number of neurons, pooling layers also make sure that successive convolution layers look at increasingly large windows, and thus the network will be able to learn patterns that span a large area of the image.

2.2.4 Image semantic segmentation

When the goal is to perform image segmentation, it is not enough to have a network with only convolution layers and pooling layers. Semantic segmentation requires all the pixels in the input image to be assigned to a class, and the output of the network should, therefore, have the same resolution as the input. Through downsampling operations like pooling layers, the resolution of the feature maps are decreased, and it is therefore not straight forward to relate this to the original spatial resolution.

Fully convolutional networks, FCNs, have shown good results for semantic segmentation [28]. A FCN replaces the fully connected layers with convolutional layers with a filter size equal to the size of the input, and the output is a classification heat map. In this way, the network can take images with arbitrary size as input, but the produced heat maps are coarse and need to be upsampled to make the pixel-wise predictions. This upsampling can be done through for example bilinear interpolation or upconvolution, often referred to as deconvolution. Upconvolution can be thought of as an inverse convolution since the convolution connects several input neurons to one output neuron and the upconvolution, on the other hand, connects one input neuron to several output neurons [29]. In this way, the spatial resolution of the feature maps can be increased so that the original resolution is obtained.

Instead of upsampling to the input resolution in one operation, having an

upsampling part that mirrors the contraction part of the network has produced good results [29, 13]. A network where both the contraction and expansion is applied gradually creates a network architecture that is shaped like a "U", as shown in figure 2.12. Networks with this architecture is, therefore, often referred to as U-nets.

The contracting part of the network extract features from the input while the expansion part produces the object segmentation. The lower layers of the expansion tend to capture the overall shape of the object while the higher layers encode the finer details [29].

In order to restore the spatial information lost during the downsampling, long skip connections can be applied. Through these connections feature maps from the contraction part of the network are joined with the feature maps in the expansion part [30]. The input feature map to the last downsampling layer is concatenated with the output from the first upsampling layer, the input to the second last downsampling layer is concatenated to the output of the second upsampling layer, and it continues like that up through the network as seen in figure 2.12.

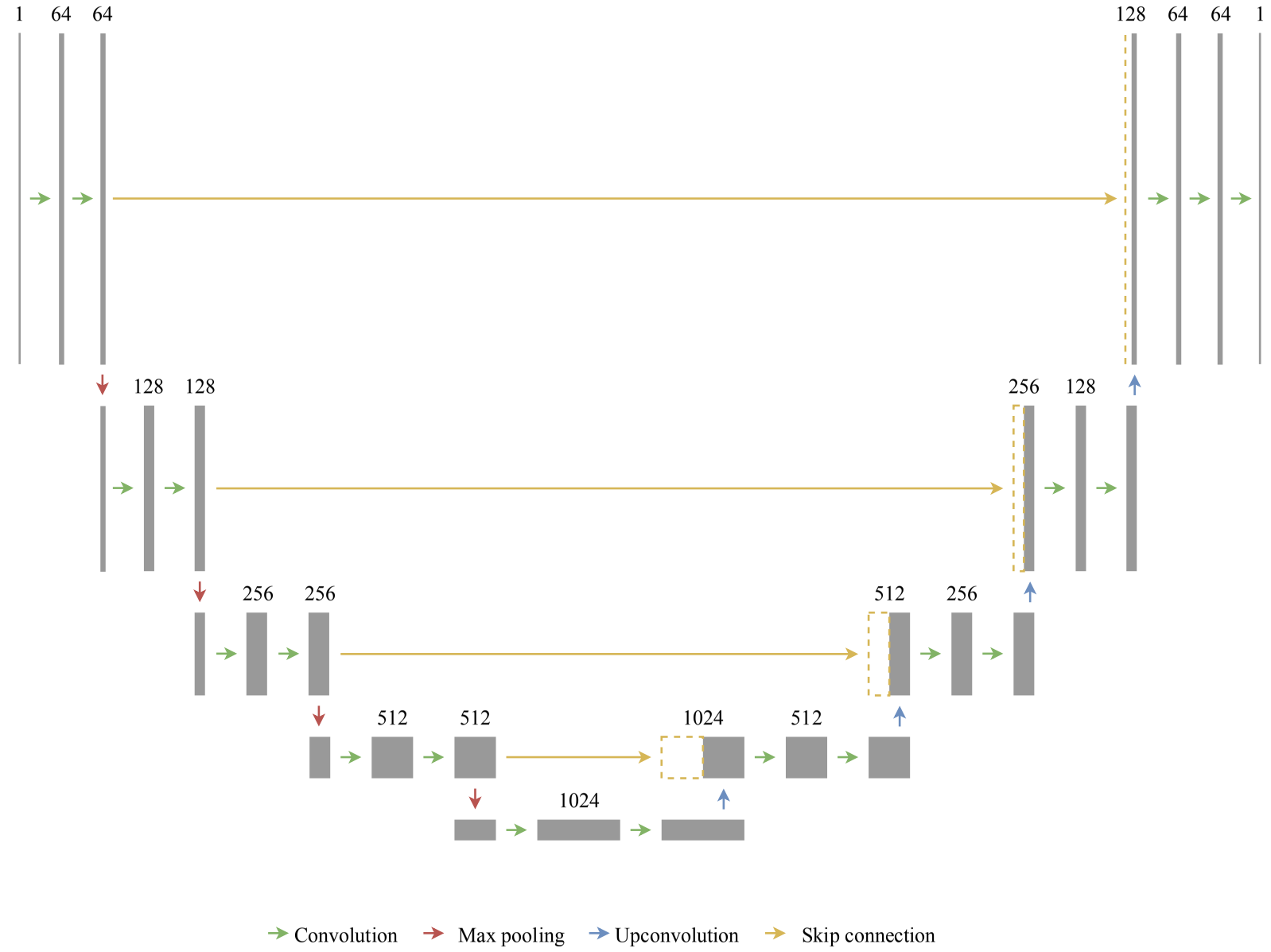


Figure 2.12: Illustration of the U-net architecture. The gray boxes represent feature maps where the height corresponds to the relative spatial size and the width corresponds to the number of channels which is also written above each box. The green arrows are convolutional layers with a 3×3 kernel, the red arrows are max pooling layers with a window size of 2×2 and the blue arrows correspond to upsampling layers that maps each pixel in the input to four pixels in the output. The yellow arrows represent the skip connections, and the yellow boxes with dotted lines are the inputs to the max pooling layers which are concatenated to the output of the upsampling layers through the skip connections.

2.2.5 Linear support vector classifier

A shallow machine learning approach can be an alternative to the deep neural networks for image segmentation. The linear support vector classifier, SVC, is a classification method that is based on the creation of a hyperplane in a p-dimensional space. A hyperplane is a $(p-1)$-dimensional flat subspace that divides the p-dimensional space into two halves and is given by the equation

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p = 0. \tag{2.20}$$

In order for the hyperplane to exist, at least one of the parameters $\beta_1, ..., \beta_p$ has to be non-zero. If a point $X = (X_1, X_2, ..., X_p)^T$ satisfy equation (2.20), then the point lies on the hyperplane [24]. For a point that is not on the hyperplane, the sign of the left hand side of equation (2.20) will indicate which side of the hyperplane the point belongs to.

Figure 2.13 illustrate the linear SVC method. Suppose we have n observations with p number of features. This will result in a $n \times p$ data matrix and each observation can be mapped to a p-dimensional space. The observations can be categorized into two different classes, ω_1 and ω_2. By creating a hyperplane that separates the two classes, any new observation can be classified depending on which side of the hyperplane it is located. The smallest of the perpendicular distances from each of the observations to the hyperplane is referred to as the margin.

It will not always be possible to separate the classes with a hyperplane, and

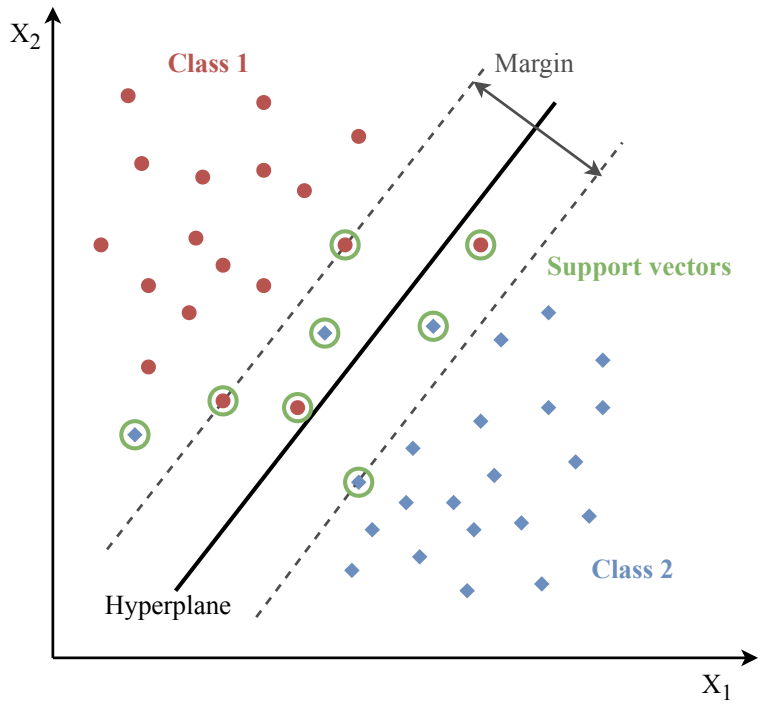


Figure 2.13: The principle of the linear support vector classification. An optimized hyperplane separates the two different classes, and observations violating the defined margin or the hyperplane are the support vectors.

in some cases, it is necessary to have some observations on the wrong side of the margin or the hyperplane in order to classify most of the observations correctly. The SVC determines the hyperplane so that it is a solution to the following optimization problem.

$$\begin{aligned} \underset{\beta_0, \beta_1, ..., \beta_p, \epsilon_1, ..., \epsilon_n, M}{\text{maximize}} \quad & M \\ \text{subject to} \quad & \sum_{j=1}^{p} \beta_j^2 = 1, \\ & y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + ... + \beta_p x_{ip} \geq M(1 - \epsilon_i), \\ & \epsilon_i \geq 0, \ \sum_{i=1}^{n} \epsilon_i \leq C, \end{aligned} \tag{2.21}$$

where M is the width of the margin, $\epsilon_1, ..., \epsilon_n$ are slack variables and C is a non-negative tuning parameter [24]. The slack variables make it possible for individual observations to be on the wrong side of the margin or the hyperplane. These violations are controlled by the tuning parameter. A smaller C will allow fewer observations to be located on the wrong side of the margin (and the hyperplane), and it also limits the severity of these violations.

The only observations that will affect the location of the hyperplane are the observations either laying on the margin or violating it. These observations are called the support vectors.

2.2.6 Performance metrics

When evaluating the performance of a classification model, it is useful to take a look at the confusion matrix. The confusion matrix is a square matrix containing the number of true positive (TP), true negative (TN), false positive (FP) and false negative (FN) predictions made by the classifier as shown in figure 2.14.

In a binary classification problem, each observation is labeled as either class 1 or class 2 by the model. For simplicity, positive will be used as the label for class 1 and negative will be used as the label for class 2. The true positive is then defined as the number of observations that are correctly labeled as positive, and the true negative is the number of observation that are correctly labeled as negative. The number of observations that are classified as positive by the model, but in reality belongs to the negative class, is referred to as the false positive. Equivalent, the false negative is the number of observation classified as false when the true label is positive. The relation between these terms is shown in figure 2.15.

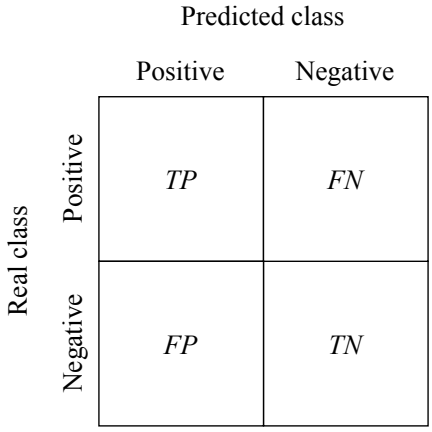


Figure 2.14: The typical construction of the confusion matrix. TP is the number of true positives, FP is the number of false positives, FN is the number of false negatives and TN is the number of true negatives.

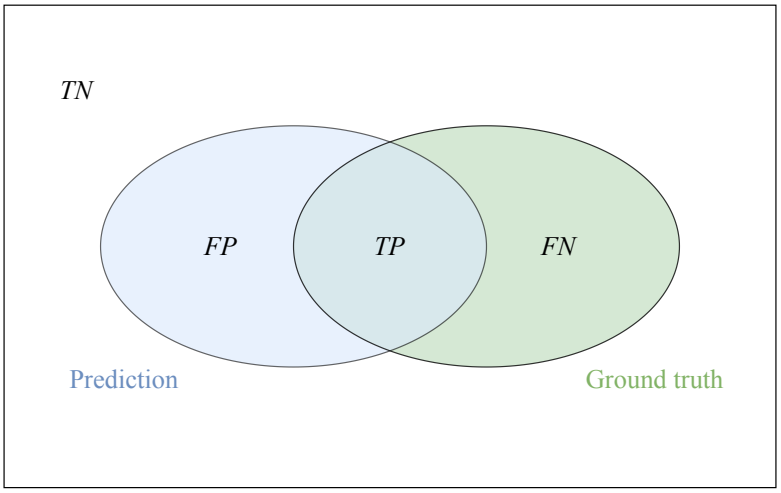


Figure 2.15: A Venn diagram showing the relation between true negative (TN), true positive(TP), false positive (FP) and false negative (FN) for a binary classification problem.

Most performance metrics are based on these values that are given in the confusion matrix. The error (ERR) and the accuracy (ACC) are defined as the number of misclassified predictions divided by the total number of predictions and the number of correct classified predictions divided by the total number of predictions respectively [31]. In terms of the values in the confusion matrix, they can be expressed as

$$ERR = \frac{FP + FN}{TP + TN + FP + FN} \tag{2.22}$$

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} = 1 - ERR. \tag{2.23}$$

These metrics give general information of the amount of misclassification made by the model, but can also be misleading in cases where the dataset is imbalanced. This means that the number of observations belonging to one class is much larger than the number of observations belonging to the other class. The model will hence be able to achieve high accuracy and low error by classifying all the observations as the class with the highest occurrence.

When dealing with imbalanced classes it can be useful to look at the true positive rate (TPR), also called recall, and the false positive rate (FPR) [31]. These are defined as

$$TPR = \frac{TP}{TP + FN} \tag{2.24}$$

$$FPR = \frac{FP}{TN + FP}. \tag{2.25}$$

The TPR gives the ratio between the number of observations correctly predicted as positive and the total number of positive observations, while the FPR gives the ratio between the number of observations that are misclassified as positive and the total number of negative observations. This gives a better understanding of the degree of error within each class.

Another metric that is used in image segmentation is precision (PRE). This gives the ratio between the number of true positive predictions and the total number of positive predictions as seen from the following equation.

$$PRE = \frac{TP}{TP + FP} \tag{2.26}$$

One of the most frequently used metric for medical image segmentation is the Dice similarity coefficient (DSC) [32]. DSC gives the spatial overlap between two segments [33], and it can be seen as a combination of the PRE and TPR as shown in the following equation.

$$DSC = 2\frac{PRE \times TPR}{PRE + TPR} = \frac{2TP}{2TP + FP + FN} \tag{2.27}$$

A complete overlap will result in $FP = FN = 0$ and hence $DCS = 1$. No overlap at all will give $TP = 0$ and therefore $DCS = 0$.

The DSC is a special case of the F_β-score which includes a weighting variable, β, to put different emphasis on the PRE and TPR. The general definition of the F_β-score is

$$F_\beta = \frac{1+\beta^2}{\frac{\beta^2}{TPR} + \frac{1}{PRE}} = \frac{(1+\beta^2)PRE \times TPR}{\beta^2 PRE + TPR}. \tag{2.28}$$

For the DSC β is set equal to 1, and it is therefore often referred to as the $F1$-score.

Chapter 3

Materials and methods

3.1 OxyTarget study

The images used in this thesis are from patients participating in the study "The OxyTarget study - Functional MRI of Hypoxia-Mediated Rectal Cancer Aggressiveness". The study includes a total of 192 patients diagnosed with rectal cancer and treated at Akershus University Hospital. They were enrolled in the study between October 2013 and December 2017. The Institutional Review Board and the Regional Committee for Medical and Health Research Ethics gave their approval for the study, and all patients participating gave a written informed consent [34]. The study aims to identify novel imaging biomarkers of hypoxia-induced rectal cancer aggressiveness, and this is important in order to predict patients with poor response to chemoradiotherapy and high risk of poor metastasis-free survival at the time of diagnosis [35].

A Phillips Achieva 1.5T system (Phillips Healthcare, Best, The Netherlands) was used to perform the MRI. The patients were given glucagon (1 mg/mL, 1 mL intramuscularly) and Buscopan (10 mg/mL, 1 mL intravenously) before the scanning to reduce bowel movement [34]. High-resolution T2 weighted images were acquired perpendicular to the tumor axis used for delineation, and with a field of view (FOV) equal to $180 \times 180\,\mathrm{mm}^2$ with 512×512 voxels in each slice. The size of the voxels was $0.3516 \times 0.3516 \times 2.5\,\mathrm{mm}^3$, and there was a 2.75 mm spacing between the slices. DWI with seven different b-values, 0, 25, 50, 100, 500, 1000 and $1300\,\mathrm{s/mm}^2$, were also acquired, and these images had a FOV of $160 \times 160\,\mathrm{mm}^2$ with 128×128 resolution. The voxel size for the DWI was $1.25 \times 1.25 \times 4\,\mathrm{mm}^3$, and they had a 4.3 mm spacing. The tumor delineations were made on the T2 weighted images by two different radiologists with 14 and 7 years of experience [34]. The DWI could be used for extra guidance.

Four of the 192 patients were excluded due to withdrawal from the study, and 19 patients lacked the histological confirmation of rectal cancer and were therefore also excluded. In addition to this, 75 other patients were excluded. 23 of these patients had non-consistent MRI sequence, 20 had dynamic images with poor quality, for six patients there were difficulties in the co-registration due to bowel movements or small tumor volume, and for 26 patients there were difficulties

encountered during the clinical MRI acquisition. Of the remaining 94 patients, images from 81 were used in this thesis. These patients had a complete set of images of acceptable quality, which includes T2 weighted images and DWI with all seven b-values.

3.2 Pre-processing

In the first part of this master thesis the raw data was sorted, co-registered, and resampled. The raw data consisted of the MR images stored as DICOM (Digital Imaging and Communications in Medicine) files and the delineations which were stored as NIfTI files. The NIfTI format is a common format used to store MRI data, and it is made up of a header containing metadata and the image data, similar to the DICOM format. The NIfTI format stores a 3D image in one file while DICOM, on the other hand, often has one file for every 2D slice. A newer DICOM format allows storing of a 3D image in one file, but for the OxyTarget images, each slice was stored as separate files.

Co-registration was done so that the T2 weighted images and the DWI aligned. Through the process of co-registration, the images are transformed into a common coordinate system so corresponding voxels represent homologous biological points [36]. This is done by defining a fixed image and then find a coordinate transform that deforms a moving image in such a way that it will match the fixed image. The transform parameters are optimized so that the difference between the two images is minimized. In this case, a rigid registration, which only allow for translation and rotation, was used. The DWI were first registered internally to correct for movements that occurred during the sequence. In the next step, the T2 weighted images were set as the fixed images and the DWI with b-value equal to zero acted as the moving images. The resulting transformation was then applied to the rest of the DWI. It was also created a dataset with only DWI, and for this dataset, the delineation masks, defined on the T2 weighted images, were transformed to match the DWI.

As a part of the co-registration, the DWI that were aligned with the T2 weighted images were interpolated using third-order B-spline interpolation and resampled to match the resolution of the T2 weighted images. Nearest Neighbour interpolation was used for the masks that were aligned with the DWI to give the masks the same resolution as the images. This resulted in two datasets, one containing T2 weighted images, DWI and masks with the resolution of the T2 weighted images, and one with only DWI and corresponding mask with the DWI resolution.

For two of the patients, the size of the T2 weighted images was larger than the standard. They had a size equal to 528×528 and 560×560 instead of the standard 512×512. The images of these two patients were therefore cropped to the standard size by removing voxels around the edges. There were also several patients where the voxel size slightly deviated from the standard, but these variations were so small that they were assumed negligible.

It was decided to remove image slices at the beginning and end of the image

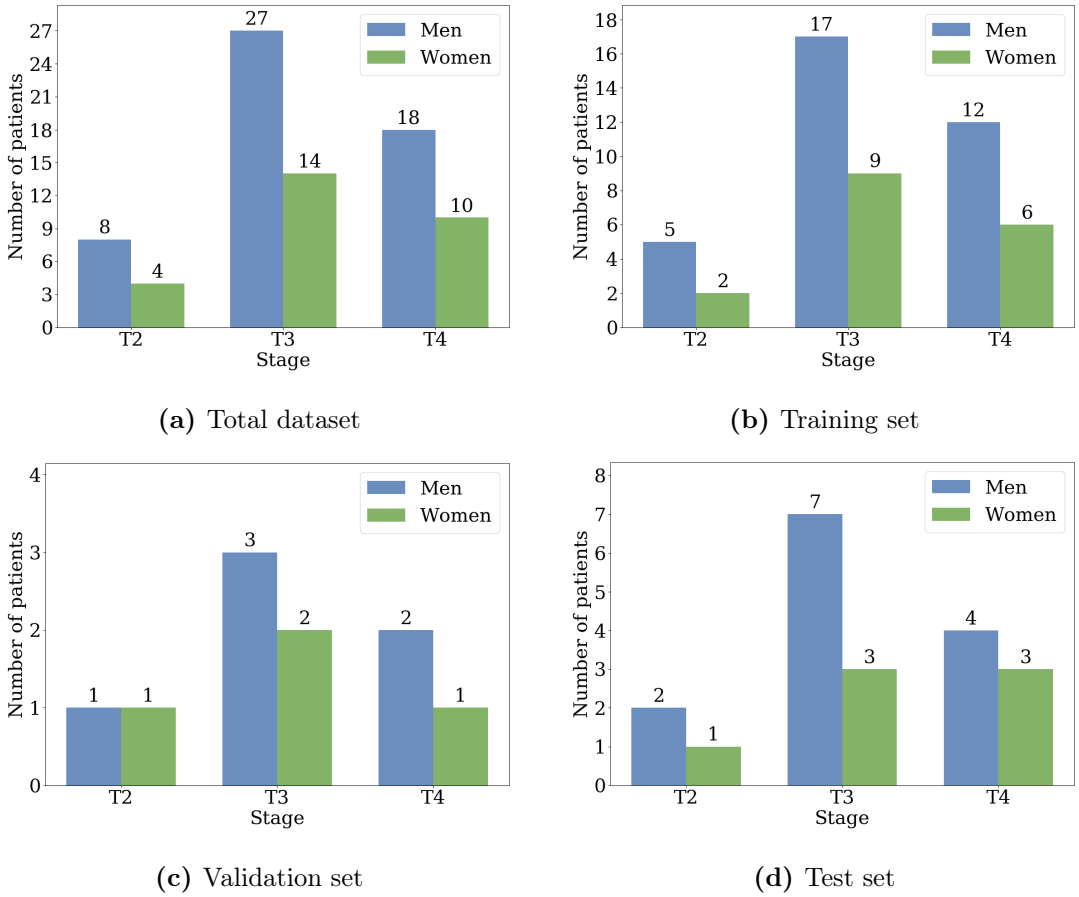
stacks where neither of the radiologists had delineated any tumor. This was done to reduce the size of the data, and because it was assumed that these slices would not contribute with any useful information to the model.

3.3 Train, validation and test split

Before using the data to train models, it was split into training, validation, and test sets. The training set is used to train the model, the validation set is used to evaluate the model and tune the model parameters, and the test set is used for the final test of the model. It is important to have both a validation set and a test set because when the model is tuned based on the performance on the validation set, information about the validation set leaks into the model [11]. The model can therefore easily end up being overfitted to the validation set. To evaluate how well the model generalizes it needs to be tested on a dataset that it has never seen before, the test set.

To make sure that all three groups contained a representative selection of the dataset, the data was stratified on a patient basis according to gender and disease stage. The stage is determined by how deep into surrounding tissues the tumor has grown, and it is defined as either T1, T2, T3, or T4. In the T1 stage the tumor has grown into the submucosa (the lining of the colon), in the T2 stage it has grown into the muscularis propria which is the thick layer of muscle outside the submucosa, in the T3 stage it has grown through the muscularis propria and into the fatty tissue surrounding the rectum (the mesorectum), and in the T4 stage the tumor has grown into the surface of the visceral peritoneum (T4a) or it has grown into surrounding organs (T4b) [37, 38]. The patients in this dataset had tumors in stage T2, T3, and T4, and consequently, the data could be divided into six different groups based on gender and stage.

Of the 81 patients, 51 (63%) was placed in the training set, 10 (12%) in the validation set, and 20 (25%) in the test set. The patients in these three sets were randomly picked within the six different groups, but the relation between the number of patients from each group was kept as similar as possible for the different sets. The distribution for the training, validation, and test set is shown in figure 3.1 and an overview of the number of patients and number of image slices in each of the sets is given in table 3.1.



(a) Total dataset

(b) Training set

(c) Validation set

(d) Test set

Figure 3.1: Histograms showing the number of patients within each group for the total dataset (a), the training set (b), the validation set (c) and the test set (d).

Table 3.1: Overview of the number of patients and number of image slices in the training, validation and test set.

	Number of patients	Number of image slices	
		T2 weighted	DWI
Train	51	962	580
Validation	10	188	114
Test	20	374	227
Total	81	1534	921

3.4 Data structure

Hierarchical Data Format version 5 (HDF5) is a convenient format for storing large numerical arrays of homogenous type, and it allows for the data to be organized in a hierarchical structure [39]. The data in an HDF5 file lies on disk until it is required which makes it possible to deal with datasets that exceed the RAM of the computer. The structure within an HDF5 file can be thought of as a folder structure, but instead of folders, there are groups. A file can contain several groups, and each group can contain several subgroups and datasets. A dataset is build up like an array, and the format allows for array slicing directly from disk. This structure makes the HDF5 format efficient for reading and writing to file [39].

The image dataset used in this thesis was saved as HDF5 files before it was used as input for the model, and the structure of the files is shown in figure 3.2. The data were divided into eight groups, five that belonged to the training set, one for the validation set, and two for the test set. These groups each contained six datasets. The dataset named "images" consisted of all the image slices for the patients in the group, "mask 1" contained the delineations corresponding to the image slices in "images" performed by radiologist 1, "mask 2" contained the delineation performed by radiologist 2, "mask intersection" contained the intersection of the two delineations, and "mask union" contained the union of the delineations. The last dataset, "patient id" consisted of the patient IDs for the patients in the group.

There were created three different HDF5 files. One that contained only the T2 weighted images, one with only DWI and one that contained both image types. In the file with the DWI alone, the image slices were stored with seven channels, one for each b-value. The file with both image types contained the DWI that were co-registered to the T2 weighted images, and each image slice consequently had eight channels. One channel for T2 weighted and seven for DWI.

In the process of saving the images as HDF5 files, they were downsampled to 64×64 voxels to speed up the training of the networks, and also make it less time consuming to debug.

3.5 Model parameters

The network architecture is based on a standard U-net as the one in figure 2.12, and an overview of the different layers is given in table 3.2. The convolution layers have a 3×3 kernel and same padding is applied. For these layers, the ReLU is used as the activation function. A window size of 2×2 and a stride of two is used in the max pooling layers. Skip connections occur between convolution layers in the contracting part and the expansion part of the network to restore spatial information.

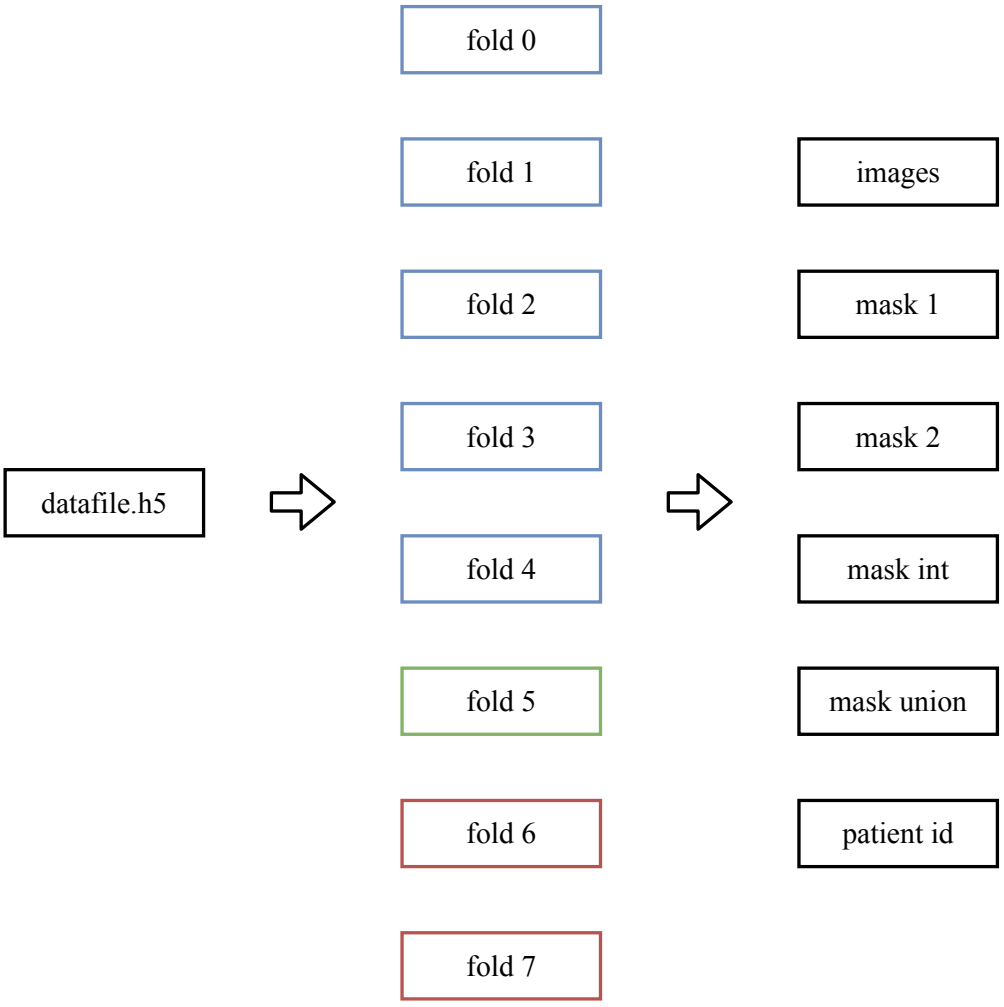


Figure 3.2: The structure of the HDF5 files. The file contains eight groups, and they each contain six datasets. The groups in blue belongs to the training set, the group in the green is the validation set and the two groups in red are the test set.

Table 3.2: The U-net architecture used in this thesis. All the convolution layers have a 3×3 kernel, same padding and ReLU as activation function. The window size for the max pooling layers is 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

The cross entropy was used as loss function in the model, but a problem with this loss function is that it is sensitive to class imbalance. In the dataset, there are a lot fewer tumor voxels than normal tissue voxels, and this imbalance might result in that the cross entropy loss function has a local minimum when most voxels are classified as normal tissue. To deal with this Milletari, Navab, and Ahmadi [40] defined a loss function based on the DSC that they called Dice loss. For a binary classification problem, it is defined as

$$J(\mathbf{p}, \mathbf{g}) = 1 - \frac{2\sum_i p_i g_i}{\sum_i p_i^2 + \sum_i g_i^2}, \tag{3.1}$$

where g_i is the ground truth for voxel i, and p_i is the predicted probability that voxel i will belong to the positive class.

Different models were trained by varying the input and the loss function, and the loss function was either the cross entropy (2.14), the Dice loss (3.1), or a modified version of the Dice loss where the squaring of the terms in the denominator is removed. The modified Dice loss is thus given by the following equation.

$$J(\mathbf{p}, \mathbf{g}) = 1 - \frac{2\sum_i p_i g_i}{\sum_i p_i + \sum_i g_i}, \tag{3.2}$$

Table 3.3 contains an overview of the different models that were trained, and all the hyperparameters for the models are listed in table 3.4. The Adam algorithm (2.19) was used to optimize the loss function during the training of the networks, and the learning rate was set to 0.0001. The training went on for 500 epochs, which corresponds to 500 iterations through the whole training set, and the weights in the network were updated after every 16th image. For all the models the union of the two delineations was used as ground truth.

Table 3.3: Overview of the different models that were trained. The input and the loss function were the parameters that varied.

Model number	Loss function	Input
1	Cross entropy	T2w images
2	Cross entropy	DWI
3	Cross entropy	T2w images, DWI
4	Dice	T2w images
5	Dice	DWI
6	Dice	T2w images, DWI
7	Modified Dice	T2w images
8	Modified Dice	DWI
9	Modified Dice	T2w images, DWI

Table 3.4: The different hyperparameters used for the models.

Activation function	ReLU
Optimizer	Adam
Learning rate	0.0001
Batch size	16
Epochs	500

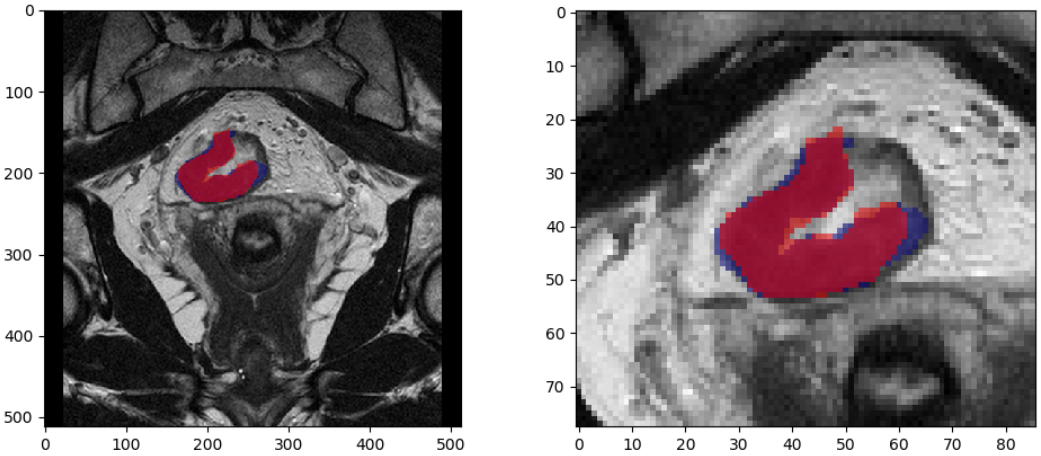
3.6 Model with linear support vector classifier

In her project thesis, the author developed an automatic segmentation model based on the linear support vector classifier. For this model, the same dataset was used but the images were pre-processed differently, and this pre-processing is visualized in figure 3.3. The T2 weighted images and the DWI were co-registered and resampled to isotropic voxels (1 mm × 1 mm × 1 mm). To reduce the size of the data and obtain a more balanced dataset, the images were cropped. This was done so that there was a 20 mm margin outside the largest extent of the tumor amongst all the slices. After the resampling and cropping, both the T2 weighted and the DWI for a given patient contained the same field of view (FOV) and equal voxel size. The processed images were then saved as NIfTI files.

In order to use the image data as input for the model, it was first structured in matrices where each element corresponded to the intensity value of a voxel. Each matrix corresponded to one of the patients in the data set, and figure 3.4 illustrates how these matrices were constructed. For each patient, the NIfTI images were loaded into the program and converted to three-dimensional arrays. These arrays were then flattened to one-dimensional arrays and set as columns in the matrix. The matrix had between 170000 and 1380000 rows, and this number corresponded to the number of voxels in the images of the patient. Normalization to a mean of zero and a standard deviation of one was performed on the data to compensate for varying intensities between images, and the data from the DWI for each patient were normalized together to keep the relation between the different b-values. The matrices were stored in a dictionary with the given patient names as keys to make it convenient to access data for different patients later.

To train and evaluate the classification model, the data needed to be divided into training and test sets. Leave-out-one cross-validation was used to perform this task, and the principle of this method is illustrated in figure 3.5. Each patient was in turn used as the test set while the rest of the patients were used for training. In this way, it is possible to see how the model performs on all the patients.

Since the images were cropped with a 20 mm margin outside the tumor volumes, it might be that this exceeded the original FOV on some of the images. This will lead to missing values, and the voxels are stored with an intensity value equal to zero. In the training set, these values were removed, so that they would not affect the model. For the test set, the indexes of these values were stored, and in that way, they could be corrected after the prediction was made.



(a) Original image **(b)** Processed image

Figure 3.3: A T2w image before (a) and after (b) resampling to isotropic voxels and cropping. The blue and red areas are the manual delineations of the tumor volume.

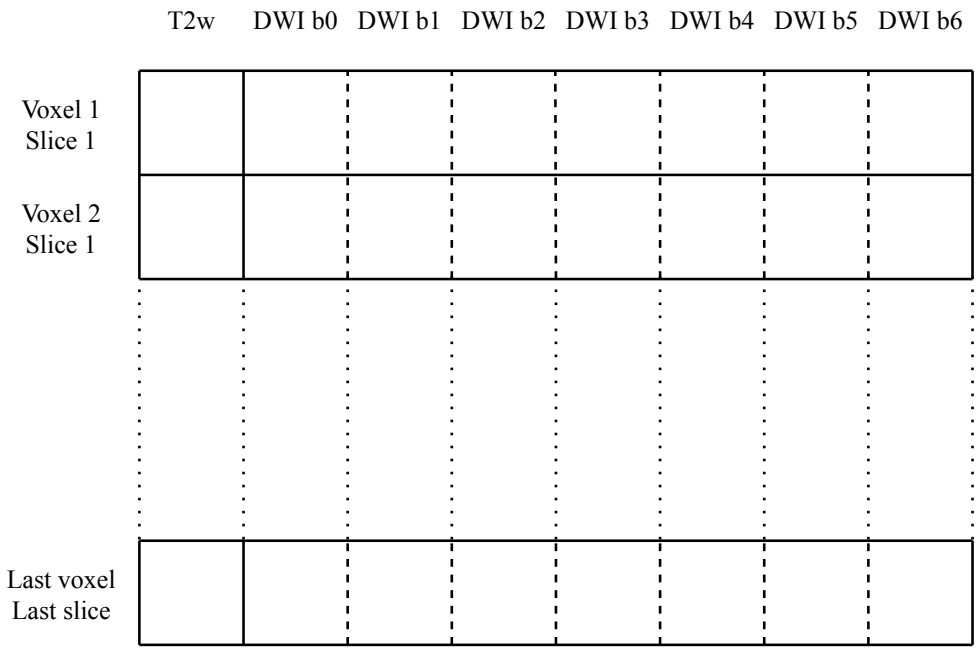


Figure 3.4: The structure of the matrix constructed from the image data for each patient. T2w refers to the T2 weighted image, DWI referrers to diffusion weighted images and b0, b1, b2, b3, b4, b5 and b6 stands for the different b-values.

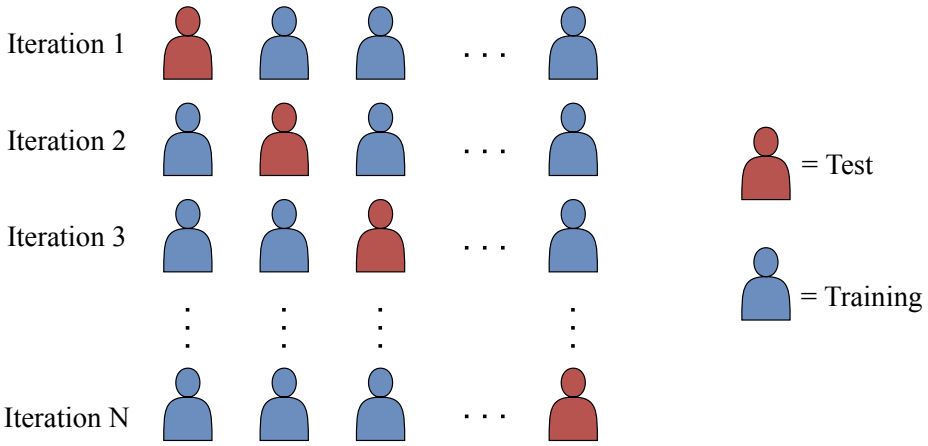


Figure 3.5: Illustration of the leave-out-one cross validation for a cohort of N patients. Each patient is in turn used as test set (red) while the rest is used as the training set (blue). This results in a total of N iterations.

Even though the images were cropped, the dataset was still very imbalanced. In this case, the data set contained 94% non-tumor voxels, and only 6% tumor voxels. Training on an imbalanced data set can cause problems because most classification models seek to maximize the accuracy and reduce the error. If the number of observations in one class is much larger than the other, the model might make a prediction boundary that classifies almost every new observation as the class with the largest number of observations. From the observations the model has seen, one class will be the correct prediction for most of the observations. To account for this, the data in the training set was re-sampled so that the training set contained an equal number of tumor voxels and non-tumor voxels. This was done by counting the number of tumor voxels and then randomly select the same amount amongst the non-tumor voxels for each patient.

3.7 Code and software

SimpleElastix was utilized to co-register and interpolate the images. This library acts as a binding to the Elastix toolbox, which contains a collection of medical image registration algorithms [36]. Ph.D. student Franziska Knuth wrote the Python code for this pre-processing step.

The code used for creating and running the models was developed by Ngoc Huynh Bao, a master student from the Norwegian University of Life Sciences (NMBU). It is a CNN framework for Python developed especially for automatic delineation of cancer tumors. The framework, *deoxys*, is based on Keras, a deep learning API that runs on top of the machine learning platform TensorFlow 2.0 [41]. Keras has implementations of building blocks that are essential for deep learning models, like different layers, activation functions, and optimizers. The *deoxys* framework code can be accessed from the GitHub repository `https://`

`github.com/huynhngoc/deoxys`.

To create models with the framework, the CNN architecture and the hyperparameters have to be defined in a JSON (JavaScript Object Notation) file, and the data must be structured in an HDF5 file. The Python scripts for generating the HDF5 files and running the CNN models, as well as the JSON files, were written by Yngve Mardal Moe, head engineer at NMBU, and they are available from the GitHub repository `https://github.com/yngvem/ntnu-analysis`.

3.7.1 Linear support vector classifier

For the SVC model, data processing and machine learning were performed in Python version 3.7.5. The main libraries used were NumPy, SimpleITK, Scikit-learn, and Dask. NumPy is used for creating n-dimensional array objects, and it is a fundamental package for scientific programming in Python [42]. SimpleITK is a simplified version of the Insight Segmentation and Registration Toolkit (ITK) that is used for image analysis [43]. With SimpleITK one can easily load different images types to Python and then converting these images to NumPy arrays. It also provides a function that can generate images from NumPy arrays. The Scikit-learn library includes a large collection of supervised and unsupervised learning algorithms for machine learning, and it works well with NumPy arrays [44]. Dask is a library for parallel computing in Python [45]. This makes it possible to work with large data that exceeds the RAM of the computer. Dask arrays are built up of several NumPy arrays, and the Dask library includes functions for machine learning where one can combine the Dask arrays with some of the learning algorithms from Scikit-learn.

The machine learning algorithm used to create this model was the `sklearn.linear_model.SDGClassifier` from the Scikit-learn library together with the wrapper function `dask_ml.wrappers.Incremental` from the Dask library. This allows for out-of-core learning, which is convenient when working with large data sets. The `sklearn.linear_model.SDGClassifier` optimizes the loss function for the linear support vector classifier, given in equation (2.21), with the stochastic gradient descent method. The complete code is available from the GitHub repository `https://github.com/elinefs/prosjektoppgave`.

3.8 Analysis of model performance

The predictions made by the U-net models were given as heatmaps where each voxel got a score that indicated whether the model found it likely that the voxel was a tumor voxel or not. A score close to one indicated tumor while a score close to zero indicated non-tumor. To generate binary prediction masks, a threshold of 0.5 was applied to the heatmaps. For the SVC model, the predictions were given as binary masks, so no threshold was needed.

The model performance was evaluated by calculating the DSC (3.1) between the prediction and the union of the two manual delineations for each patient in the validation set. The average DSC was used to compare the different models.

Ideally, the best performing model should also have been evaluated on the test set to see how well it generalizes, but since several modifications could be done to further improve the performance, it was decided to save the test set until a more precise model was obtained. The modification will be elaborated in the discussion chapter under the section about further work.

Chapter 4

Results

The different U-net models that were created had a performance that ranged from a DSC of 0.58 to 0.67, calculated as the mean DSC for the patients in the validation set. The DSC on a patient basis ranged from about 0.21 and up to 0.85. In figure 4.1 the training and validation curves for the models trained with both image types are presented. As one can see, there are slight differences between the different loss functions, but they all stabilize around the same values. The binary F_β term refers to the DSC calculated on all the image slices in the dataset combined, with a threshold equal to 0.5. The training and validation curves for the rest of the models can be found in Appendix A. In the following sections, the performance of the different models will be presented in detail.

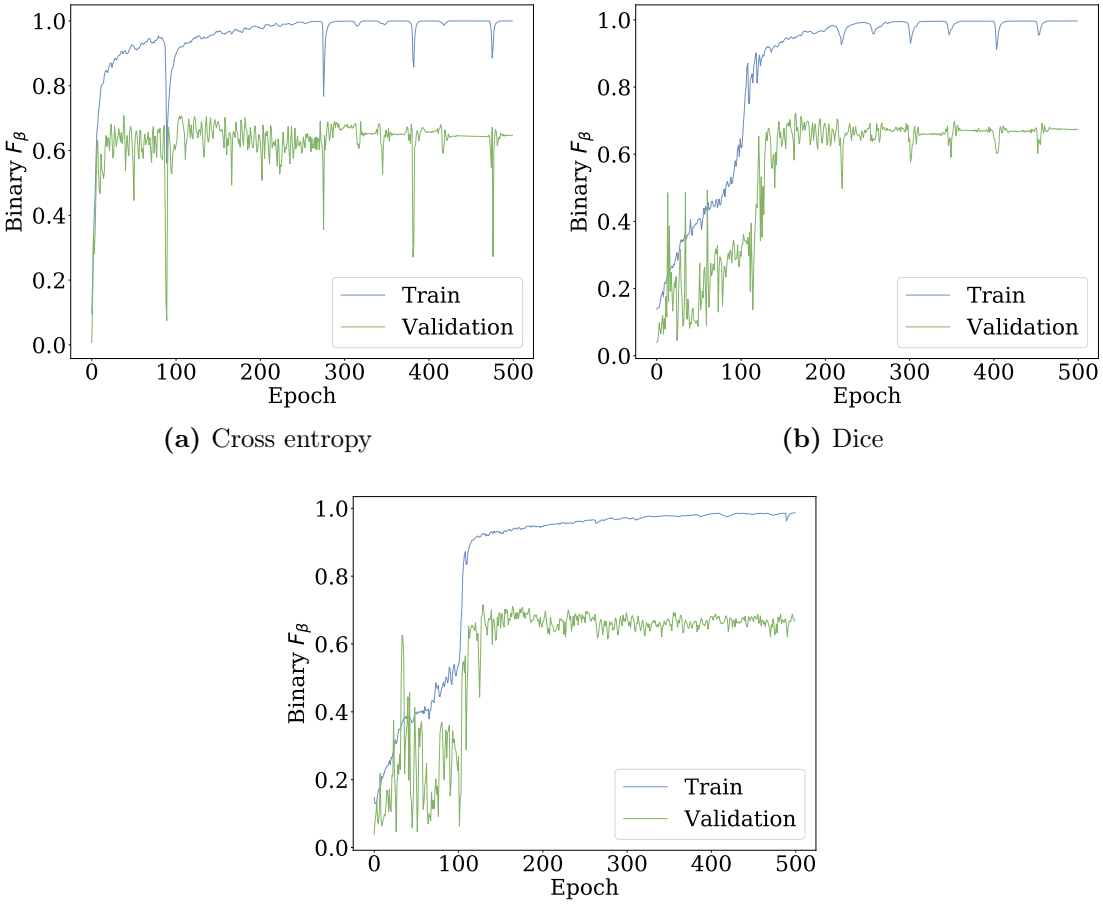
4.1 Effect of input images

4.1.1 T2 weighted images

The models trained with only the T2 weighted images resulted in a mean DSC of 0.61 with cross entropy loss, 0.58 with Dice loss, and 0.67 with the modified Dice loss. The performance for each of the patients in the validation set is shown in the scatter plot in figure 4.2. In this plot, the DSC between the two delineations, the interobserver variation, is marked with a black line for each patient.

From figure 4.2 one can see that there is only a small difference between the three models, with a few exceptions. The model with the Dice loss function seems to be performing poorly on patient 72, and also patient 125. It is also worth noticing that all three models seem to be struggling with patient 124.

When compared with the interobserver variation, the performance for five of the patients is fairly close to this or even higher. For patient 88 and 157, the DSC for the interobserver variation is a bit lower than for the rest of the validation set, which indicates a larger disagreement between the two radiologists. These two patients have an interobserver DSC close to 0.6, while the rest have a DSC around 0.8.



(a) Cross entropy

(b) Dice

(c) Modified dice

Figure 4.1: Training and validation curves for the models with T2 weighted images and DWI as input. (a) corresponds to the model with cross entropy loss (2.14), (b) corresponds to the model with dice loss (3.1), and (c) corresponds to the model with modified dice loss (3.2). The binary F_β is the DSC calculated on all image slices in the dataset combined, with a threshold equal to 0.5. The green curves represent the validation set while the blue curves represent the training set.

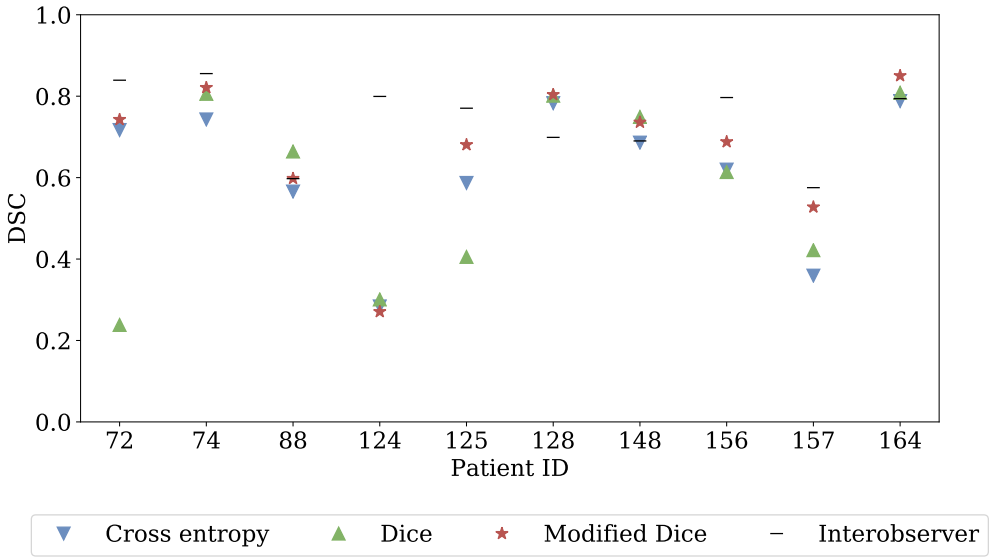


Figure 4.2: The DSC for the patients in the validation set for the models with T2 weighted images as input. The blue triangles represent the model with the cross entropy loss, the green triangles represent the model with the Dice loss, while the red stars represent the model with the modified Dice loss. The black horizontal lines are the interobserver variation for each patient.

4.1.2 Diffusion weighted images

The models trained with DWI had a DSC of 0.63, 0.65, and 0.66 for cross entropy loss, Dice loss, and modified Dice loss, respectively. Figure 4.3 visualize the performance for each of the patients in the validation set for these models.

Here one notices that there are relatively small variations between the different loss functions for most of the patients. Compared to the models trained with T2 weighted images the models with DWI gives a better prediction for patient 124 and 125, while they perform worse for patient 88.

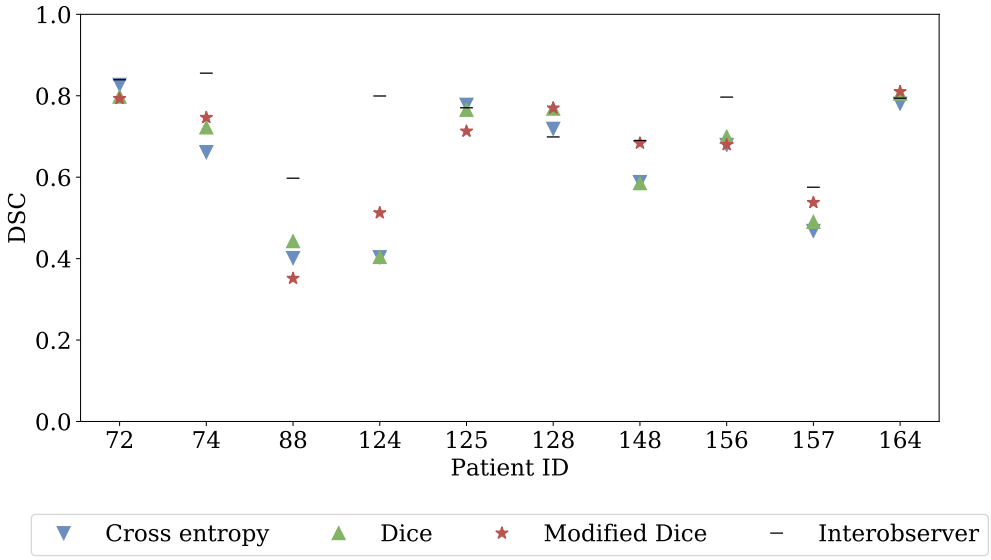


Figure 4.3: The DSC for the patients in the validation set for the models with DWI as input. The blue triangles represent the model with the cross entropy loss, the green triangles represent the model with the Dice loss, while the red stars represent the model with the modified Dice loss. The black horizontal lines are the interobserver variation for each patient.

4.1.3 Combined T2 weighted and diffusion weighted images

The third input that was tested was the T2 weighted images and DWI combined. For this input the model with the cross entropy loss gave a DSC equal to 0.66, the Dice loss resulted in a DSC of 0.67 and the modified Dice loss gave a DSC of 0.67. In figure 4.4 the DSC for the patients in the validation set is shown for the models that have this combined input.

From the plot in figure 4.4 on can see that, in this case, changing the loss function has very little impact on the results. Other than that, the performance is similar to the performance of the models that had only one of the image types as input. One might expect that more input data to the model would yield improved results, but this does not seem to be the case here.

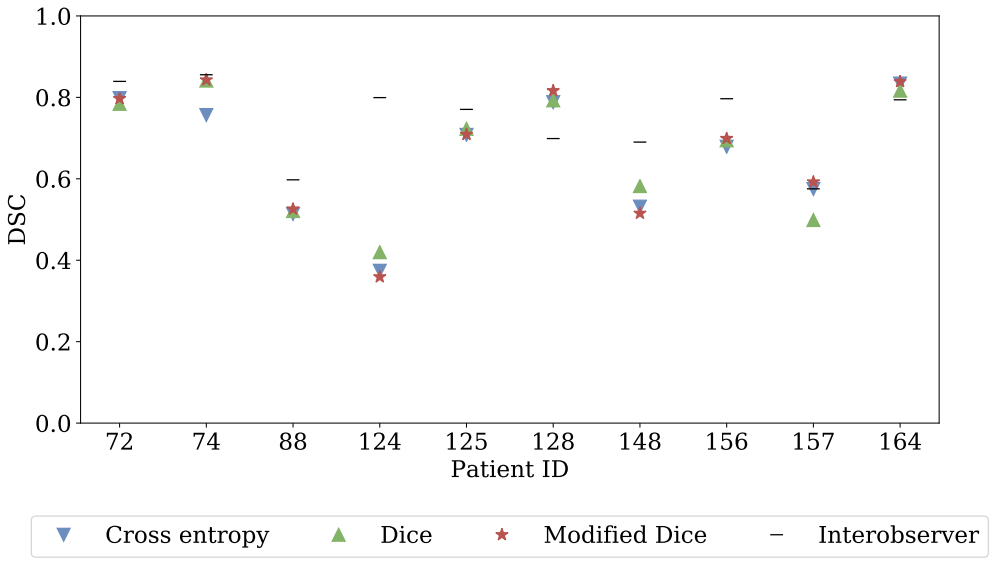


Figure 4.4: The DSC for the patients in the validation set for the models with T2 weighted images and DWI as input. The blue triangles represent the model with the cross entropy loss, the green triangles represent the model with the Dice loss, while the red stars represent the model with the modified Dice loss. The black horizontal lines are the interobserver variation for each patient.

4.2 Best performing model

Table 4.1 contains the mean DSC and standard deviation for all the different models. From this one can see that the model trained with T2 weighted images and the modified Dice loss function results in the best performance, but the differences are marginal.

Figure 4.5 shows a boxplot with the different models, and here one can also see that the model with T2 weighted images and the modified Dice loss performs best. This is also the model with the least spread. All the patients in the validation set have a DSC between 0.52 and 0.85, except one outlier with a DSC of 0.27. The exact DSC for each of the patients in the validation set with the best performing model is given in table 4.2. Some of the other models yield better performance for some of the patients, but this model has the highest average.

Table 4.1: An overview of the mean DSC and corresponding standard deviation for the validation set with the different U-net models.

Loss function	Input	DSC
Cross entropy	T2w	0.613 ± 0.164
Cross entropy	DWI	0.631 ± 0.150
Cross entropy	T2w, DWI	0.656 ± 0.143
Dice	T2w	0.581 ± 0.210
Dice	DWI	0.648 ± 0.146
Dice	T2w, DWI	0.667 ± 0.143
Modified Dice	T2w	0.672 ± 0.164
Modified Dice	DWI	0.660 ± 0.140
Modified Dice	T2w, DWI	0.670 ± 0.157

Table 4.2: The DSC for the patients in the validation set obtained with the U-net model that took T2 weighted images as input and used the modified Dice loss function.

Patient ID	DSC
Oxytarget 72	0.742
Oxytarget 74	0.821
Oxytarget 88	0.598
Oxytarget 124	0.271
Oxytarget 125	0.681
Oxytarget 128	0.804
Oxytarget 148	0.736
Oxytarget 156	0.688
Oxytarget 157	0.528
Oxytarget 164	0.850

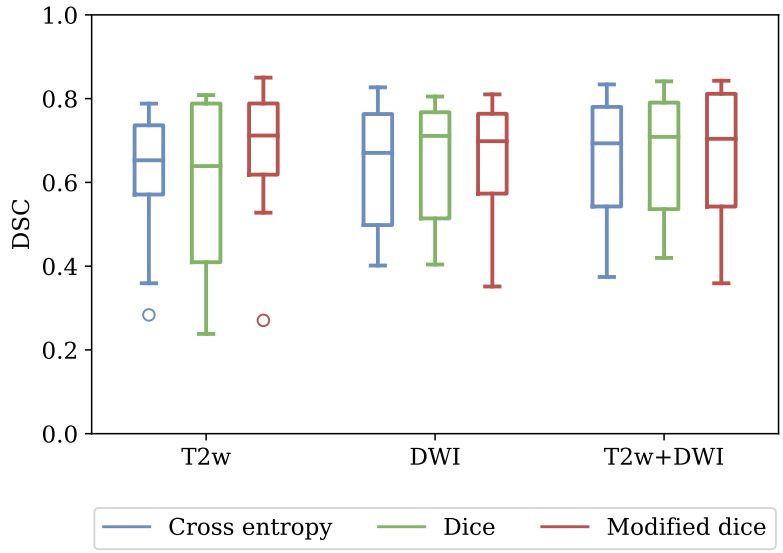
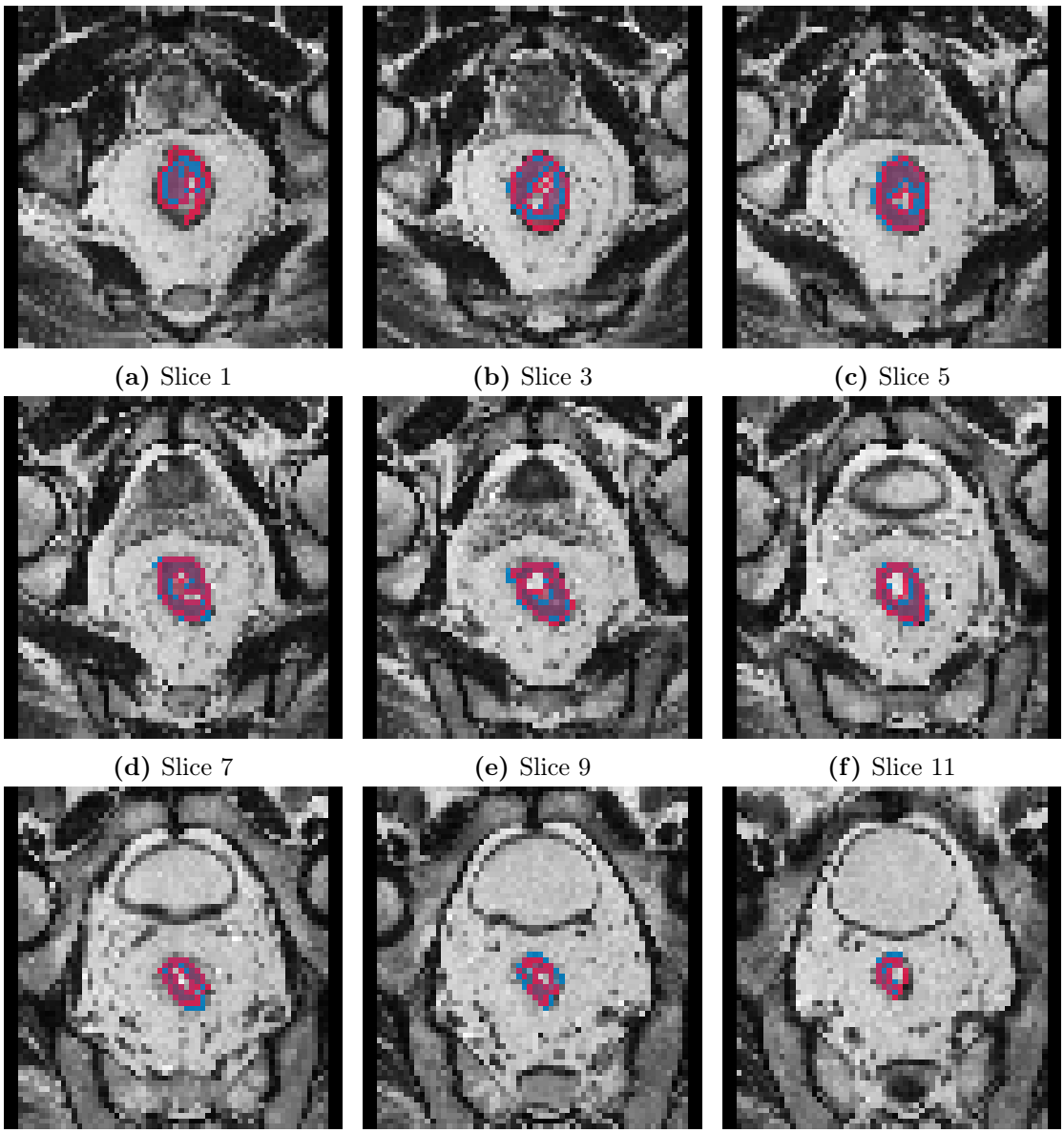


Figure 4.5: Boxplot with the performance of all the different U-net models on the validation set. The blue boxes represent the models with the cross entropy loss, the green boxes represent the models with the Dice loss and the red boxes represent the models with the modified Dice loss. The horizontal line within each box is the median DSC for the model, and 50% of the patients should have a DSC that lies within the box. The criterion for the outliers (circles) is that they are more than 1.5 times the box height away from the box edge.

In figure 4.6 the delineations made by the model are shown together with the union of the manual delineations made by the radiologists for some image slices from Oxytarget patient 164. This is the patient with the highest DSC, and the delineations made by the model are, therefore, relatively accurate. Figure 4.7, on the other hand, shows the predicted and manual delineations for Oxytarget patient 124 which has the lowest DSC in the validation set. Here the model did not perform well, and there is little overlap between the predicted and manual delineations. The delineations on images from the rest of the patients in the validation set can be found in Appendix B

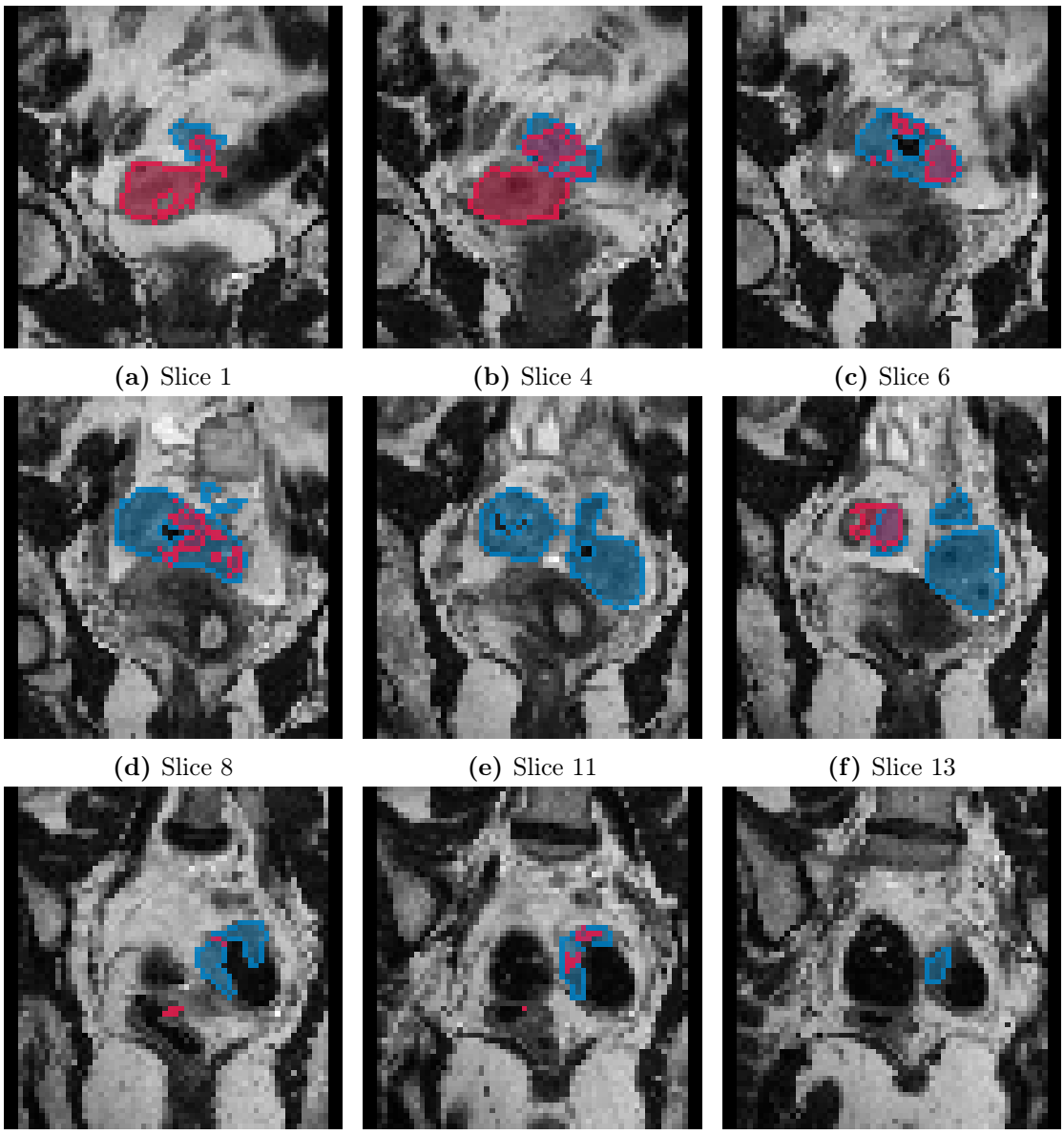


(a) Slice 1 **(b)** Slice 3 **(c)** Slice 5

(d) Slice 7 **(e)** Slice 9 **(f)** Slice 11

(g) Slice 13 **(h)** Slice 15 **(i)** Slice 17

Figure 4.6: The delineation predicted by the best U-net model (red) and the union of the delineations made by the two radiologists (blue) on a selection of T2 weighted image slices from OxyTarget patient 164. The DSC for this patient is 0.85.



(a) Slice 1 **(b)** Slice 4 **(c)** Slice 6

(d) Slice 8 **(e)** Slice 11 **(f)** Slice 13

(g) Slice 16 **(h)** Slice 18 **(i)** Slice 20

Figure 4.7: The delineation predicted by the best U-net model (red) and the union of the delineations made by the two radiologists (blue) on a selection of T2 weighted image slices from OxyTarget patient 124. The DSC for this patient is 0.27.

4.3 Threshold

For the heatmaps that were outputted from the U-net models, the threshold was set to 0.5, but this threshold is not necessarily the one that yields the best performance. To explore this, the performance in terms of average DSC on the validation set was plotted against the threshold. Figure 4.8 shows this curve for the model with T2 weighted images and cross entropy loss, and the light blue area represents the standard deviation. Appendix C contains the threshold curves for the other models.

The threshold curve is relatively flat, which means that the performance will not change significantly by changing the threshold. There is however a small peak very close to zero. The performance for all the models was therefore also calculated with a threshold of 0.01, and the results are displayed in table 4.3. From this table, one can see that changing the threshold makes the largest difference for the models with the cross entropy loss and the models with Dice loss that includes the T2 weighted images. For the models with the modified Dice loss, the change is very small.

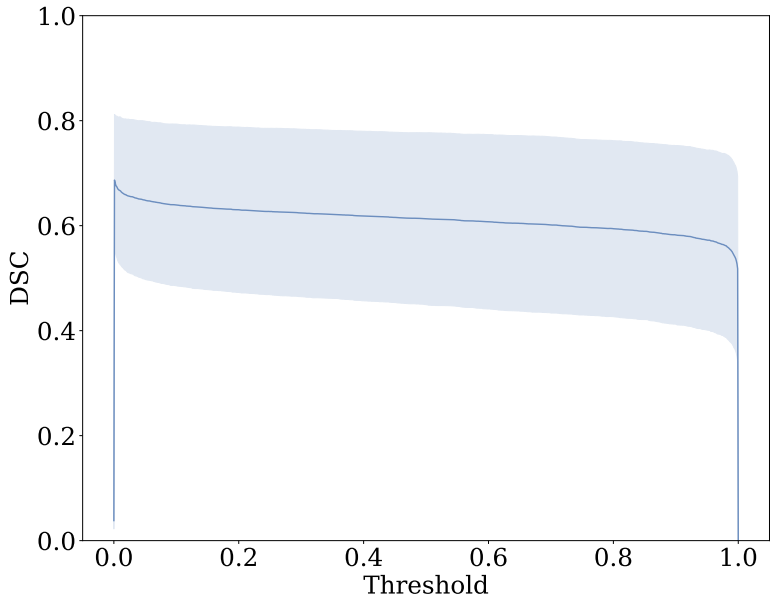


Figure 4.8: The average DSC plotted against the threshold for the U-net model which took T2 weighted images as input and used the cross entropy loss function. The light blue area corresponds to the standard deviation.

Table 4.3: The mean DSC and the corresponding standard deviation for all the U-net models with threshold equal to 0.5 and 0.01. The percentage performance increase obtained by changing the threshold from 0.5 to 0.01 is given to the far right.

Loss function	Input	Threshold= 0.5	Threshold= 0.01	Δ
Cross entropy	T2w	0.613 ± 0.164	0.667 ± 0.141	8.8%
Cross entropy	DWI	0.631 ± 0.150	0.669 ± 0.103	6.0%
Cross entropy	T2w, DWI	0.656 ± 0.143	0.699 ± 0.109	6.6%
Dice	T2w	0.581 ± 0.210	0.621 ± 0.192	6.9%
Dice	DWI	0.648 ± 0.146	0.675 ± 0.090	4.2%
Dice	T2w, DWI	0.667 ± 0.143	0.712 ± 0.106	6.7%
Modified Dice	T2w	0.672 ± 0.164	0.684 ± 0.154	1.8%
Modified Dice	DWI	0.660 ± 0.140	0.668 ± 0.129	1.2%
Modified Dice	T2w, DWI	0.670 ± 0.157	0.687 ± 0.142	2.5%

4.4 Comparison with the support vector classifier

Compared to the SVC model, the U-net models performed considerably better. The best results for the SVC model were obtained when both T2 weighted images and DWI were used as input and the union of the delineations was used as the ground truth. This SVC model gave an average DSC of 0.48, while the average DSC with the best U-net model was 0.67. This corresponds to a performance increase of 40%. In figure 4.9 the DSC values for patients in the validation set is shown for both models together with the interobserver variation. One can observe that the U-net model gives a significantly higher DSC for most of the patients, and patient 124 is the only one where the SVC model gives a better result. For patients 72 and 157 there is not a large difference between the two models.

The boxplot in figure 4.10 gives a more clear picture of the overall performance. The U-net model has a performance that is relatively close to the interobserver variation which has an average DSC of 0.74 for the patients in the validation set. The average DSC between the two delineations on the whole data set is 0.78.

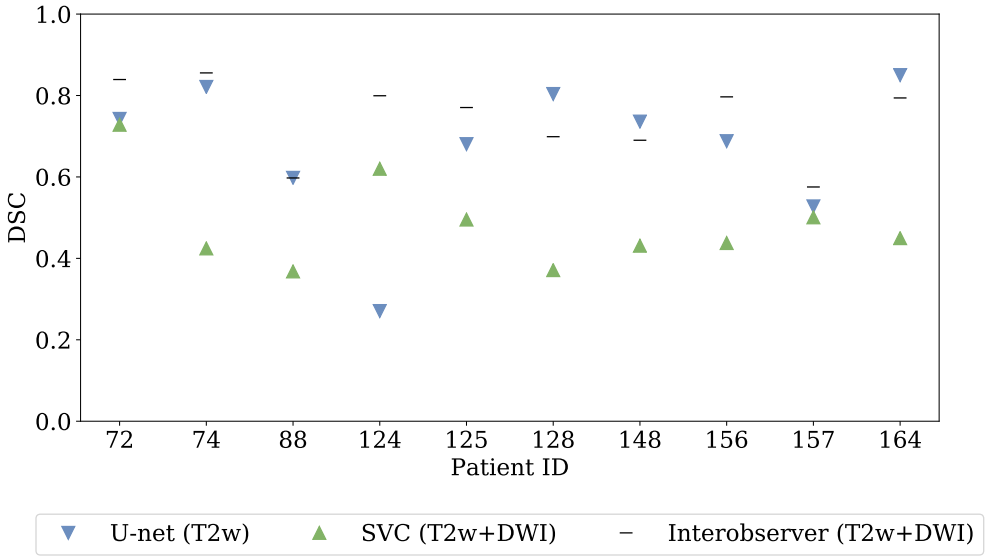


Figure 4.9: The DSC for the patients in the validation set for the SVC model and the U-net model with T2 weighted images as input and the modified Dice loss as the loss function. The blue triangles represent the U-net model while the green triangles represent the SVC model. The black horizontal lines are the interobserver variation for each patient.

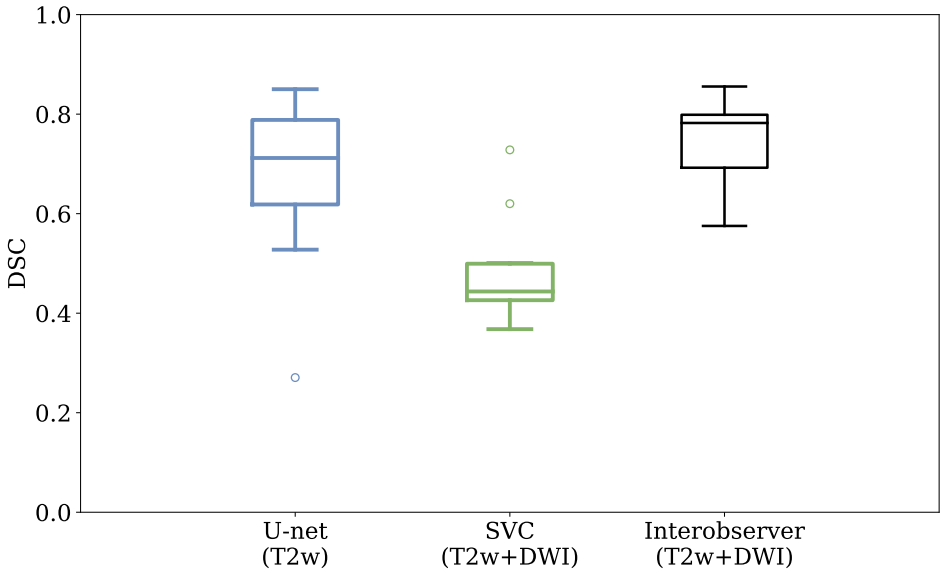


Figure 4.10: Boxplot with the performance of the SVC model, the highest performing U-net model and the interobserver variation on the validation set. The blue box represents the U-net model, the green box represents the SVC model and the black box represents the interobserver variation. The horizontal line within each box is the median DSC for the model, and 50% of the patients should have a DSC that lies within the box. The criterion for the outliers (circles) is that they are more than 1.5 times the box height away from the box edge.

Chapter 5

Discussion

The U-net models for segmentation of rectal tumor volume from MR images developed as a part of this master thesis showed some promising results, but with an average DSC ranging from 0.58 to 0.67 for the different models, there is still room for improvements. This chapter will consist of a discussion of the model performance, challenges regarding the input image data, comparisons with similar models, and a discussion of the clinical impact of such a model. This is followed by a section containing suggestions for further work.

5.1 Model performance

The training and validation curves in figure 4.1 and in Appendix A show that the models trained with cross entropy loss almost immediately went up to relatively high performance, before a slow increase until about 300 epochs into the training. After that, both the training and validation curve is more or less flat. One can observe some downward spikes on the curves, and this might suggest that the models are very sensitive to some specific weight updates. For the models trained with the Dice loss, it takes a little over 100 epochs to reach a score of over 0.8 for the training set. When the training curve reaches 1.0, the curves flatten out, and also here some downward spikes can be observed, but they are much smaller than for the models with the cross entropy loss function. The curves for the models trained with the modified Dice loss are relatively similar to the ones for the models with the Dice loss, which makes sense since the two loss functions are quite similar and share some of the same properties. For all the models the validation curve is stabilizing around 0.6, except for the one trained with T2 weighted images and Dice loss. Here the curve is more uneven and does not seem to have stabilized completely after 500 epochs. By training this model for a larger number of epochs the performance on the validation set might have increased further. One other thing to notice about these curves are the gaps between the training and validation curves. These gaps indicate that not a lot of information from the validation set has leaked into the models. One can, therefore, speculate that the performance on the validation set is close to what would have been the case for the test set if the model was tested on this. The size of the gap also indicates that the models

are well fitted to the training data, but are struggling a bit to generalize to the validation data.

The fact that the models are very well fitted to the training set and not to the validation set can suggest overfitting. This is a common problem for complex models like deep neural networks. One way to deal with overfitting is to reduce the size and complexity of the network. Another option is to add weight regularization and/or layers with dropout. A larger training set would also contribute to a lower risk of overfitting, and this could be obtained by adding data augmentation. Data augmentation is transforming an image by processes like rotation, flipping, and cropping in order to create several versions of the image [46].

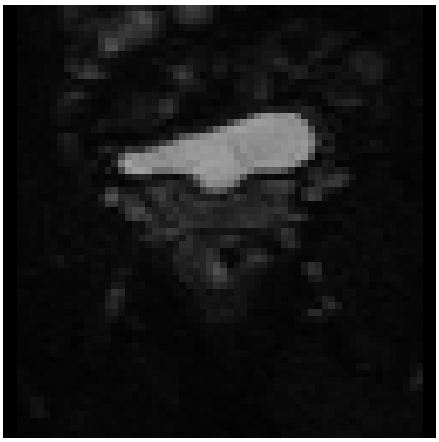
As seen from table 4.1, there are some variations in the model performance when changing the loss function and the input to the models. The choice of loss function made the most impact for the models with only T2 weighted images as input, and the models trained with the modified Dice loss gave the highest performance for all three input variations. However, since most of the models have relatively similar performance, the loss function and input that gives the best performance might change if the models are tested with another dataset.

The best U-net model gave a lower DSC than the interobserver variation between the two manual delineations for the OxyTarget data. Still, less than 20% performance increase is needed before the model is as good as the interobserver variation. This is relatively promising considering that very little pre- and post-processing are implemented in the current model.

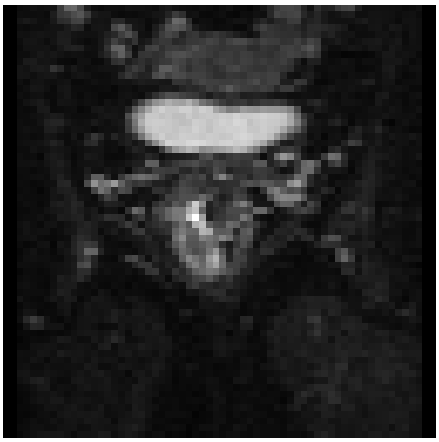
For all the U-net models patient 124 had a relatively low DSC, which can be seen from figure 4.2, 4.3 and 4.4. In figure 4.7 one can see that the delineated tumor is quite large and complex for this patient, and that is a possible reason for why the models have problems with the prediction. Another thing one can notice is that the images for this patient look quite different from the images in figure 4.6 from patient 164. This is due to a different image orientation, and not having a consistent image orientation could also pose a problem for the models.

The models that only take the DWI as input had a low performance for patient 88, and this is the case for all three models, as seen in figure 4.3. After comparing the DWI for patient 88 to the DWI for the other patients, it was discovered that the images for patient 88 are slightly darker and that makes it hard to detect structures. In figure 5.1, a DWI with b-value $0\,\mathrm{s/mm^2}$ is shown for patient 88 and patient 72 together with a histogram of the voxel intensities for these images. Here one can see the difference in brightness. A way to account for this would be to do a patient wise normalization of the image intensities before the images were taken as input to the models. This was done for the SVC model, but not for the U-net models.

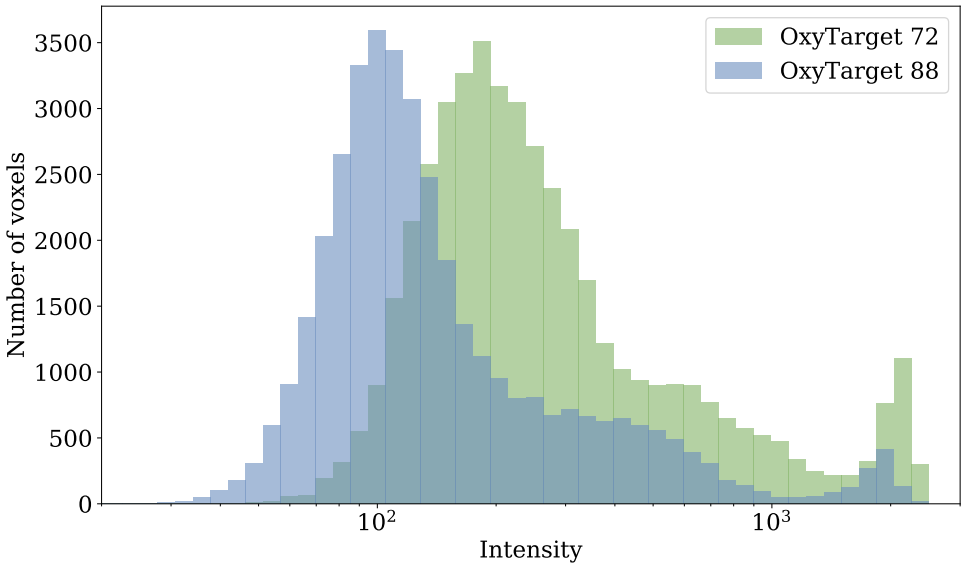
Looking at figure 4.2 one observes that patient 72 has low DSC when the Dice loss function is used, but not for the other two models with T2 weighted images as input. A closer look at the predictions for this patient revealed that the model with Dice loss does not seem to recognize the tumor in many of the image slices, and instead wrongly predicts some voxels at the black edge to the left of the image as tumor. An example of this is given in figure 5.2. This may also be the case



(a) OxyTarget 88



(b) OxyTarget 72



(c) Histogram

Figure 5.1: DWI with b-value $0\,s/mm^2$ for OxyTarget patient 88 (a) and OxyTarget patient 72 (b). On can observe that image (b) is brighter than image (a), and more details are visible in this image. The histogram (c) shows the distribution of intensities for the two images.

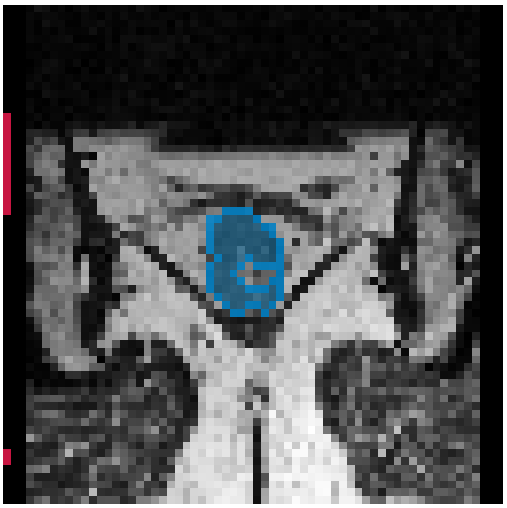
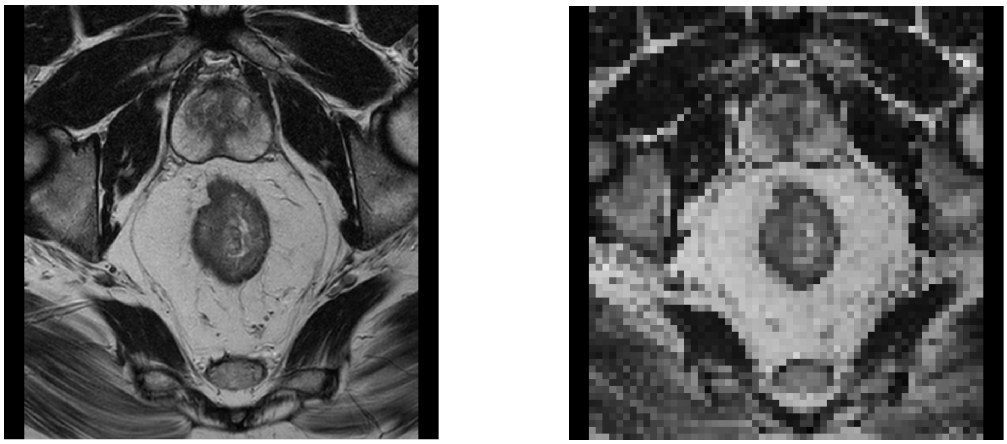


Figure 5.2: The predicted delineation (red) resulting from the model with T2 images as input and the Dice loss function, together with the ground truth (blue), on a selected T2 weighted image slice from OxyTarget patient 72. Here the model failed to locate the tumor.

with other patients. It is no obvious explanation for why this happens with this particular model, but it might be possible to avoid by cropping the images.

One might have expected that training models with both T2 weighted images and DWI would give a higher performance compared to models that were only trained on one image type. This does not seem to be the case here, and the best model was trained with only T2 weighted images, as can be seen from table 4.1 and figure 4.5. There is not a significant difference in the performance of the different models, and including both image types seem to result in models with an average performance of the models trained with only T2 weighted images and the models with only DWI. This can mean that there are few relations between the T2 weighted images and the DWI that the U-net is able to detect, and that will give valuable information regarding the classification.

As shown in section 4.3, the choice of threshold in the heatmaps to create binary prediction masks did not have a large impact on the results. This indicates that the models mostly set a score that is either close to zero or close to one, and very few voxels get a score in between. The fact that it is a slight increase in the model performances by setting a low threshold (0.01), makes it reasonable to think that when the models are not certain of which class the voxel belongs to, they will rather classify the voxel as non-tumor than tumor. This often results in that the models predict smaller tumor volumes than what is the case according to the manual delineations.



(a) Original image **(b)** Downsampled image

Figure 5.3: A T2 weighted image slice from OxyTarget patient 164 with the original 512×512 resolution (a) and the downsampled 64×64 resolution (b).

5.2 The images

There are several things regarding the images used to train and validate the models that are not optimal. The large degree of downsampling that is performed on the images before they are used as input to the models to reduce the training time might cause loss of important features in the images. This is most crucial for the T2 weighted images since the resolution here is reduced by a factor of eight. Figure 5.3 shows a T2 weighted image with the original resolution and the same image with the downsampled resolution. It is clear that even for a trained human eye it is much more difficult to locate the tumor in the latter. Since the downsampling causes trouble for the human eye, it is reasonable to think that this will also cause some difficulties for the U-net models. With the DWI this will not be as substantial considering that they only get downsampled by a factor of two. However, a full resolution would most likely have been preferred for these images as well.

When the DWI were matched to and used together with the T2 weighted images, it was discovered that for approximately half of the patients the FOV of the DWI did not cover the whole tumor volume delineated on the T2 weighted images. As a result, in the dataset with both image types it occurred image slices where voxels within the tumor mask only had values for the T2 weighted image and in the DWI they were set to zero. This phenomenon was present in several image slices for patient 52, and it is visualized in figure 5.4. Due to this mismatch of FOV, the models trained with both image types might learn that voxels that are black in the DWI and have some specific values in the T2 weighted image correspond to tumor. This can again lead to incorrect predictions, and it might be part of the explanation as to why using both image types did not give improved performance. A simple solution to this problem would be to remove all slices where this occurs, in other words, use the FOV of the DWI when both image types are used as input. The downside of this is that some information will be lost, and this will be information about the tumor edges which might already be challenging to

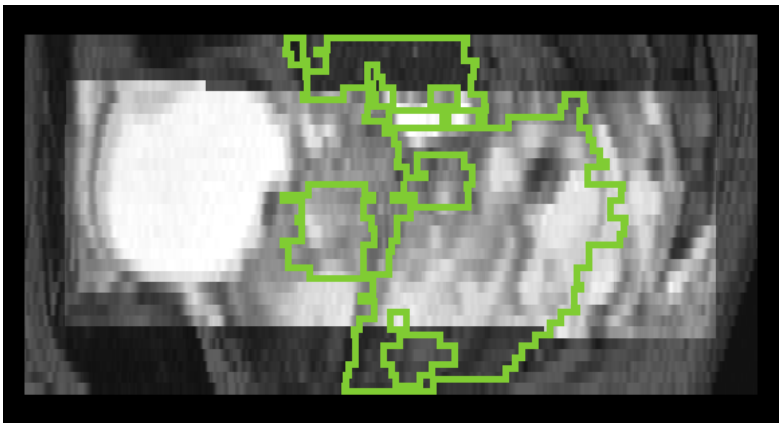


Figure 5.4: Visualization of the difference in FOV between the T2 weighted image and DWI for OxyTarget patient 52. The DWI is overlaid on the T2 weighted image making the region with both image types brighter. The tumor delineation from one of the radiologists is shown in green. The tumor extends beyond the FOV of the DWI on both sides.

predict due to a small tumor area in these slices.

The national guidelines for MR imaging of rectal cancer state that the image slices should be placed perpendicular to the tumor axis [4]. This causes a large variation in image orientation with regard to surrounding anatomy for the patients in the dataset. A consequence of this is that the images look different and it can be hard for the models to find consistent patterns with a limited amount of data. If the models try to make a prediction on images with an uncommon anatomical orientation, they might not recognize the tumor and/or predict a tumor in the wrong area. By adding data augmentation on the input images, the models would be exposed to different orientations and the size of the dataset would increase. As a result, this might get the models to learn the different anatomical image orientations and hence provide increased performance.

5.3 The support vector classifier model

By only looking at figure 4.9 and 4.10 it seems like the U-net model is superior to a shallow machine learning approach like the SVC. However, these two models might not be comparable. The images that are used as input to the model are different both in terms of which images are used and how they are pre-processed. For the SVC model, both T2 weighted images and DWI were used, while the U-net model only used the T2 weighted images. Thus, the models have access to different information and can not be directly compared.

Another important factor is that the images used for the SVC model were cropped significantly. The predictions were therefore done on a restricted part of the image where the tumor was placed in the center. This also removed the potential problem of having black edges around the images. For the images used as input to the U-net model, minimal pre-processing were performed, but they were

downsampled with a factor of eight. This gives a voxel size in the plane of about $2.8 \times 2.8\,\mathrm{mm}^2$, which is almost three times the size of the isotropic voxels used for the SVC model. As discussed above, less downsampling might give an increased performance for the U-net models, and one might also expect an increased performance by adding cropping of the images so that the tumor voxels make up a larger fraction of the total number of voxels, and thus increase the class balance.

No post-processing was done for the U-net model, and the only post-processing that was done for the SVC model was to remove the voxels in the tumor region that were not included in the FOV for both image types. These voxels were set to zero both in the prediction and ground truth before calculating the DSC. Since the U-net model used for the comparison only included T2 weighted images, the difference in FOV between the T2 weighted images and DWI was not a problem for this model. The predictions from the SVC model contained both large areas and smaller islands outside the true tumor volume, and could, therefore, benefit a lot from post-processing where one would remove these. With the U-net model, this was not a large problem, and it, therefore, seems like this type of post-processing will not be needed for this model.

It is also important to consider the training and validation process of the two models. The U-net model was trained on a cohort consisting of 51 patients and 10 different patients were used for the validation. The SVC model, on the other hand, used leave-one-out cross-validation and none of the patients were hold out as a final test set. It might have been better to train the SVC model on the same cohort that was used for training of the U-net model, and then tested the model on the validation set. In this way, it would have made more sense to compare the performance on the patients in the validation set. As for now, it is uncertain how the SVC model would perform on a separate dataset, and the U-net model is also not tested with a completely independent dataset.

5.4 Related work

The recent years there have been several approaches and attempts on automatic segmentation of tumor volumes and organs at risk for radiotherapy purposes based on medical images. Trebeschi et al. [14] developed a CNN for automatic segmentation of rectal cancer on multi-parametric MRI, similar to what was done in this thesis. They used a dataset consisting of T2 weighted images and DWI with four different b-values, ranging from $0\,\mathrm{s/mm}^2$ to $1100\,\mathrm{s/mm}^2$, from 140 patients with locally advanced rectal cancer. The classification was done by extracting a fixed patch around a voxel, the patch was then classified by the network and a probability was assigned to the corresponding voxel. By repeating this procedure for all the voxels in the image, a heatmap was generated. In this way, no encoder-decoder architecture was needed. To balance the training set an equal number of voxels was sampled from the tumor region and non-tumor region. For the non-tumor region the sampling was weighted to favor challenging regions like the tumor border and areas, apart from the tumor, that appeared bright in the DWI. This was done because it was assumed that the network would need more training to make

correct predictions in these regions. As a post-processing step, the largest component of the predicted segmentation was chosen as the tumor volume. Two expert radiologists had delineated the tumor volumes, and the first was used to train the network while the other one acted as an additional evaluation. It was reported a DSC of 0.68 and 0.70 for the two delineations respectively, and this is very similar to the results obtained on the validation set with the U-net models in this thesis.

Lee et al. [47] proposed a model for rectal cancer segmentation where they first segmented the rectum on the images and included data augmentation in order to reduce the model variance. The rectum segmentation was motivated by the geometric correlation between the rectum and rectal cancer. T2 weighted images from 457 patients were used, and one to two image slices was selected from each patient. This resulted in a dataset consisting of 907 image slices. The network consisted of an encoder-decoder architecture similar to the U-net models. By adding a rectum segmentation task they reported a reduction in the model variation by a factor of 0.90, and data augmentation further decreased the variance by a factor of 0.89. The resulting DSC for the rectal cancer delineations was 0.742 ± 0.0185 which is a slight improvement from the U-net models that were explored in this thesis. The DSC for the rectum segmentation was 0.943 ± 0.072, and this proves that the model in most cases was very accurate when predicting the location of the rectum.

A 3D U-net was recently suggested by Gurney-Champion et al. [48] for the delineation of metastatic lymph nodes in head and neck cancer. The dataset consisted of DWI taken from 48 patients and a total of 68 lymph nodes. Images were taken both before and during the treatment, and the patients were divided into two groups. One group of patients received definitive chemo-radiotherapy and the other received induction-chemotherapy. The images were cropped before they were used as input to the network, and this was done by selecting a random voxel within the lymph node contour to simulate a "mouse click" from a clinician, and then placing a bounding box with fixed size centered at the selected voxel. With 8-fold cross-validation, it was reported a DSC of 0.87 for the patients that received definitive chemo-radiotherapy and DSC of 0.80 for the patients that received induction-chemotherapy. The model was also tested on an independent dataset from an MR-Linac consisting of 3 patients and 8 lymph nodes, and this resulted in a DSC of 0.80. Considering that this model was used to delineate lymph nodes in head and neck cancer, it is not directly comparable to the models in this thesis, but it still proves that U-nets has a great potential in medical image segmentation. The high DSC for this model can be due to the cropping of the images. By cropping the images, one can obtain a data set that is relatively balanced between tumor voxels and non-tumor voxels, one removes areas that might be confusing for the models, and it limits the number of voxels that the model can classify incorrectly because there are fewer voxels in total.

5.5 Clinical impact

A method for automatic segmentation of tumor volume could potentially be time-saving. An ideal model would be able to take any image as input and output the segmentation of the tumor volume with an accuracy that is at least as good as a manual delineation made by an expert. A similar model could be used for the segmentation of organs at risk, and in that way limit the workload and time it takes to create radiotherapy treatment plans. As a result, this enables the possibility of changing the treatment plans between fractions, also known as adaptive radiotherapy, in order to optimize the treatment response for the individual patient.

The model required manual delineations for the training and validation process, and there is important to be aware that this makes the model biased in terms of the reader that made the delineations. The interobserver variation can be relatively large, and the two manual delineations used in this thesis resulted in a DSC equal to 0.78. The model will only be as good as the ground truth that is used, and one can ask the question of how accurate one would wish such a model to be. If a model gives an extremely accurate performance on a test set that has delineations from the same reader that made the delineations used for the training set, it is reasonable to think that a test set with delineations from a different reader would not perform as good. One should, therefore, consider if a model that performs reasonably well for a large range of delineations is a better option than a model that performs great but is biased towards delineations from a specific reader. An alternative to using the manual delineations as the ground truth would be to evaluate the predicted masks against the pathology of the tumors. In this way, one could obtain a more exact ground truth, but it would only be possible for the patients who undergo surgery and it might not be intuitive how the shape obtained from pathology relates to the images in terms of rotation.

In order to truly get an insight into how well an automatic segmentation model performs, the predictions should be evaluated and rated by several experts. There might be inaccuracies in the model predictions that can be tolerated and are no worse than the interobserver variations, but there can also be mistakes that can possibly get large consequences. Thus, if a model for automatic segmentation were to be implemented in the clinic, it is important that the predicted delineations are checked by experts before they are used for treatment planning.

An automatic segmentation would not only be beneficial for radiotherapy treatment planning, but also for research areas like quantitative image biomarkers and radiomics. Radiomics utilize statistical methods and machine learning approaches to explore shape and texture features in medical images in order to predict response to treatment and prognosis. This will make it possible to develop more personalized treatments for cancer patients. The data used in the analysis is extracted from the segmented volumes, and hence the segmentation is a crucial step in the radiomics process [16, 49]. The interobserver variation in the delineations can, therefore, cause reproducibility issues for the results obtained. Radiomics require large datasets, and the fact that manual delineation is a time-consuming task can be challenging. With an automatic segmentation model, one could obtain a standardized and fast method for segmentation, and hence more robust and

reproducible results for the image biomarkers and radiomics.

5.6 Further work

Several modifications can be implemented that might improve the model performance. A relatively easy modification would be to train the model without downsampling the images, and then see how this will affect the performance. Gurney-Champion et al. [48] investigated the image resolution in relation to their U-net model, and they found that a lower resolution decreased the performance slightly. Based on this finding, it is reasonable to believe that less downsampling will result in an improved model, but it will also result in longer training time.

To solve the issue with the different anatomical image orientations, data augmentation could be applied. By flipping and transforming the images, the model gets exposed to several variations of the same image. Data augmentation was shown to reduce the model variance by Lee et al. [47], and it could potentially improve the model's capability to give accurate predictions on a large variety of image orientations. As a consequence, it is also probable that this will improve the model's ability to generalize to other datasets.

Another approach to deal with the image orientation could be to create a model based on a 3D convolutional neural network like V-net [40] or 3D U-net [50] which have proven to give good results. This would allow the model to take 3D images as input, and thus the image orientation of the individual image slices would most likely not be important.

The U-net models took the complete images as input, and it should be investigated how the performance would change by adding cropping to the pre-processing. One can explore different amounts of cropping, and it will also be interesting to implement the "mouse click" approach that was used by Gurney-Champion et al. [48]. Having the radiologist click inside the tumor could be easily implemented in the clinic. Cropping of the images would lead to a more balanced dataset, and possibly also reduce the time it takes to train the models due to fewer voxels in each image.

For models that take both T2 weighted images and DWI as input, the FOV on the T2 weighted images should be matched to the FOV of the DWI to avoid having image slices with information from only one image type. It would be interesting to see if this results in improved performance for models including both image types. One can also investigate if it is necessary to include all seven different b-values for the DWI, or if the same performance can be achieved by only using a couple.

Except for the loss function, the hyperparameters for the models were kept unchanged. In a more thorough study, the hyperparameters like learning rate, batch size, and the number of epochs should be tuned to achieve the optimal model. The choice of activation function and optimizer should also be examined further.

Finally, when a model with adequately good performance for the validation set is found, the model should be evaluated on a test set it has never seen before. The result from this will then give an indication of the degree of overfitting present in

the model, and it will give a more generalized evaluation of the model performance. To take it one step further, the model could also be tested with datasets obtained from other hospitals. This will be completely independent datasets, both when it comes to the image acquisition and the manual delineation, and hence the ultimate test to see how well the model generalizes.

Chapter 6

Conclusion

In this thesis, a deep convolutional neural network was explored for the task of automatic segmentation of rectal tumor volume based on MR images. The dataset consisted of T2 weighted images and DWI with seven different b-values, ranging from $0\,\mathrm{s/mm^2}$ to $1300\,\mathrm{s/mm^2}$, from 81 patients. The data was divided between training, validation, and test set, and the union of manual delineations made by two radiologists was used as the ground truth. A total of nine models with a U-net architecture were trained. Between the different models, the input and the loss function were varied.

The best performing model was trained with T2 weighted images and the modified Dice loss function, and it resulted in a DSC of 0.67 for the validation set. The DSC for each patient in the validation set ranged from 0.27 to 0.85. This U-net model proved to be superior to a shallow machine learning model based on the SVC, which gave an average DSC of 0.48.

Compared to the interobserver variation of the two manual delineations, the U-net model had a lower DSC, but several modifications can be done in an attempt to improve the performance. Having less image downsampling, and adding data augmentation and cropping are modifications that are believed to give increased performance. In conclusion, a convolutional neural network with U-net architecture gives promising results for rectal tumor segmentation, and it should be explored further.

Bibliography

[1] World Health Organization. *Cancer*. URL: https://www.who.int/health-topics/cancer#tab=tab_1. (accessed: 23.11.2019).

[2] Cancer Research UK. *Worldwide cancer statistics*. URL: https://www.cancerresearchuk.org/health-professional/cancer-statistics/worldwide-cancer#heading-Zero. (accessed: 23.11.2019).

[3] Cancer Registry of Norway. *Cancer in Norway 2018 - Cancer incidence, mortality, survival and prevalence in Norway*. 2019.

[4] Helsedirektoratet. *Nasjonalt handlingsprogram med retningslinjer for diagnostikk, behandling og oppfølging av kreft i tykktarm og endetarm*. IS-2849. 2019.

[5] Fiona G. M. Taylor et al. "A Systematic Approach to the Interpretation of Preoperative Staging MRI for Rectal Cancer". In: *American journal of roentgenology* 191.6 (2008), pp. 1827–1835. DOI: 10.2214/AJR.08.1004.

[6] Luís Curvo-Semedo et al. "Diffusion-weighted MRI in rectal cancer: Apparent diffusion coefficient as a potential noninvasive marker of tumor aggressiveness". In: *Journal of Magnetic Resonance Imaging* 35.6 (2012), pp. 1365–1371. DOI: 10.1002/jmri.23589.

[7] Elisabeth Weiss and Clemens F. Hess. "The Impact of Gross Tumor Volume (GTV) and Clinical Target Volume (CTV) Definition on the Total Accuracy in Radiotherapy: Theoretical Aspects and Practical Experiences". In: *Strahlentherapie und Onkologie* 179.1 (2003), pp. 21–30. DOI: 10.1007/s00066-003-0976-5. (Visited on 04/29/2020).

[8] C. F. Njeh. "Tumor delineation: The weakest link in the search for accuracy in radiotherapy". In: *Journal of medical physics* 33.4 (2008), pp. 136–140. DOI: 10.4103/0971-6203.44472.

[9] Shalini K. Vinod et al. "Uncertainties in volume delineation in radiation oncology: A systematic review and recommendations for future studies". In: *Radiotherapy and Oncology* 121.2 (2016), pp. 169–179. DOI: 10.1016/j.radonc.2016.09.009.

[10] Miriam M. van Heeswijk et al. "Automated and Semiautomated Segmentation of Rectal Tumor Volumes on Diffusion-Weighted MRI: Can It Replace Manual Volumetry?" In: *International Journal of Radiation Oncology*Biology*Physics* 94.4 (2016), pp. 824–831. DOI: 10.1016/j.ijrobp.2015.12.017.

[11] François Chollet. *Deep Learning with Python*. Manning Publications Co., 2017. ISBN: 9781617294433. URL: https://www.manning.com/books/deep-learning-with-python.

[12] Daniele Ravì et al. "Deep Learning for Health Informatics". In: *IEEE journal of biomedical and health informatics* 21.1 (2017), pp. 4–21. DOI: 10.1109/JBHI.2016.2636665.

[13] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Ed. by Nassir Navab et al. Springer International Publishing, 2015, pp. 234–241. ISBN: 9783319245744.

[14] Stefano Trebeschi et al. "Deep Learning for Fully-Automated Localization and Segmentation of Rectal Cancer on Multiparametric MR". In: *Scientific Reports* 7.1 (2017), p. 5301. DOI: 10.1038/s41598-017-05728-9.

[15] Isabel Dregely et al. "Imaging biomarkers in oncology: Basics and application to MRI". In: *Journal of Magnetic Resonance Imaging* 48.1 (2018), pp. 13–26. DOI: 10.1002/jmri.26058.

[16] Virendra Kumar et al. "Radiomics: the process and the challenges". In: *Magnetic Resonance Imaging* 30.9 (2012), pp. 1234–1248. DOI: 10.1016/j.mri.2012.06.010.

[17] Catherine Westbrook, Carolyn Kaut Roth, and John Talbot. *MRI in Practice*. Fourth edition. Blackwell Publishing Ltd, 2011. ISBN: 9781118273869.

[18] Vinit Baliyan et al. "Diffusion weighted imaging: Technique and applications". In: *World Journal of Radiology* 8.9 (2016), pp. 785–798. DOI: 10.4329/wjr.v8.i9.785.

[19] Roland Bammer. "Basic principles of diffusion-weighted imaging". In: *European Journal of Radiology* 45.3 (2003), pp. 169–184. DOI: 10.1016/S0720-048X(02)00303-0.

[20] Edward O Stejskal and John E Tanner. "Spin diffusion measurements: spin echoes in the presence of a time-dependent field gradient". In: *The journal of chemical physics* 42.1 (1965), pp. 288–292. DOI: 10.1063/1.1695690.

[21] Shijun Wang and Ronald M. Summers. "Machine learning and radiology". In: *Medical Image Analysis* 16.5 (2012), pp. 933–951. DOI: 10.1016/j.media.2012.02.005.

[22] Tom M. Mitchell. *Machine learning*. McGraw-Hill Science/Engineering/Math, 1997. ISBN: 0070428077.

[23] Valentino Zocca et al. *Python Deep Learning*. Packt Publishing, 2017. ISBN: 9781786464453. URL: http://search.ebscohost.com/login.aspx?direct=true&db=e000xww&AN=1513367&site=ehost-live.

[24] Gareth James et al. *An Introduction to Statistical Learning: with Applications in R*. First edition. Springer, 2013. ISBN: 9781461471370.

[25] Sebastian Ruder. *An overview of gradient descent optimization algorithms*. 2016. arXiv: 1609.04747 [cs.LG].

[26] Ning Qian. "On the momentum term in gradient descent learning algorithms". In: *Neural Networks* 12.1 (1999), pp. 145–151. DOI: 10.1016/S0893-6080(98)00116-6.

[27] Diederik Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2014. arXiv: 1412.6980 [cs.LG].

[28] Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully Convolutional Networks for Semantic Segmentation". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 3431–3440. ISBN: 9781467369640. DOI: 10.1109/CVPR.2015.7298965.

[29] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. "Learning Deconvolution Network for Semantic Segmentation". In: *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*. ICCV '15. USA: IEEE Computer Society, 2015, pp. 1520–1528. ISBN: 9781467383912. DOI: 10.1109/ICCV.2015.178.

[30] Michal Drozdzal et al. *The Importance of Skip Connections in Biomedical Image Segmentation*. 2016. arXiv: 1608.04117 [cs.CV].

[31] Sebastian Raschka. *Python Machine Learning*. Packt Publishing, 2015. ISBN: 1783555130. URL: http://search.ebscohost.com/login.aspx?direct=true&db=e000xww&AN=1071004&site=ehost-live.

[32] Lee R. Dice. "Measures of the amount of ecologic association between species". In: *Ecology* 26.3 (1945), pp. 297–302.

[33] Kelly H. Zou et al. "Statistical Validation of Image Segmentation Quality Based on a Spatial Overlap Index". In: *Academic radiology* 11.2 (2004), pp. 178–189. DOI: 10.1016/S1076-6332(03)00671-8.

[34] Kine Bakke et al. "Comparison of Intravoxel incoherent motion imaging and multiecho dynamic contrast-based MRI in rectal cancer". In: *Journal of Magnetic Resonance Imaging* 50 (2019). DOI: 10.1002/jmri.26740.

[35] Acredit. *The OxyTarget study*. URL: https://www.acredit.no/the-oxytarget-study/. (accessed: 27.11.2019).

[36] Kasper Marstal. *SimpleElastix Documentation, Release 0.1*. URL: https://readthedocs.org/projects/simpleelastix/downloads/pdf/latest/. (accessed: 27.04.2020).

[37] American Society of Clinical Oncology (ASCO). *Colorectal Cancer: Stages*. URL: https://www.cancer.net/cancer-types/colorectal-cancer/stages. (accessed: 23.04.2020).

[38] Rhiannon van Loenhout et al. *Rectal Cancer - MR staging 2.0*. URL: https://radiologyassistant.nl/abdomen/rectal-cancer-mr-staging-2-0. (accessed: 23.04.2020).

[39] Andrew Collette. *Python and HDF5: Unlocking Scientific Data*. O'Reilly Media, 2013. ISBN: 9781491944981.

[40] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. "V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation". In: *2016 Fourth International Conference on 3D Vision (3DV)*. IEEE, 2016, pp. 565–571. ISBN: 9781509054077. DOI: 10.1109/3DV.2016.79.

[41] François Chollet et al. *Keras*. https://keras.io. 2015.

[42] Travis E Oliphant. *A guide to NumPy*. Trelgol Publishing USA, 2006.

[43] Bradley Lowekamp et al. "The Design of SimpleITK". In: *Frontiers in Neuroinformatics* 7 (2013), p. 45. DOI: 10.3389/fninf.2013.00045.

[44] Fabian Pedregosa et al. "Scikit-learn: Machine learning in Python". In: *Journal of machine learning research* 12 (2011), pp. 2825–2830.

[45] Dask Development Team. *Dask: Library for dynamic task scheduling*. 2016. URL: https://dask.org.

[46] Luis Perez and Jason Wang. *The Effectiveness of Data Augmentation in Image Classification using Deep Learning*. 2017. arXiv: 1712.04621 [cs.CV].

[47] Joohyung Lee et al. "Reducing the Model Variance of a Rectal Cancer Segmentation Network". In: *IEEE Access* 7 (2019), pp. 182725–182733. DOI: 10.1109/access.2019.2960371.

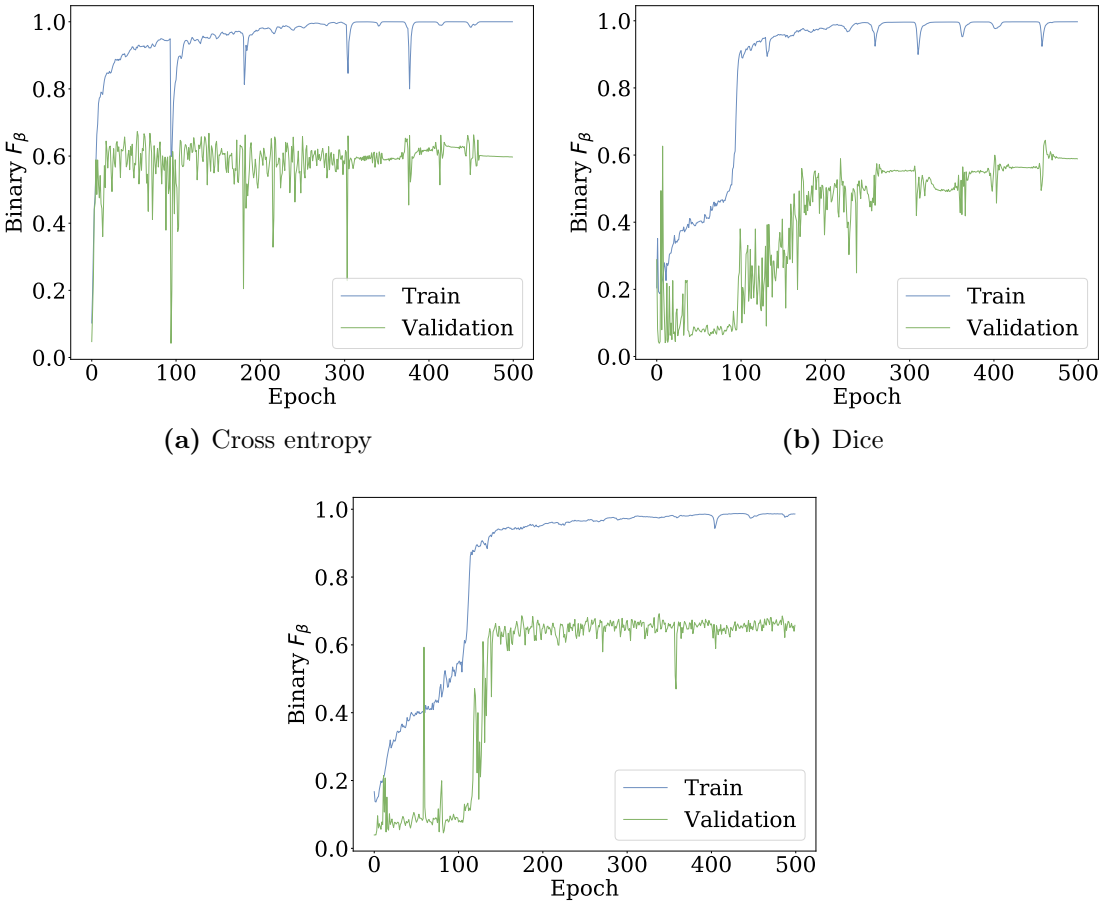
[48] Oliver J Gurney-Champion et al. "A convolutional neural network for contouring metastatic lymph nodes on diffusion-weighted magnetic resonance images for assessment of radiotherapy response". In: *Physics and Imaging in Radiation Oncology* 15 (2020), pp. 1–7. DOI: 10.1016/j.phro.2020.06.002.

[49] Stefania Rizzo et al. "Radiomics: the facts and the challenges of image analysis". In: *European Radiology Experimental* 2 (2018). DOI: 10.1186/s41747-018-0068-z.

[50] Özgün Çiçek et al. "3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation". In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016*. Ed. by Sebastien Ourselin et al. Springer International Publishing, 2016, pp. 424–432. ISBN: 9783319467238.

Appendix A

Training and validation curves

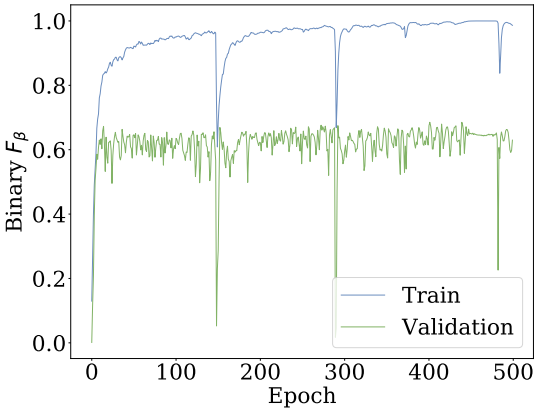


(a) Cross entropy

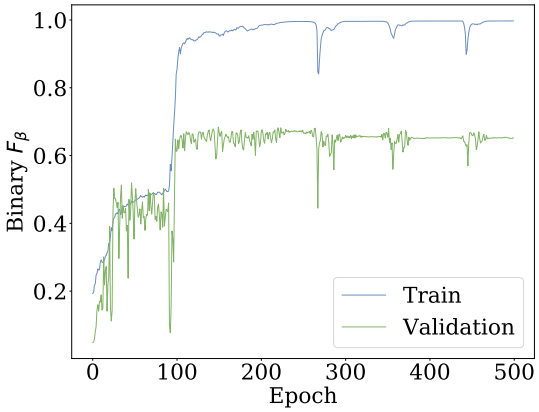
(b) Dice

(c) Modified dice

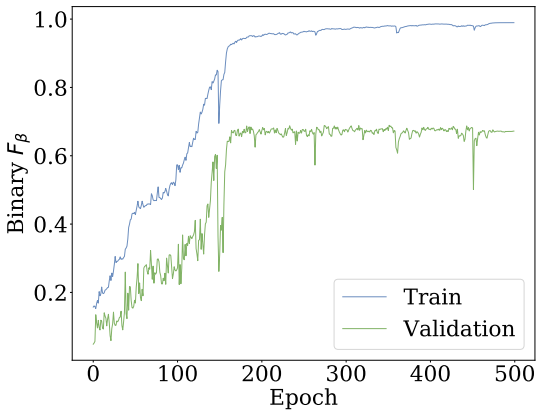
Figure A.1: Training and validation curves for the models with T2 weighted images as input. (a) corresponds to the model with cross entropy loss, (b) corresponds to the model with dice loss, and (c) corresponds to the model with modified dice loss. The binary F_β is the DSC calculated on all image slices in the dataset combined, with a threshold equal to 0.5. The green curves represent the validation set while the blue curves represent the training set.



(a) Cross entropy



(b) Dice

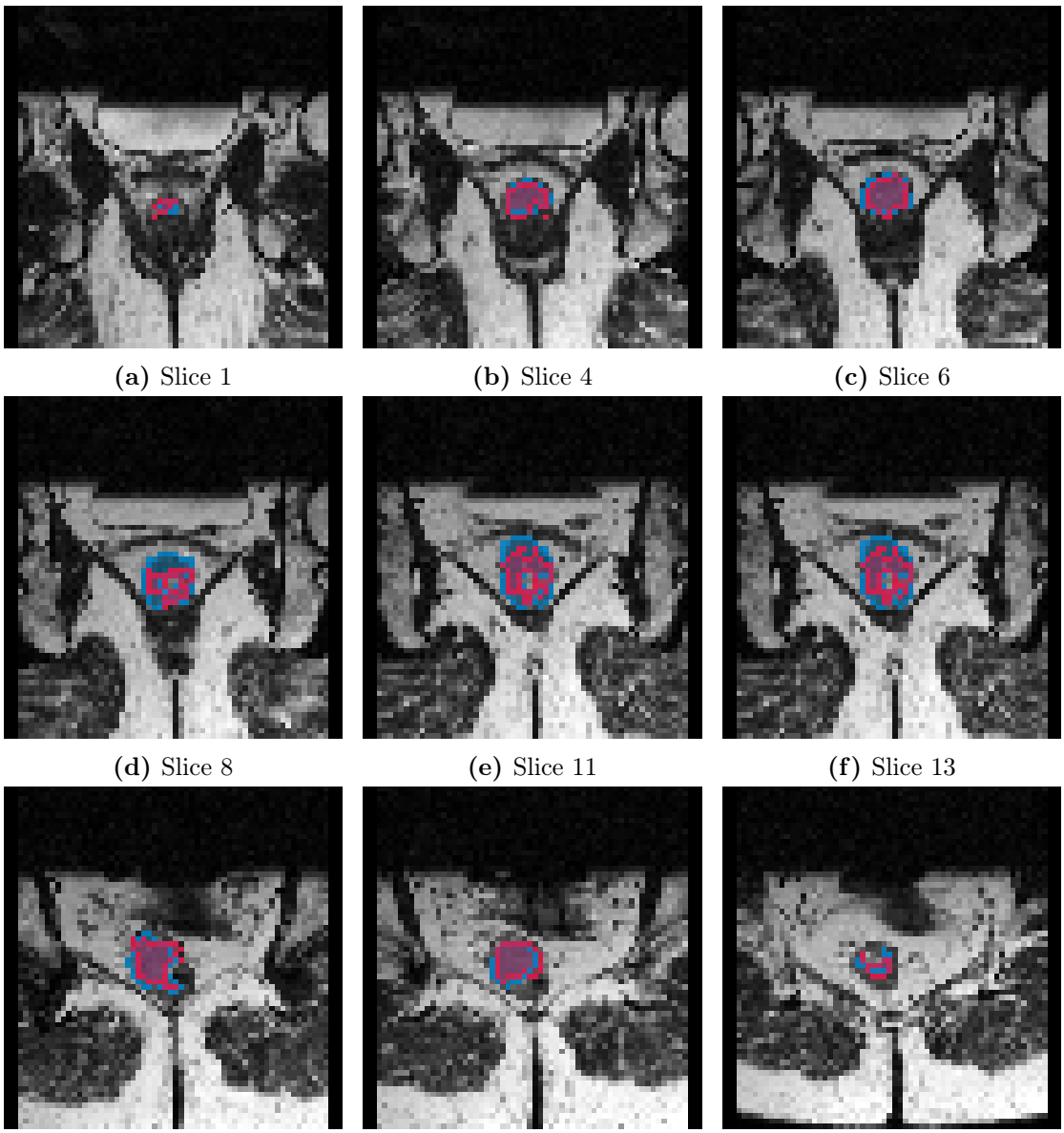


(c) Modified dice

Figure A.2: Training and validation curves for the models with DWI as input. (a) corresponds to the model with cross entropy loss, (b) corresponds to the model with dice loss, and (c) corresponds to the model with modified dice loss. The binary F_β is the DSC calculated on all image slices in the dataset combined, with a threshold equal to 0.5. The green curves represent the validation set while the blue curves represent the training set.

Appendix B

Delineations on the validation set

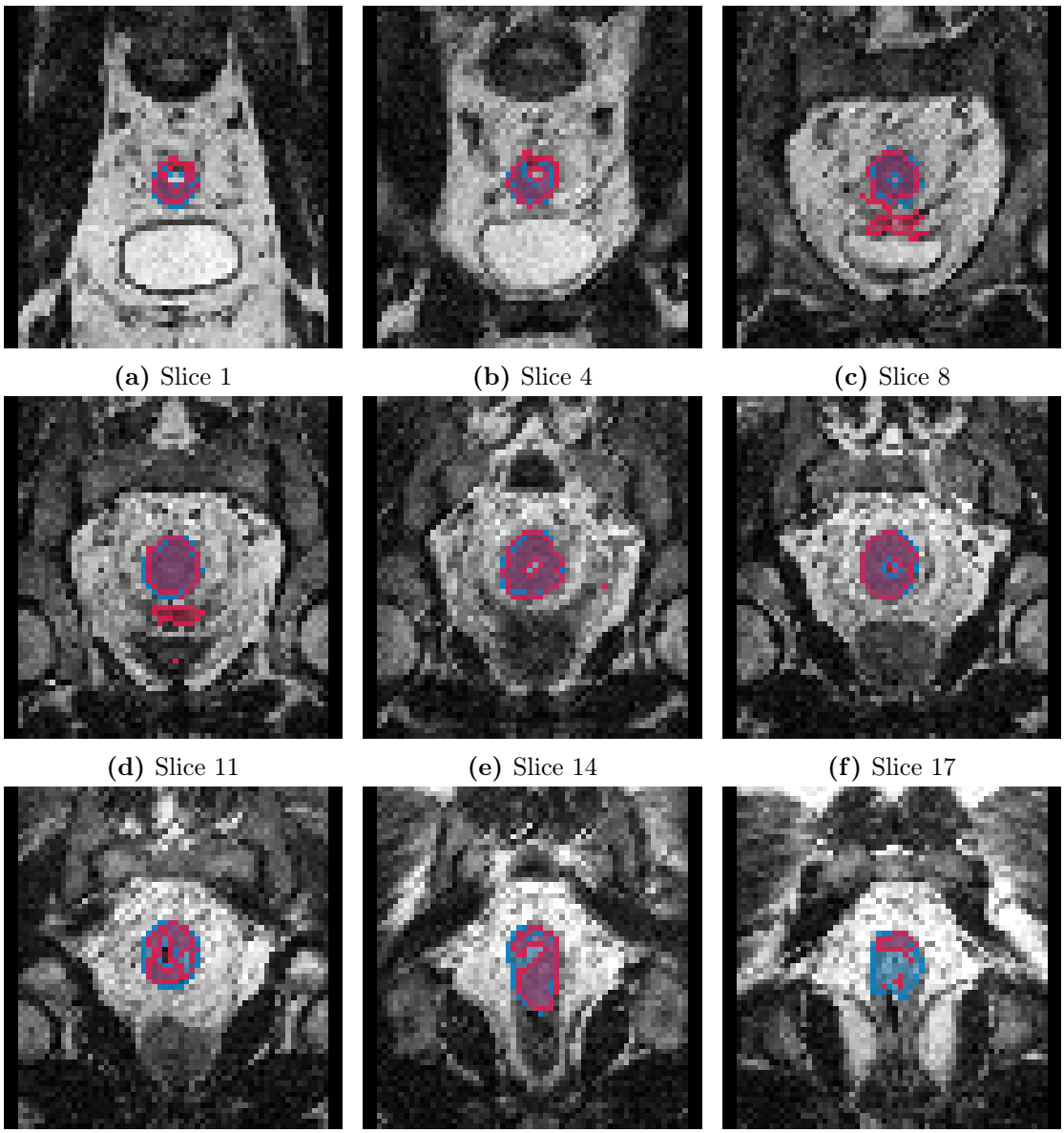


(a) Slice 1 **(b)** Slice 4 **(c)** Slice 6

(d) Slice 8 **(e)** Slice 11 **(f)** Slice 13

(g) Slice 16 **(h)** Slice 18 **(i)** Slice 21

Figure B.1: The delineation predicted by the best U-net model (red) and the union of the delineations made by the two radiologists (blue) on a selection of T2 weighted image slices from OxyTarget patient 72. The DSC for this patient is 0.74.

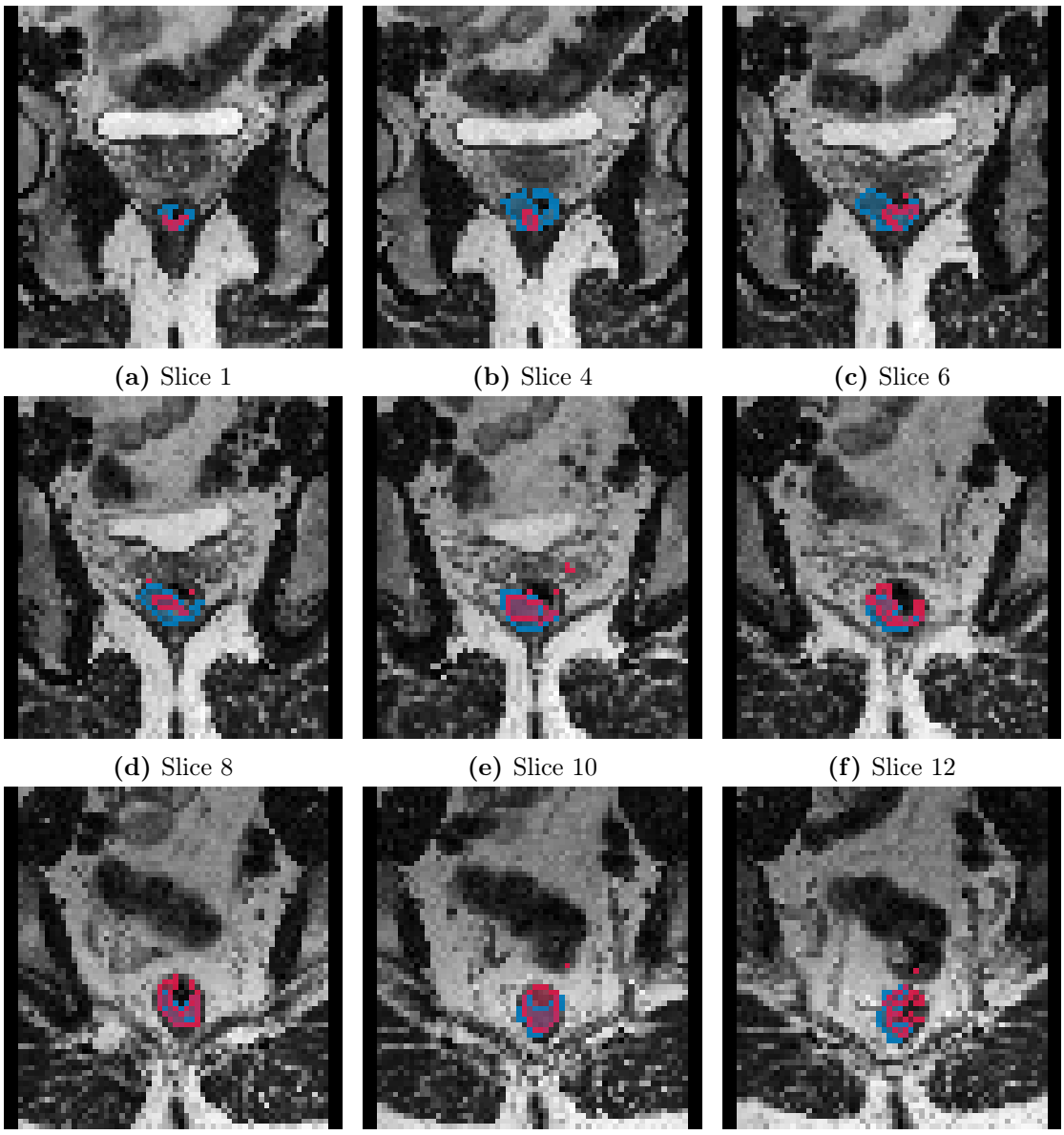


(a) Slice 1 **(b)** Slice 4 **(c)** Slice 8

(d) Slice 11 **(e)** Slice 14 **(f)** Slice 17

(g) Slice 20 **(h)** Slice 24 **(i)** Slice 27

Figure B.2: The delineation predicted by the best U-net model (red) and the union of the delineations made by the two radiologists (blue) on a selection of T2 weighted image slices from OxyTarget patient 74. The DSC for this patient is 0.82.

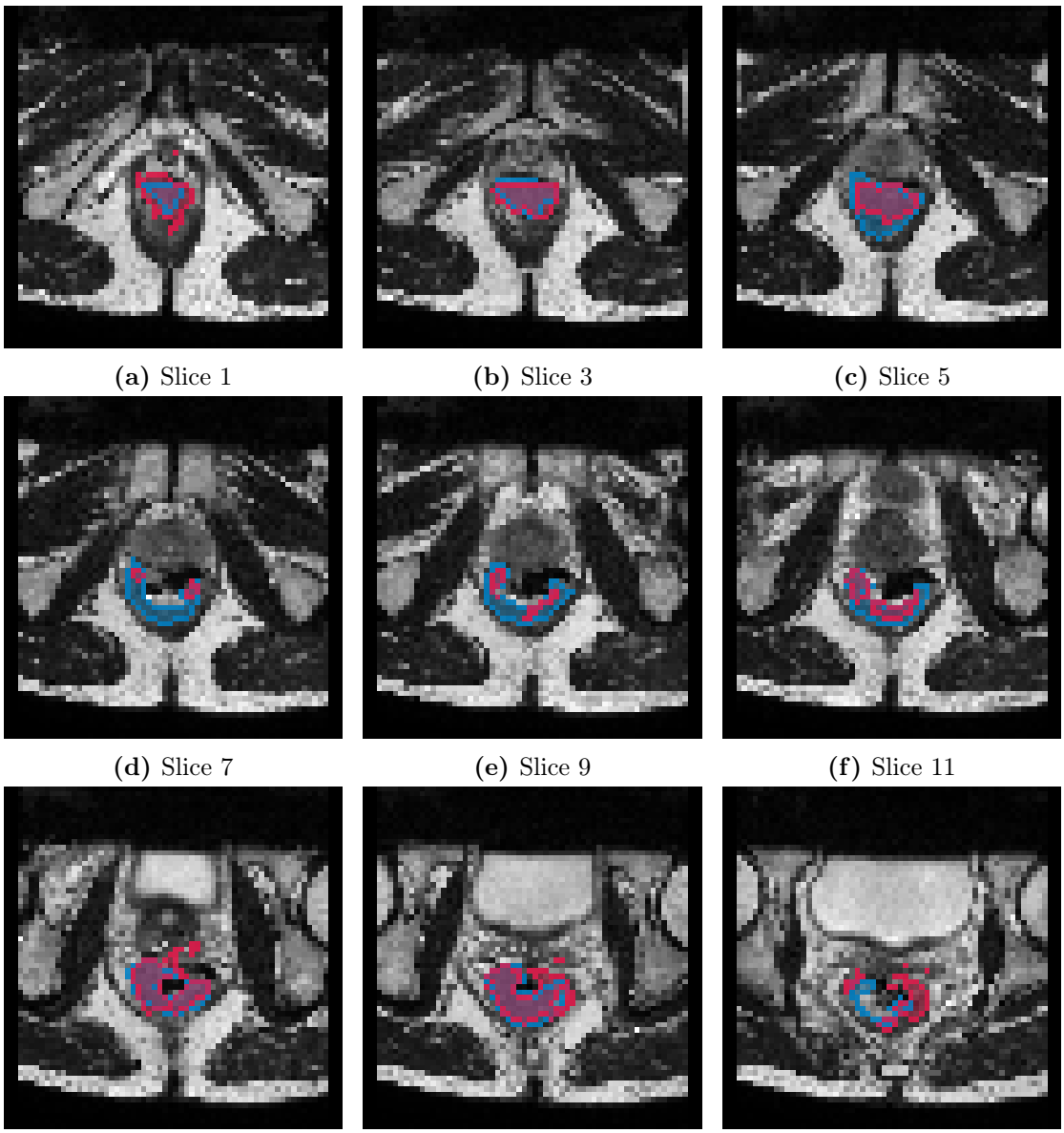


(a) Slice 1 **(b)** Slice 4 **(c)** Slice 6

(d) Slice 8 **(e)** Slice 10 **(f)** Slice 12

(g) Slice 14 **(h)** Slice 16 **(i)** Slice 18

Figure B.3: The delineation predicted by the best U-net model (red) and the union of the delineations made by the two radiologists (blue) on a selection of T2 weighted image slices from OxyTarget patient 88. The DSC for this patient is 0.60.

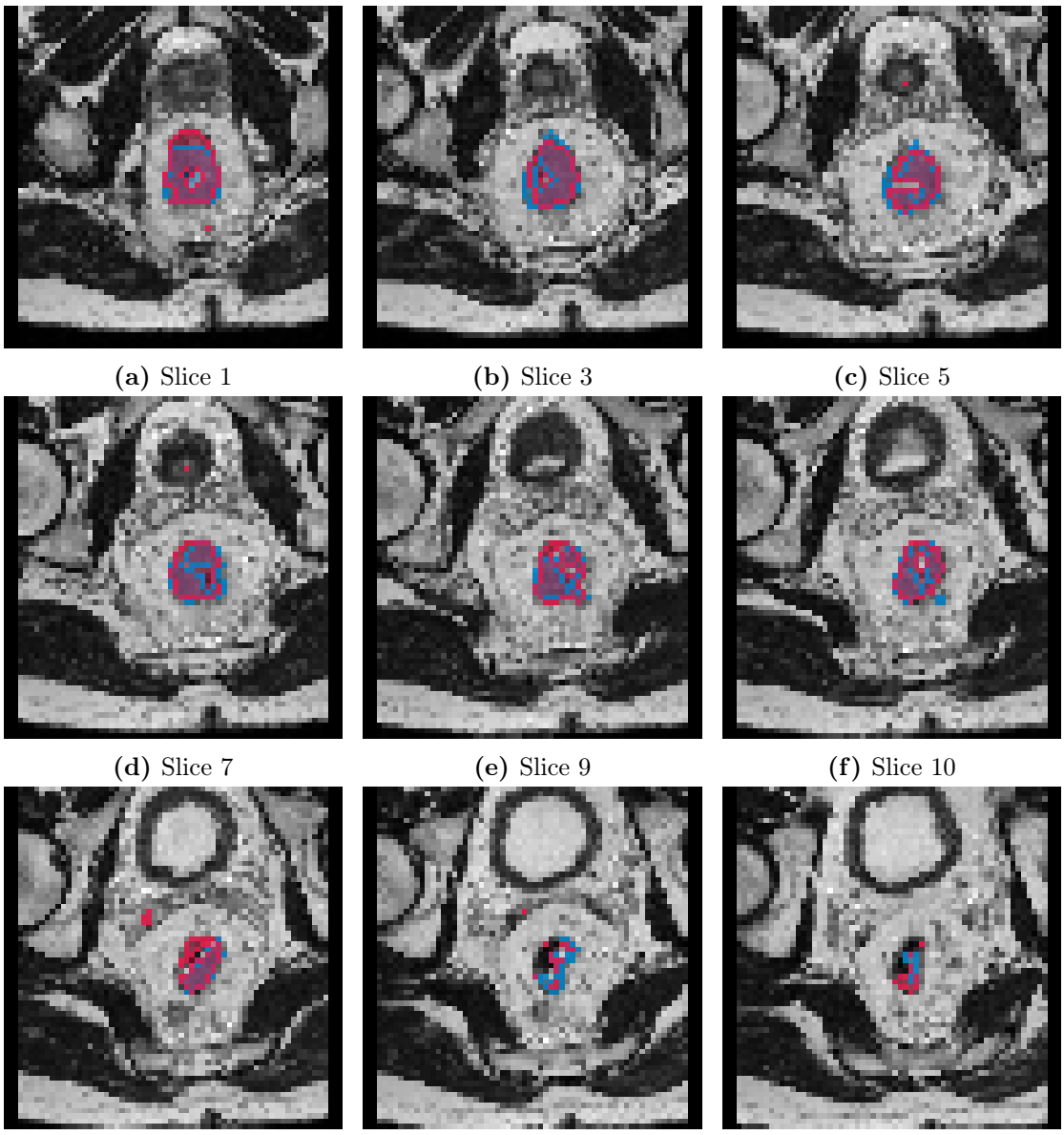


(a) Slice 1 **(b)** Slice 3 **(c)** Slice 5

(d) Slice 7 **(e)** Slice 9 **(f)** Slice 11

(g) Slice 13 **(h)** Slice 15 **(i)** Slice 18

Figure B.4: The delineation predicted by the best U-net model (red) and the union of the delineations made by the two radiologists (blue) on a selection of T2 weighted image slices from OxyTarget patient 125. The DSC for this patient is 0.68.

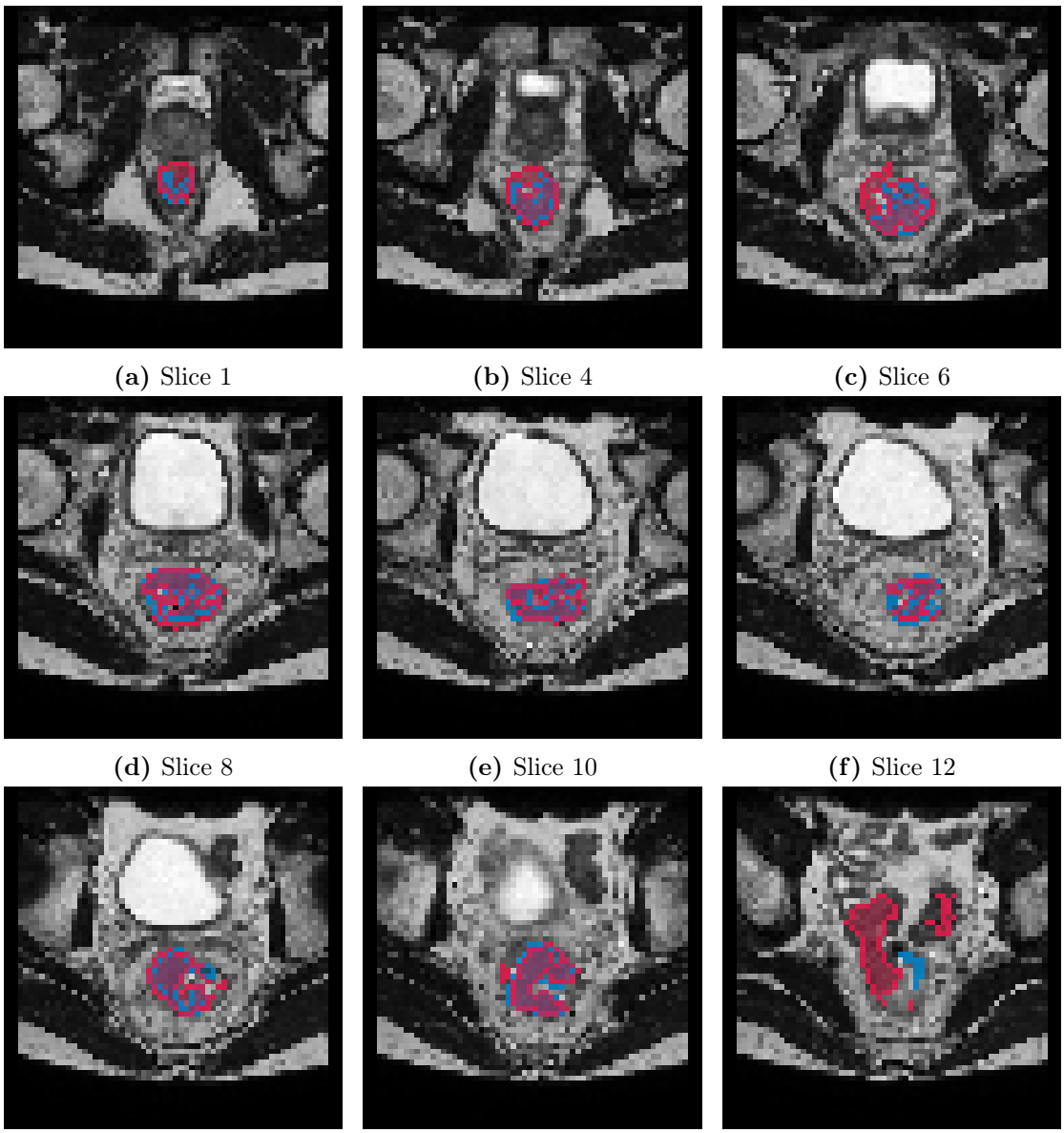


(a) Slice 1 **(b)** Slice 3 **(c)** Slice 5

(d) Slice 7 **(e)** Slice 9 **(f)** Slice 10

(g) Slice 12 **(h)** Slice 14 **(i)** Slice 16

Figure B.5: The delineation predicted by the best U-net model (red) and the union of the delineations made by the two radiologists (blue) on a selection of T2 weighted image slices from OxyTarget patient 128. The DSC for this patient is 0.80.

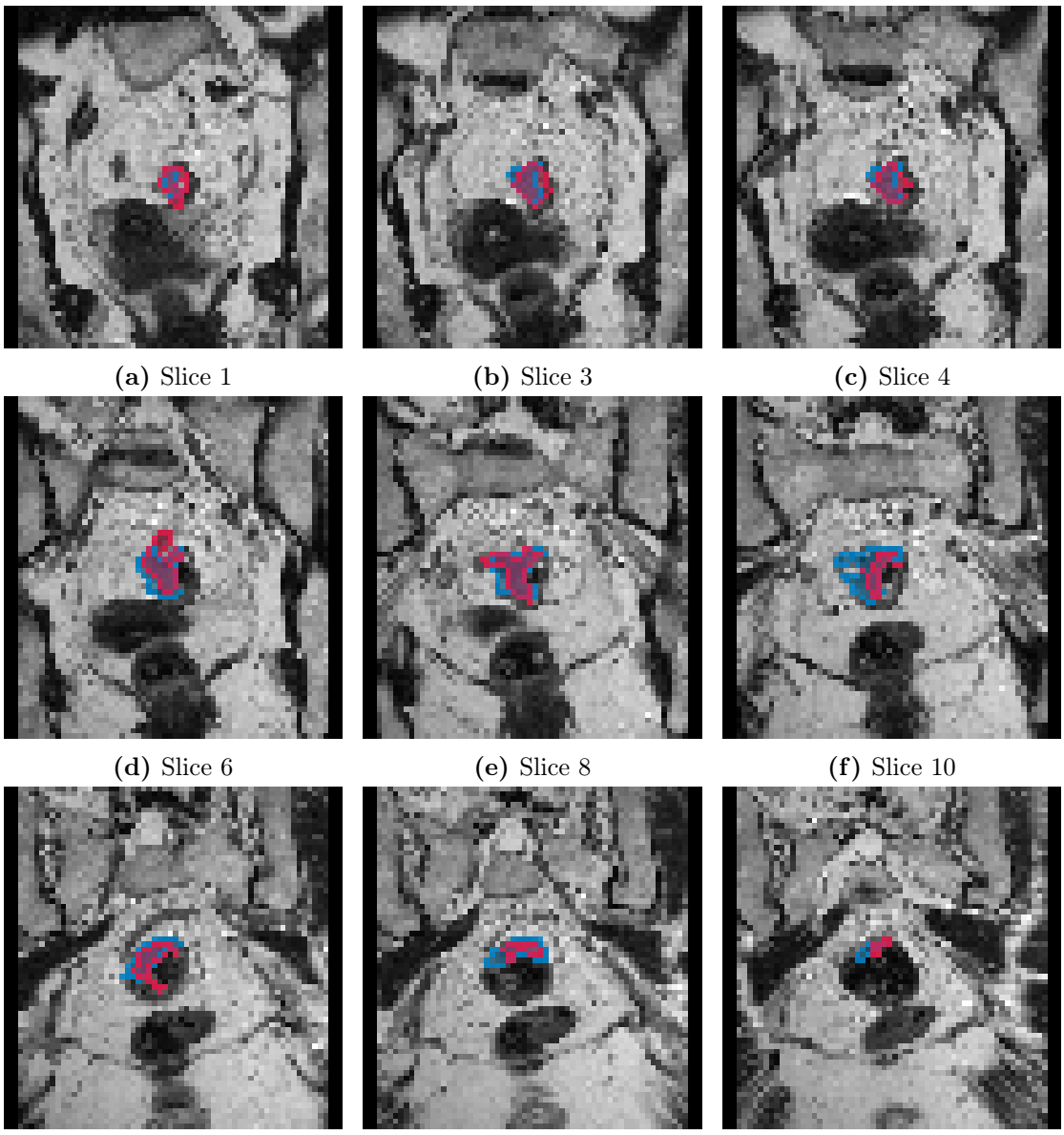


(a) Slice 1 **(b)** Slice 4 **(c)** Slice 6

(d) Slice 8 **(e)** Slice 10 **(f)** Slice 12

(g) Slice 14 **(h)** Slice 16 **(i)** Slice 19

Figure B.6: The delineation predicted by the best U-net model (red) and the union of the delineations made by the two radiologists (blue) on a selection of T2 weighted image slices from OxyTarget patient 148. The DSC for this patient is 0.74.

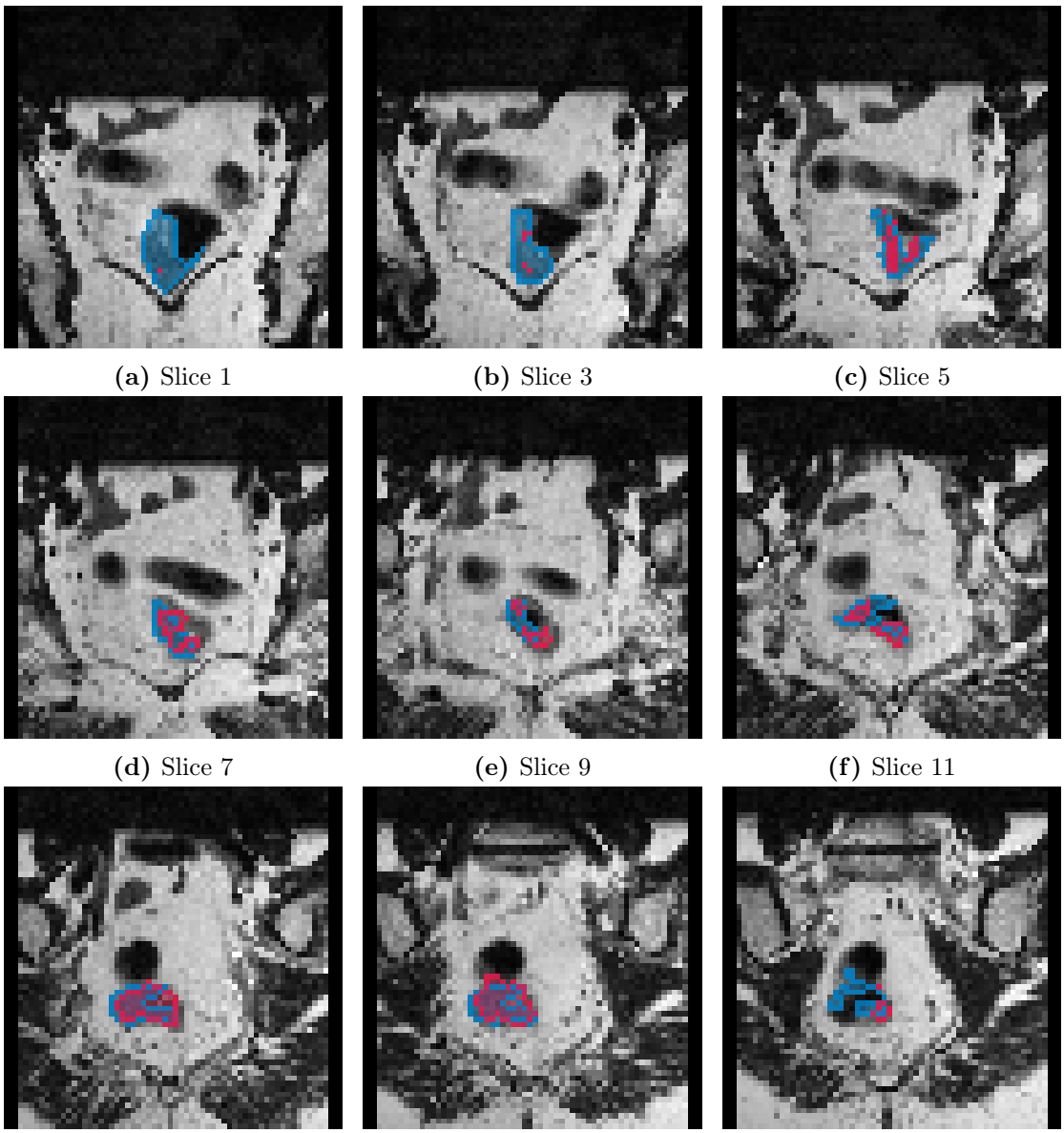


(a) Slice 1 **(b)** Slice 3 **(c)** Slice 4

(d) Slice 6 **(e)** Slice 8 **(f)** Slice 10

(g) Slice 12 **(h)** Slice 13 **(i)** Slice 15

Figure B.7: The delineation predicted by the best U-net model (red) and the union of the delineations made by the two radiologists (blue) on a selection of T2 weighted image slices from OxyTarget patient 156. The DSC for this patient is 0.69.



(a) Slice 1 **(b)** Slice 3 **(c)** Slice 5

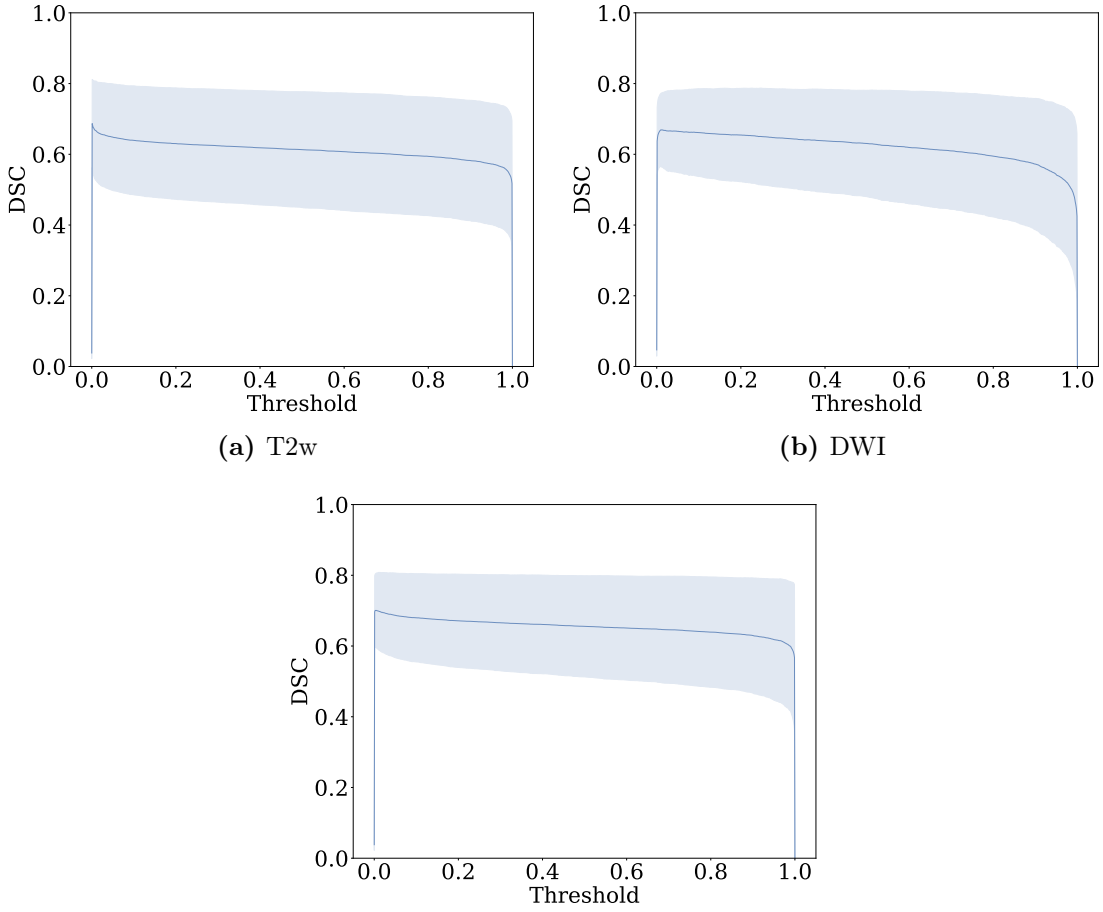
(d) Slice 7 **(e)** Slice 9 **(f)** Slice 11

(g) Slice 13 **(h)** Slice 15 **(i)** Slice 17

Figure B.8: The delineation predicted by the best U-net model (red) and the union of the delineations made by the two radiologists (blue) on a selection of T2 weighted image slices from OxyTarget patient 157. The DSC for this patient is 0.53.

Appendix C

Threshold

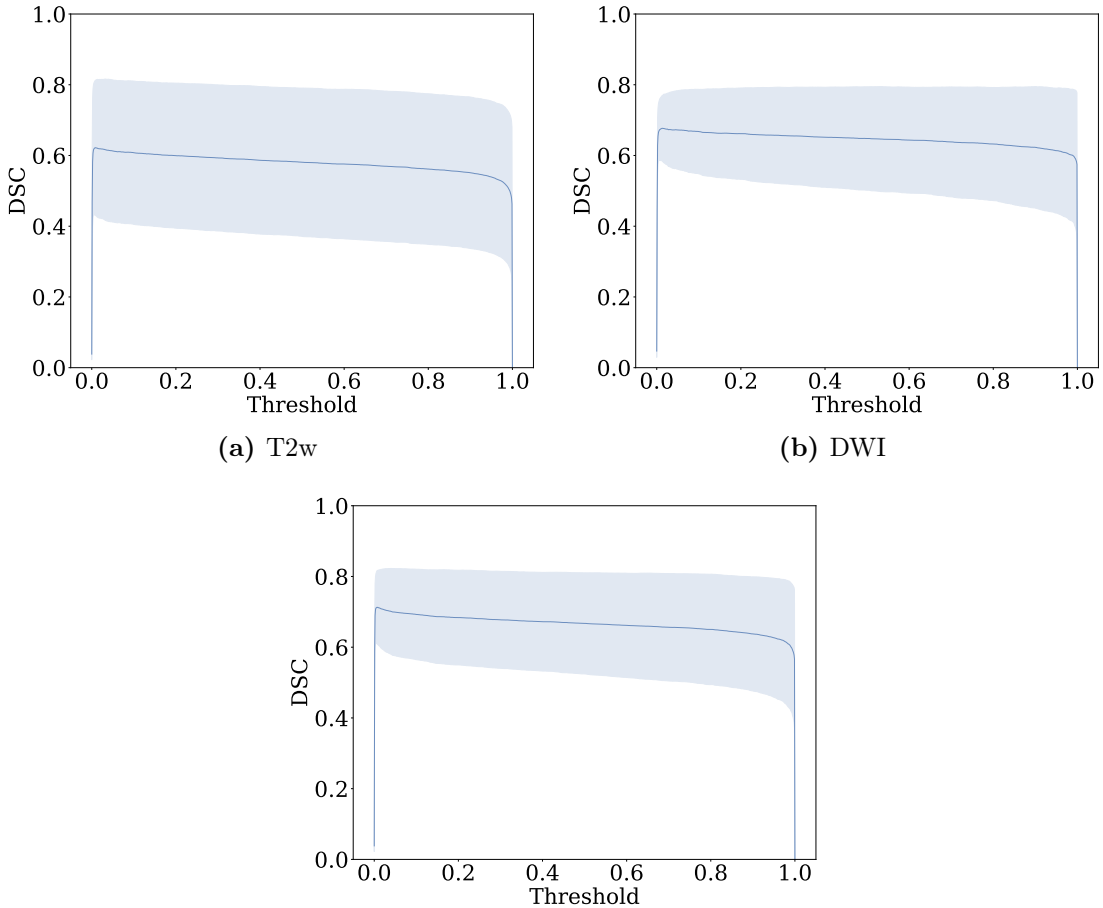


(a) T2w

(b) DWI

(c) T2w + DWI

Figure C.1: The average DSC plotted against the threshold for the U-net models with the cross entropy loss function. (a) corresponds to the model with T2 weighted images as input, (b) corresponds to the model with DWI as input, and (c) corresponds to the model with the two image types combined. The light blue areas represent the standard deviation.

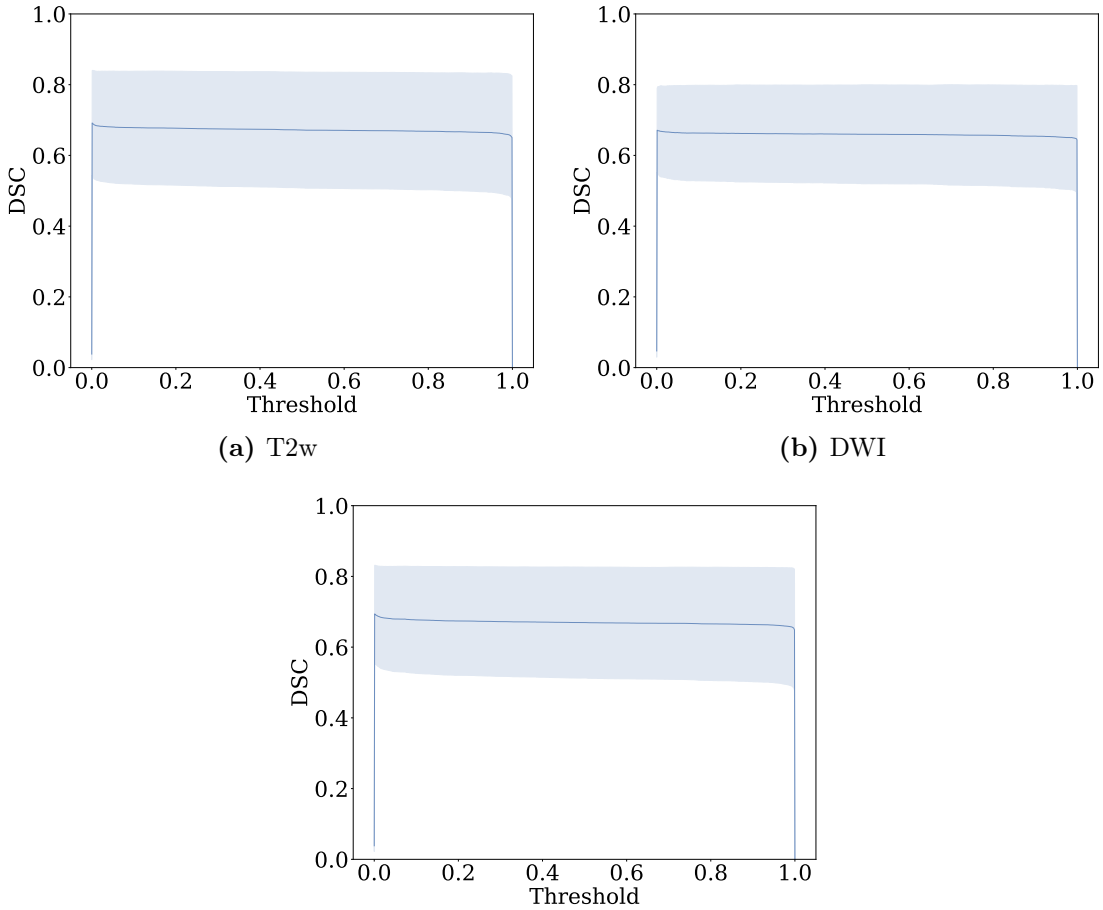


(a) T2w

(b) DWI

(c) T2w + DWI

Figure C.2: The average DSC plotted against the threshold for the U-net models with the Dice loss function. (a) corresponds to the model with T2 weighted images as input, (b) corresponds to the model with DWI as input, and (c) corresponds to the model with the two image types combined. The light blue areas represent the standard deviation.



(a) T2w

(b) DWI

(c) T2w + DWI

Figure C.3: The average DSC plotted against the threshold for the U-net models with the modified Dice loss function. (a) corresponds to the model with T2 weighted images as input, (b) corresponds to the model with DWI as input, and (c) corresponds to the model with the two image types combined. The light blue areas represent the standard deviation.

Eline Furu Skjelbred

Tumor segmentation by deep learning

NTNU
Norwegian University of
Science and Technology