Automatic classification of prostate cancer Gleason scores from biparametric MRI using deep convolutional neural networks

Master of Science Thesis
University of Turku
Department of Computing
Computer Science
2023
Kadir Demir

UNIVERSITY OF TURKU
Department of Computing

Kadir Demir: Automatic classification of prostate cancer Gleason scores from bi-parametric MRI using deep convolutional neural networks

Master of Science Thesis, 54 p.
Computer Science
June 2023

Prostate cancer is one of the most common types of cancer in the world. To reduce the number of deaths caused by it, effective diagnostic methods are of paramount importance to detect the clinically significant cases early enough. The current diagnostic protocols include, among other methods, magnetic resonance imaging which can be used to assess whether a patient suffers from prostate cancer and whether the possible cancer lesions are clinically significant. However, the images are difficult to interpret, and thus the inter-reader reliability is not very good. To address this problem, in this thesis machine learning models are trained to automatically segment and classify prostate cancer lesions from magnetic resonance images.
The problem proved to be difficult even for computers, at least with the relatively small data set size. The highest Dice similarity coefficients for the used Gleason score groups approached 0.4, which is not enough to replace the work of professionals or even provide meaningful help for doctors. In conclusion, the task of automatic segmentation and classification of prostate cancer lesions remains an open problem. Improving the performance to a useful level would likely require a noticeably larger dataset or at least a model that better incorporates the knowledge of the trained professionals.

Keywords: prostate cancer, image segmentation, machine learning, magnetic resonance imaging

Contents

1 Introduction

Prostate cancer is the second most frequently discovered type of cancer among men in the world, being only a tiny amount less commonly found than lung cancer. In 2020, it was estimated that there would be over 1.4 million new cases of prostate cancer globally, and that the disease would cause 375,000 deaths. (Sung et al., 2021.) Prostate cancer is especially commonly discovered in the more developed nations, in which it is the most common cancer in men. However, when standardized for age, prostate cancer is also very common in South America, Southern Africa, and the Caribbean. (Center et al., 2012; Sung et al., 2021.) In 2011 the annual cost of prostate cancer care was estimated at nearly 12 billion dollars in the United States alone. Back then, this figure was projected to rise to around 16 billion dollars by 2020. (Mariotto et al., 2011.)

In order to reduce the number of deaths, diagnosing prostate cancer early enough is important. Since it is not currently feasible to perform diagnostic tests for all men frequently, it is critical to identify the people who are most likely to have prostate cancer. Age is the most notable risk factor, and very significant portions of older people have been found to be affected by prostate cancer in different studies. Other risk factors include for example genetic background, taller height, vertex pattern baldness, and certain diets. Over the last few years, performing widespread screening for prostate cancer among men with significant risk factors has been gaining popularity. (Bergengren et al., 2023.)

The screening method that is currently recommended by the EU council consists of first measuring the amount of prostate-specific antigen (Bergengren et al., 2023). The biggest drawback of this measurement is that it is known to produce many false positive results, thus resulting in overdiagnosis (Schröder et al., 2009). For this reason, the European Association of Urology recommends prostate-specific antigen testing only for men who have at least 15 years of individualized life expectancy remaining and who are well-informed about the potential shortcomings of the antigen-based tests (Mottet et al., 2021).

After prostate cancer is suspected, whether due to results of an antigen test or other reasons, follow-up tests are needed. In the European Union, multiparametric magnetic resonance imaging is recommended as the next test (Bergengren et al., 2023). It allows trained doctors to further evaluate whether prostate cancer is present and at the same time estimate the clinical significance of cancer lesions if such are identified (Weinreb et al., 2016).

The standard method for confirming suspected prostate cancer and assessing its clinical significance is a procedure known as biopsy (Mottet et al., 2021). In biopsy, tissue samples are obtained from the prostate gland, and the samples are analyzed at the cell-level. Because of its invasiveness and potential of infection, it would be beneficial to perform biopsy only when the suspicion of prostate cancer is significant enough.

Jambor et al. (2017) showed that the more widely used multiparametric magnetic resonance imaging protocol could be efficiently replaced with a faster and cheaper biparametric imaging protocol. In addition, this method uses neither an endorectal coil nor intravenous contrast, which makes it much less invasive for the patient. When combined with a targeted biopsy afterwards, this procedure was shown to reduce the number of performed biopsies by 24% while missing only 2% of clinically significant prostate cancer cases.

In this thesis, machine learning models are developed in order to automatically identify, segment, and classify possible prostate cancer lesions from magnetic resonance images acquired with the biparametric imaging protocol. It is known that inter-reader reliability of interpreting magnetic resonance images with prostate cancer lesions is not very good even with the multiparametric imaging protocol which has more available information than the biparametric protocol (Mottet et al., 2021). As a result, it would be very beneficial to have a computational model that could interpret the images in an optimal manner at all locations or at least assist professionals in the analysis of the magnetic resonance images.

The next section includes a brief introduction to prostate cancer grading from magnetic resonance images, as well as necessary background to understand the used machine learning models. After that, some closely related work by others is outlined. The final sections include the description of the study, its results, and analysis of those results. The study itself is divided into three distinct phases. First, in order to validate that the chosen machine learning approach works, segmentation of the entire prostate gland is attempted. After this, lesions of certain clinical significance are segmented from the images without further classification. As the last step, the lesions are simultaneously segmented and assigned to one of the five groups defined by the International Society of Urological Pathology that indicate the severity of the disease.

2 Background

2.1 Prostate cancer grading with MRI

2.1.1 Anatomy of the prostate and prostate cancer

The human prostate is an anatomically heterogeneous organ that can be divided into four distinct anatomic regions: the central zone (CZ), the peripheral zone (PZ), the transition zone (TZ) and the anterior fibromuscular stroma. These regions have different tissue compositions, and thus they have distinct pathological properties. (McNeal, 1981.)

The CZ surrounds the ejaculatory ducts and contains 20–25% of the glandular tissue in the prostate (Greene et al., 1995; McNeal, 1981). It has been shown that it is rare for prostate cancer to originate from the CZ. McNeal et al. (1988) estimated the portion of prostate cancers originating from the CZ to be close to 10% based on 104 investigated prostate glands. However, Cohen et al. (2008) found the portion to be only around 2.5% based on a much larger sample of nearly 2500 tumors from over 1700 different patients. While the CZ tumors are not very common, they have been found to be more aggressive than tumors originating from the other regions of the prostate (Cohen et al., 2008).

The PZ is the largest region of the prostate, containing closer to 75% of the prostatic gland. It is located at the base of the prostate and surrounds the CZ partially. (Greene et al., 1995; McNeal, 1981.) The majority of prostate cancer

tumors, over 60%, originate from the PZ (Cohen et al., 2008; McNeal et al., 1988).

The TZ is located around the urethra close to the bladder. It is normally insignificant in size but can start growing, resulting in benign prostatic hyperplasia. (McNeal, 1981.) Around 30% of prostate cancer tumors originate from the TZ, which makes the TZ the second most important origin of prostate cancer tumors after the PZ (Cohen et al., 2008). TZ tumors have been found to be less aggressive than PZ tumors on average. They are less likely to spread outside the prostate and the tissue in them is usually not as deformed as in PZ tumors. (Greene et al., 1991; McNeal et al., 1988.)

There are no glands in the anterior fibromuscular stroma even though it contains roughly 30% of the mass of the prostate tissue (McNeal, 1981). Because of this, prostate cancer does not originate from the anterior fibromuscular stroma.

2.1.2 Prostate MRI

Magnetic resonance imaging (MRI) is a technique that can be used to get a view inside the human body in a non-invasive way. The patient is placed inside a magnetic field and the protons of the tissues are excited by oscillating the magnetic field at a radio frequency. When the protons relax back to the original state, they radiate the absorbed energy at a frequency which can be measured. The tissue in which the protons are, affects the received signal, which makes MRI great for separating different types of tissues. (Steyn and Smith, 1982.)

When MRI is used for prostate imaging, the images are often obtained using several different settings which emphasize specific types of tissue. The Prostate Imaging - Reporting and Data System Version 2 (PI-RADS™ v2) specifies standards for the MRI sequences that should be obtained for the prostate (Weinreb et al., 2016).

According to PI-RADS™ v2, both T_1-weighted (T_1w) and T_2-weighted (T_2w)

images are important. T_1w images are primarily used for finding the outline of the prostate gland and detecting hemorrhage while T_2w images reveal the zonal anatomy of the prostate and can be used to locate tumors and other abnormalities. Clinically significant tumors of the PZ appear as hypointense regions in T_2w images but this appearance can be caused by other conditions as well. Tumors of the TZ can show up in several different ways in the T_2w images, for example as a hypointense region or as a lenticular shape. The likelihood of tumor correlates with the number of these features that are present. (Weinreb et al., 2016.)

Diffusion-weighted imaging measures the random movement of water molecules in the tissues. For prostate MRI, the PI-RADS™ v2 recommends using diffusion-weighted images (DWI) with a b-value setting of over 1400 s/mm^2 as well as apparent diffusion coefficient (ADC) maps which are derived from several DWIs with different b-values by using a model of signal decay. (Weinreb et al., 2016.)

Dynamic contrast-enhanced (DCE) MRI is performed by doing T_1w gradient echo scans before, during and after injecting a contrast agent into the bloodstream. DCE MRI can reveal small but significant tumors that would be missed by the other methods, which is why it is recommended in the PI-RADS™ v2. However, the added value of DCE MRI is controversial and its role in classifying tumors is secondary to the other methods. (Weinreb et al., 2016.)

Using T_2w images, DWI, ADC maps, DCE MRI and sometimes other techniques together is known as multiparametric MRI (mpMRI) (Weinreb et al., 2016). It has recently been shown that using only T_2w images along with the diffusion-weighted imaging methods (Figure 2.1), which is known as biparametric MRI (bpMRI), can achieve very good results while avoiding some of the problems with the traditional mpMRI (Jambor et al., 2017; Stanzione et al., 2016). As DCE MRI is not performed in bpMRI, the process becomes significantly less invasive and quicker. Stanzione et al. (2016) were able to reduce the time spent for the MRI protocol from around 24

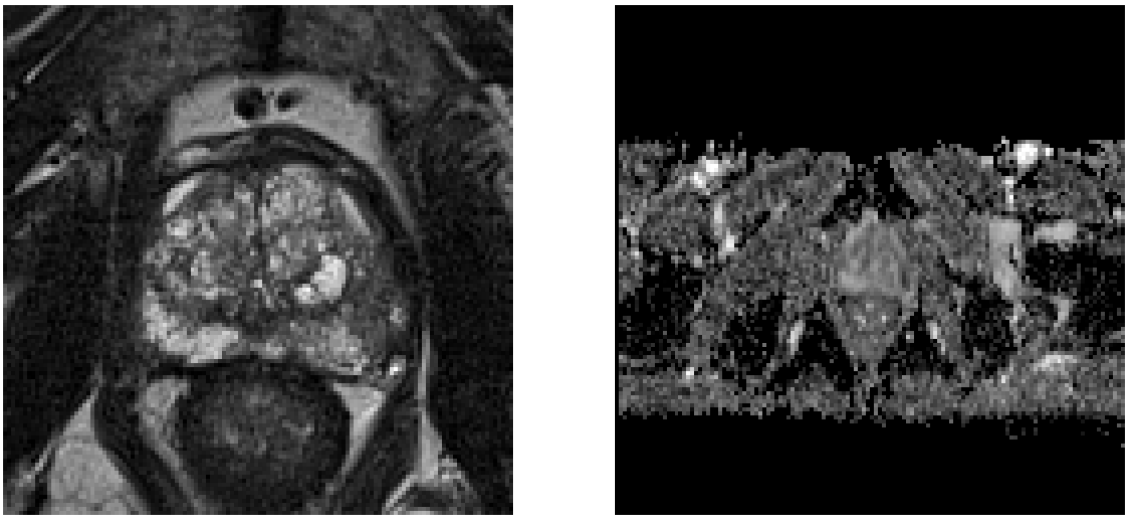

Figure 2.1: A T_2-weighted magnetic resonance image on the left and an apparent diffusion coefficient (ADC) map on the right. The prostate is located at the center of both images.

minutes to 17 minutes. Another benefit of dropping DCE MRI from the protocol is avoiding the use of gadolinium based contrast agents which are known to be relatively expensive and have been shown to likely accumulate in the dentate nuclei of the brain (Stanzione et al., 2016; Tedeschi et al., 2016).

2.1.3 Prostate cancer grading

The prognosis of prostate cancer can vary greatly from case to case. The aggressiveness of the cancer is used to select an appropriate treatment, and in many slowly progressing cases it may be unnecessary to use any treatment as the disease does not have a significant effect on the expected survival rate during the next few years (Wiederanders et al., 1963). Because of this, it is important to have an established system for grading different prostate cancer cases.

Perhaps the most well-known prostate cancer grading system was developed by Donald Gleason in 1966. The system assigns a score known as Gleason score to a

patient with cancer based on the morphology of the cells. The score is composed of two numbers, each of which can be between 1 and 5 and represents a cell pattern found in the prostate. The larger the pattern number, the more aggressive and dangerous the cancer is. The first of these numbers is the most prevalent pattern found in the prostate. The second number was originally the second most prevalent pattern found. (Gleason, 1966; Gleason and Mellinge, 1974.)

The Gleason grading system received a few updates at the 2005 International Society of Urological Pathology (ISUP) consensus conference. These changes included, for example, that for each separate tumor found in radical prostatectomy, a distinct grade should be reported. For needle biopsies, the second number of the Gleason score was agreed to represent the highest grade instead of the second most prevalent pattern. It was decided that Gleason scores 1 and 2 should not be assigned based on needle biopsies as these scores are hard to reproduce, have a low correlation with prostatectomy grades and may misguide clinicians and patients. (Epstein et al., 2005.)

In 2014, the grading system was further updated at a new ISUP conference, where the instructions for classifying certain patterns were updated. In addition to the updates, a new grouping system for the Gleason scores was agreed upon. This system reduced the number of grade groups to five from the large number of possible Gleason score combinations. It also works better than the nine possible groups that can be obtained by summing the two Gleason score numbers together because combined scores lower than 6 are rarely assigned and patients with the same combined score can have different prognoses. For example, Gleason scores 3 + 4 and 4 + 3 have a combined score of 7 even though 4 + 3 has a significantly worse prognosis. The new ISUP grouping system is shown in Table 2.1. (Epstein et al., 2016.)

While the Gleason scoring system is useful for grading tissue samples, it cannot

Table 2.1: The ISUP Gleason grade grouping system.

ISUP grade group	Combined Gleason group	Gleason scores
1	≤ 6	$\leq 3+3$
2	7	$3+4$
3	7	$4+3$
4	8	$4+4, 3+5, 5+3$
5	9 or 10	$4+5, 5+4, 5+5$

be used to grade images obtained from MRI. The PI-RADS™ v2 includes a grading system for quantifying the likelihood of clinically significant cancer being present in the images. Grade 1 is assigned when clinically significant cancer is unlikely and 5 when it is likely based on the mpMRI. If the grade is high enough, a biopsy is performed to get a more accurate diagnosis. (Weinreb et al., 2016.)

2.2 Image segmentation with deep convolutional neural networks

2.2.1 Machine learning

Machine learning encompasses computational techniques that learn how to perform a task by becoming better at it, as measured by a defined performance metric, when gaining more experience with data related to the task (Mitchell, 1997). Most machine learning algorithms can be divided into two groups: supervised and unsupervised learning algorithms. Unsupervised learning algorithms try to identify and learn useful properties of the available data itself. On the other hand, supervised learning algorithms experience target values in addition to the data and try to learn how to predict these targets when given new data. (Goodfellow et al., 2016.) The problems that are solved in this thesis fall into the domain of supervised learning.

Most supervised learning algorithms can be further classified as either parametric or non-parametric. When using parametric methods, the functional form is chosen first, and the training of the model is performed afterwards by estimating coefficients, or weights, that optimize a chosen loss function. For some learning algorithms, the optimal weights can be easily found with a closed-form expression, but in most cases an optimization algorithm must be used for the estimation. Non-parametric methods do not make assumptions about the functional form of the model and thus they have potential to fit the model to the data in more complex ways than parametric methods. However, non-parametric methods typically require far more data than parametric methods to obtain an accurate estimate. (James et al., 2013.)

The quality of the predictions made by a supervised learning algorithm can be measured by using performance metrics that compare the predictions with the actual target values. These metrics give an idea of how well the model is performing, and make comparing results to previously conducted studies easier. (Goodfellow et al., 2016.) In order to get a reliable estimation of the true performance of the model, the performance metrics must be computed using data that were not used to train the model. This can be done by dividing the data into training and test sets. The model is then fit to the data of the training set, after which the performance can be estimated on the test set. When there is not enough data for a separate test set, a method known as cross-validation can be used to estimate the performance. In cross-validation, the data is divided into folds of approximately equal size. The first fold is then used as the test set while the model is trained on all the other folds. This is then repeated until all the folds have been the test fold once. Finally, the performance metrics of each fold are combined. There are additional benefits to using cross-validation: it typically uses a larger portion of the available data to train the model, giving a better estimate of the true performance of the model trained on all the data, and it partially avoids the problem of some models having highly

variable performance depending on the exact training set that was used. (Hastie et al., 2009; James et al., 2013.)

Supervised learning algorithms typically have one or more parameters that determine the complexity of the model. The model complexity affects the training and test set errors, as measured by a performance metric. While the training set error continues to decrease as the complexity of the model is increased, the test set error starts to increase after the model passes a certain complexity threshold, which is shown in Figure 2.2. When the complexity is too low, the model cannot properly fit to the patterns present in the data and is said to underfit. On the other hand, when the complexity is too high, the model overfits by adapting too closely to the data of the training set, reducing the generalization capability. (Hastie et al., 2009.) The parameters that determine the complexity of the model, as well as other parameters that are not set by the fitting of the model, are known as hyperparameters. In order to choose the optimal values for the hyperparameters, additional data, a validation set, is needed. If the hyperparameters were set on the regular training set, the values that result in the highest model complexity would be chosen which would lead into overfitting. The validation set is split from the training data and if a simple split is not enough, cross-validation can be used to select the optimal parameter values. (Goodfellow et al., 2016.)

2.2.2 Feedfoward neural networks

Artificial neural networks are a group of machine learning techniques that are based on attempts of McCulloch and Pitts (1943) to create a mathematical model of the human nervous system. These techniques have grown to encompass a variety of different approaches: for example, unsupervised methods such as self-organizing maps (Kohonen, 1982) and supervised methods like feedforward and recurrent neural networks.

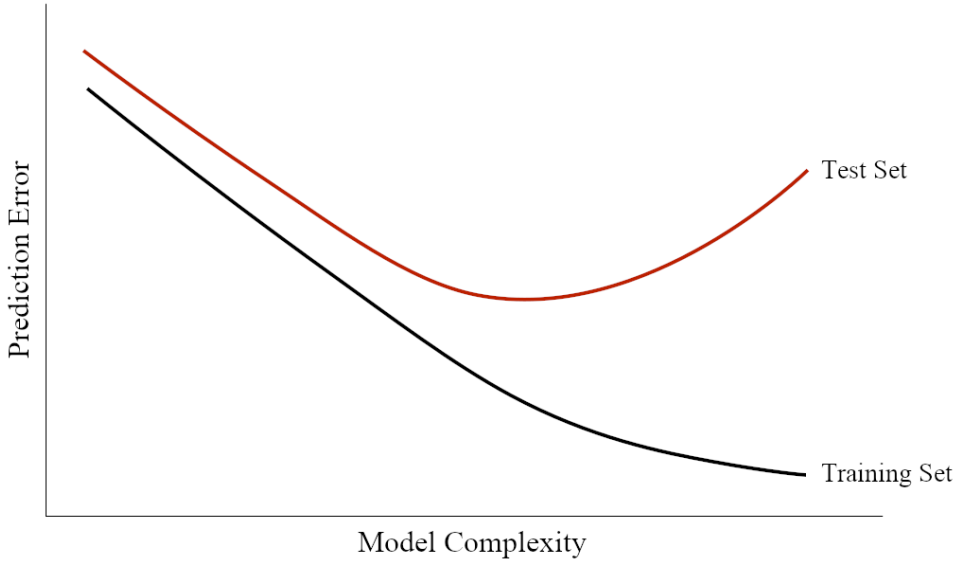

Figure 2.2: Effects of model complexity on training and test set errors as described by Hastie et al. (2009). As the model complexity increases, the training set error keeps decreasing, but the test set error starts to increase after a certain point.

Feedforward neural networks are used to approximate some unknown function by learning the parameters $\boldsymbol{\theta}$ that give the best approximation of the function $\boldsymbol{y} = f(\boldsymbol{x}; \boldsymbol{\theta})$. In feedforward networks, the information only flows forward, while in recurrent networks there are backwards connections as well. (Goodfellow et al., 2016.) The basic structure of a feedforward neural network can be thus illustrated with a directed acyclic graph in which the vertices represent the units of the network, and the edges the connections between them. An example of this is shown in Figure 2.3. Feedforward neural networks consist of multiple layers of functions which are chained together. The first layer is called the input layer, the last one is known as the output layer, and the layers between these are referred to as hidden layers. The overall number of the layers determines the depth of the network. The number of layers along with their dimensionality and the connections between them define the architecture of the model. (Goodfellow et al., 2016.) Feedforward neural networks with only one hidden layer have been proved to be universal approximators which means that they are able to approximate any Borel measurable function (Hornik et al., 1989). However, having multiple hidden layers is typically desirable as they tend to perform better than a single layer with a large dimensionality (Goodfellow et al., 2016).

The output of each layer of the network is defined by a function known as the activation function $g(\boldsymbol{z})$. These functions usually take in as parameters the affine transformation $\boldsymbol{z} = \boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{b}$ where $\boldsymbol{x}$ is a vector of the outputs of the connected nodes of the previous layers, $\boldsymbol{W}$ is a weight matrix, and $\boldsymbol{b}$ is a vector of values known as biases. Most activation functions apply an element-wise nonlinear transformation to the parameters. (Goodfellow et al., 2016.)

Training of feedforward neural networks is a three-step process. First, the output value of the network is computed. This step is known as the forward pass as the information flows through the network from the input layer to the output layer. After

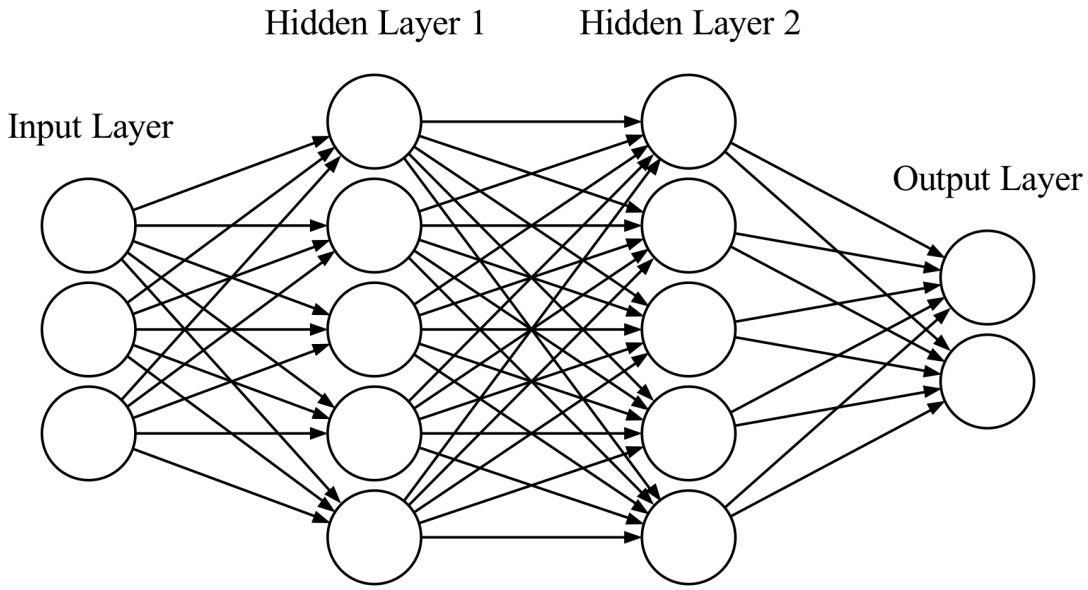

Figure 2.3: The structure of a basic feedforward neural network.

this, the output value is compared to the true target value by using a loss function to compute a cost of the error that the network made. In the third step which is known as the backward pass, the weights of the network are updated according to the loss value. This is typically done by using a gradient descent optimization algorithm which updates the weights by a small amount to the opposite direction of the gradient. In order to make calculating these changes possible, an algorithm known as backpropagation is used to traverse backwards through the network and compute the gradient. These steps are then repeated until the performance stops improving. (Goodfellow et al., 2016.) The components of feedforward neural networks are next discussed in greater detail.

Activation functions

There are many activation functions that are commonly used for feedforward neural networks. Typically, the activation functions of the hidden layers are the same for every layer, and the activation function of the output layer is chosen based on the type of the problem. (Goodfellow et al., 2016.)

Nair and Hinton (2010) introduced the rectified linear unit (ReLU)

$$g(z) = \max(0, z)$$

which is one of the most common activation functions of the hidden layers currently. The output of the ReLU is linear if the input is greater than 0, which helps training the network as the gradients stay large and consistent. However, if the values of the input vector $\boldsymbol{z} = \boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{b}$ are non-positive, the layer cannot learn anymore as the gradient is 0 in every dimension. Because of this, multiple modified versions of the basic ReLU have been created. These retain the good properties of the ReLU while alleviating the weaknesses. (Goodfellow et al., 2016.) Some of the commonly used modified versions include leaky ReLU

$$g(z) = \begin{cases} z & \text{for } z \geq 0 \\ 0.01z & \text{for } z < 0 \end{cases},$$

its generalization, parametric ReLU or PReLU which treats the coefficient as a learnable parameter

$$g(z) = \begin{cases} z & \text{for } z \geq 0 \\ \alpha z & \text{for } z < 0 \end{cases},$$

and exponential linear unit (ELU)

$$g(z) = \begin{cases} z & \text{for } z > 0 \\ \alpha(\exp(z) - 1) & \text{for } z \leq 0 \end{cases}$$

(Clevert et al., 2015; He et al., 2015; Maas et al., 2013).

Before ReLU and its modifications became popular, common activation functions of the hidden layers included the logistic sigmoid and hyperbolic tangent functions. These functions are sensitive only when the input is close to 0 as the output saturates to a constant value when the input is very positive and to another constant value when the input is very negative. These functions are still used in some applications, but the ReLU based functions have replaced them in the hidden layers of feedforward networks. (Goodfellow et al., 2016.)

There are three activation functions which are commonly used for the output layer. A linear function produces the output predictions

$$\hat{\boldsymbol{y}} = \boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{b}.$$

The linear output function is usually used for regression problems where the task is to output a value with no bounds. When the task involves predicting a binary variable y with a Bernoulli distribution, the predictions are given by a sigmoid output function

$$\hat{y} = \frac{1}{1 + \exp(-\boldsymbol{w}^\top \boldsymbol{x} - b)}.$$

The third commonly occurring case is predicting labels for a variable with a categorical distribution. The output function that is chosen for this problem type is known as the softmax function. It uses the unnormalized log probabilities $\boldsymbol{z} = \boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{b}$ to compute the output for each category using the function

$$\text{softmax}_i = \frac{\exp(z_i)}{\sum_j \exp(z_j)}.$$

(Goodfellow et al., 2016.)

Loss functions

The loss function provides the target value that is minimized in order optimize the neural network. In most simple cases, the cross-entropy between the predictions and

the training data is used as the loss function, as this results in the maximum likelihood solution. The exact form of the loss function then depends on the probability distribution of the model. For example, if the model distribution is assumed to be Gaussian, the mean squared error

$$\text{MSE} = \frac{1}{n}\sum_{i}^{n}(y_i - \hat{y}_i)^2$$

where $\boldsymbol{y}$ is the vector of true values and $\hat{\boldsymbol{y}}$ is the vector of predicted values, should be used as the loss function. Likewise, in two-class classification problems the binary cross-entropy

$$\text{BCE} = -y \log \hat{y} - (1 - y) \log (1 - \hat{y})$$

is the loss function that produces the maximum likelihood solution. (Goodfellow et al., 2016.)

Optimization

As mentioned earlier, neural networks are trained by optimizing the weights in a way that reduces the loss value. The key component for efficiently optimizing the weights of a feedforward neural network is the backpropagation algorithm which was popularized by Rumelhart et al. (1986). The algorithm is based on the chain rule of calculus

$$\frac{dz}{dx} = \frac{dz}{dy}\frac{dy}{dx}$$

which can be generalized for both vectors

$$\nabla_{\boldsymbol{x}^z} = (\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}})^\top \nabla_{\boldsymbol{y}^z}$$

and tensors

$$\nabla_{\mathsf{X}^z} = \sum_{j}(\nabla_{\mathsf{X}}\mathsf{Y}_j)\frac{\partial z}{\partial \mathsf{Y}_j}.$$

By applying the chain rule recursively backwards through the network starting from the loss function, the gradient can be computed with respect to every parameter of the network. (Goodfellow et al., 2016; Rumelhart et al., 1986.)

The network is trained by updating its weights against the gradient iteratively with small steps. However, computing the exact gradient based on the whole training set is computationally expensive, and training a neural network with a large data set would become infeasible if this was done. Instead, an estimate of the true gradient, $\boldsymbol{g}$, is computed by sampling a small batch of data from the training set during each iteration. The weights are then updated based on the estimate of the true gradient:

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \epsilon \boldsymbol{g}$$

where ϵ represents the learning rate which determines the length of the steps taken against the gradient. This simple optimization algorithm is known as stochastic gradient descent (SGD). The learning rate of SGD is an important parameter: if it is too low, the training of the network is slow, and if it is too high, the loss function may not converge to a local minimum. The learning rate is usually decreased over the duration of the training in order to make sure that the algorithm finds a local minimum. (Goodfellow et al., 2016; Rumelhart et al., 1986.)

One of the reasons for the difficulty of training neural networks is that the optimization problems are highly non-convex. Because of this, the learning algorithm most likely converges to a local minimum instead of the global minimum. (Goodfellow et al., 2016; Gori and Tesi, 1992; Sontag and Sussmann, 1989.) However, it has been proved that the local minima do not cause problems for large-size neural networks as they all have quite low test errors (Choromanska et al., 2014). There are still some other problems with the optimization process: for example, the Hessian matrix may be ill-conditioned which causes the loss to increase even with a small learning rate, there can be regions where the gradient is extremely flat or steep, and the initial choice of the parameters can have a large effect on the result of the optimization (Goodfellow et al., 2016).

There are a great number of more advanced optimization algorithms based on the SGD that aim to converge faster and avoid some of the problems that the

regular SGD may encounter. One way to improve the learning speed of SGD is by using a momentum algorithm, discovered by Polyak (1964), which gives weight to the previous gradients in addition to the current one. The algorithm updates the weights in a two-step process:

$$\boldsymbol{v} \leftarrow \alpha \boldsymbol{v} - \epsilon \boldsymbol{g}$$

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \boldsymbol{v}$$

where $\boldsymbol{v}$ is known as the velocity and α is a coefficient that gives the weight for the previous gradients. (Goodfellow et al., 2016; Polyak, 1964.) A common alternative for the regular momentum algorithm is known as the Nesterov momentum algorithm. It is very similar to the momentum algorithm but it computes the estimated gradient $\boldsymbol{g}$ at the position $\boldsymbol{\theta} + \alpha \boldsymbol{v}$ instead of using the current parameters $\boldsymbol{\theta}$. (Goodfellow et al., 2016; Nesterov, 1983.)

Using optimization algorithms with adaptive learning rates for every parameter has been another strategy for improving the performance of SGD (Goodfellow et al., 2016). For example, the AdaGrad algorithm and its improved version RMSProp, discovered by Duchi et al. (2011) and Hinton (2012) respectively, achieve great results by updating the learning rates after each iteration. Some of the newer optimization algorithms, such as Adam and Nadam, combine the properties of adaptive learning rate algorithms with the concept of momentum. Adam incorporates momentum and some minor changes to the RMSProp algorithm, and Nadam modifies Adam to use Nesterov momentum. (Dozat, 2016; Kingma and Ba, 2014.)

Adding so called batch normalization layers to the model is a very commonly used method for achieving faster and more reliable optimization. Batch normalization first computes the mean and variance of its inputs over the batch of data used for the current training iteration. These values are then used to first normalize and

then scale and shift the input:

$$y_i \leftarrow \gamma \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} + \beta$$

where y_i is the output, x_i is the input, μ_B is the computed mean, σ_B^2 is the computed variance, ϵ is a small constant, and γ and β are learnable parameters. At test time, μ_B and σ_B^2 are replaced by population statistics. (Ioffe and Szegedy, 2015.) Originally, batch normalization was thought to improve the optimization performance by reducing a phenomenon known as internal covariate shift, but later it has been shown that at least most of the benefits are a result of smoother gradients. (Ioffe and Szegedy, 2015; Santurkar et al., 2018)

Regularization

When a neural network model is trained, it is most important that the model performs well on new data, which is typically measured by computing the error rate on a test set that has not been used for the training. There are many so called regularization strategies that aim to improve only the generalization performance of the model. (Goodfellow et al., 2016.)

One way to lower the generalization error is by avoiding overfitting. A common strategy for achieving this is using parameter norm penalties which limit the capacity of the model by punishing high parameter values. A regularizing term $\lambda\Omega(\boldsymbol{\theta})$, where λ is a parameter that scales the regularization strength and $\Omega(\boldsymbol{\theta})$ is the norm penalty function, is added to the loss function. Common norm penalty functions include L_2 regularization, where $\Omega(\boldsymbol{\theta}) = \frac{1}{2}||\boldsymbol{\theta}||_2^2$, and L_1 regularization, where $\Omega(\boldsymbol{\theta}) = ||\boldsymbol{\theta}||_1$. Compared to L_2 regularization, L_1 regularization results in more parameters acquiring value of 0 during the training. (Goodfellow et al., 2016; Hastie et al., 2009.)

Early stopping is another very common regularization strategy that aims to avoid overfitting the model. It works by monitoring the validation set performance of the

model during the training, and storing the parameter values every time the performance improves. When the validation set error starts increasing due to overfitting, the training is stopped, and the optimal parameters are returned. In order to accurately conclude that the validation performance has started decreasing, a value known as patience is used to decide how long the training continues after the last improvement before stopping. (Bengio, 2012; Goodfellow et al., 2016.)

As the generalization performance of a neural network is very often limited by the amount available training data, an effective method of improving the results in some cases is creating more data for the training. This technique is known as data augmentation, and it is useful especially for image classification tasks. Applying randomized modifications such as rotation, translation and noise injection to image data can produce a much larger training set and thus help reduce the generalization error. (Goodfellow et al., 2016.)

Dropout is a regularization method which, as its name suggests, drops out some random set of the nodes of the network during each training iteration. By doing this, the nodes cannot co-adapt to the training data as some of the nodes are always disconnected from the others. Dropout can be thought as a computationally effective way to average many different pruned models. At test time, all nodes are connected again, but the weights are scaled down to approximate the combination of the trained models. (Srivastava et al., 2014.)

2.2.3 Convolutional neural networks

Convolutional neural networks (CNNs) are a specific type of neural networks that use an operation known as convolution instead of normal matrix multiplication in at least some of the layers (Goodfellow et al., 2016). The history of CNNs dates to the work of Fukushima (1980). The studies of Hubel and Wiesel (1962), who discovered how the mammalian visual nervous system functions, served as an inspiration for the

first CNN model. The CNN models, as well as the hardware used for the training, developed quite slowly over the decades until the 2010s (Schmidhuber, 2015). After Ciresan et al. (2011) and Krizhevsky et al. (2012) achieved remarkable performance levels on image classification tasks, the popularity of feedforward CNNs exploded. These days CNNs are used especially for image-related tasks with great success (LeCun et al., 2015).

Convolution

Convolution, denoted with the symbol $*$, is a mathematical operation between two functions f and g:

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

or in the discrete case

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

The first function f is usually called the input and the second function g the kernel. The convolution operation can be interpreted as a procedure that flips the kernel, slides it along the input, and outputs a function where the value at point t corresponds to the overlap of the input and the flipped kernel that is shifted by t. Typically, CNNs do not use the regular convolution but rather the cross-correlation which is a non-commutative version of convolution that does not flip the kernel. In addition, when processing image data, the cross-correlation operation is usually two-dimensional:

$$(f \star g)(i,j) = \sum_{m} \sum_{n} f(i+m, j+n)g(m,n).$$

(Goodfellow et al., 2016.)

When compared to the layers of a regular feedforward neural network, convolutional layers have a few major differences. The weights that are optimized during

training of the convolutional layers are the values of the kernel. The size of the kernel is typically much smaller than the size of the input, which means that there are fewer connections in the network. This benefits the computational efficiency and reduces the memory requirements. As the same kernels are used at every location of the inputs, the weights are shared, and as a result the memory required for storing the weights becomes even smaller. Another consequence of the shared weights is that CNNs are equivariant to translation: for example, if an object is moved in an input image, its representation in the output will move by the same amount. (Goodfellow et al., 2016; LeCun et al., 2015.)

In practice, the convolutional layers almost always use multiple kernels in parallel to extract different kinds of features from the inputs. In order to further reduce the computational cost of training the network, the output of a convolutional layer can be downsampled by specifying a stride s so that the convolution is computed at only every sth location of the input. (Goodfellow et al., 2016.)

Pooling

In most CNN architectures, the outputs of the convolutional layer are fed into an activation function just like in regular feedforward neural networks. However, after this step most CNNs add a pooling function. This function merges its inputs into a summary statistic and thus introduces invariance to local translations. The most used pooling function is called max pooling, and it simply outputs the maximum value found inside its input. Pooling is often combined with downsampling by adding a stride. (Goodfellow et al., 2016; LeCun et al., 2015.) The effect of max pooling with striding is illustrated in Figure 2.4.

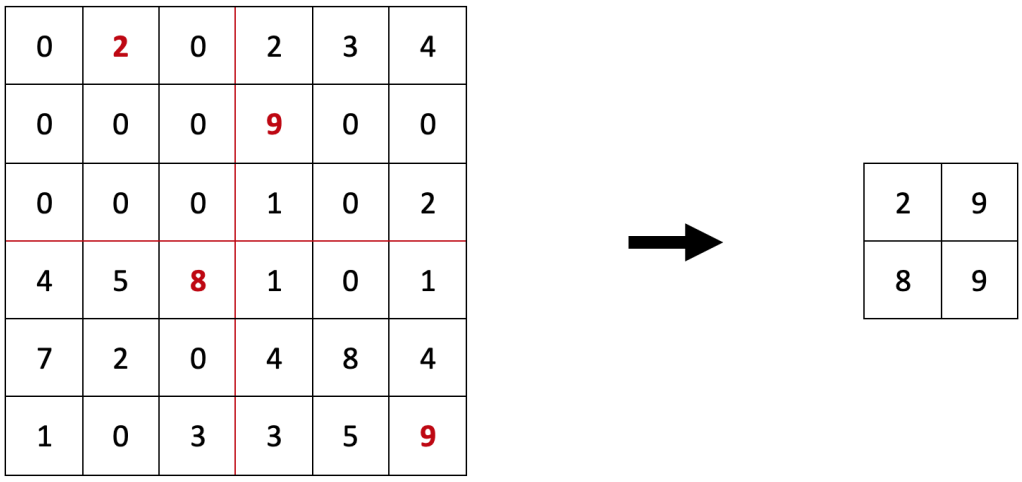

Figure 2.4: The effect of a downsampling max pooling operation with input size of 3×3 and stride of $(3, 3)$ applied to a grid of size 6×6. For the output, max pooling picks the maximum value from each input rectangle, and these rectangles are made non-overlapping with the stride.

Deep convolutional neural networks

As CNNs are computationally more efficient than fully connected feedforward neural networks, they can typically use more layers, which in turn seems to produce better results. CNNs learn to combine hierarchical structures within the data. For example, in the case of image data, the kernels of the first layer start to recognize simple colors and edges, and these features are combined in the later layers to recognize more and more complex patterns. (Goodfellow et al., 2016; LeCun et al., 2015.)

2.2.4 Image segmentation

Deep convolutional neural networks can be used for many different types of image processing tasks. These tasks include, for example, image classification, object detection, semantic segmentation, and instance segmentation. Image classification tasks are about predicting whether an object is or is not present in the given image. In object detection tasks, it is important to not only correctly predict that an object of some type is present in the image but finding the locations of the objects is also required. The locations are usually marked by boxes surrounding the objects. In semantic segmentation tasks, every pixel of the image is assigned to a class. Detection of separate objects is not deemed important in these problems. Instance segmentation tasks are a combination of object detection and semantic segmentation: every individual object is detected and segmented on pixel-level. (Lin et al., 2014.) The differences of these task types are explained with examples in Figure 2.5.

While the different types of image recognition tasks may sound similar, the popular CNN architectures used for them can differ from each other. A good performance can be achieved on image classification tasks with CNNs that have several convolutional layers followed by fully connected layers and the softmax function that outputs the probabilities of detecting each possible class in the image (Krizhevsky et al., 2012). A network architecture known as R-CNN, as well as its more recent im-

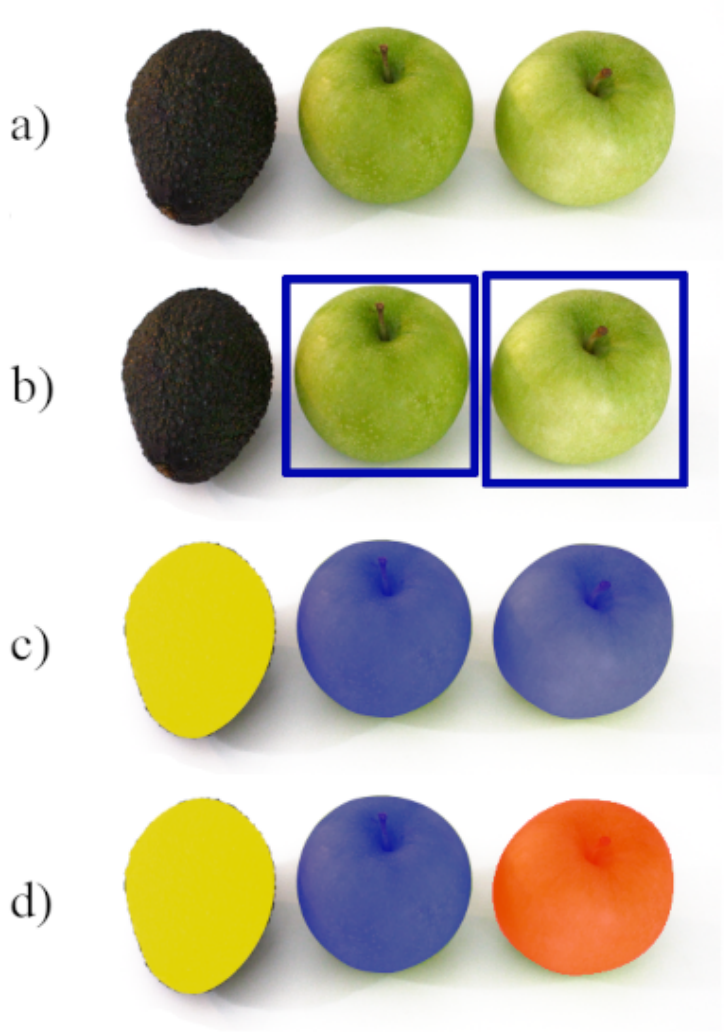

Figure 2.5: Different types of image recognition tasks. In image classification, the task could be to predicted whether the original image a) contains an apple. In object detection, the locations of the apples would be detected as shown in image b). In semantic segmentation, the different fruits would be segmented into distinct classes. For example in image c), the apples are segmented into one class and the avocado into another. Finally, in instance segmentation all individual instances of the fruits would be segmented, as shown in image d).

proved versions like Faster R-CNN, are popular for object detection tasks. R-CNN works by first using an algorithm to create proposals of regions that may contain objects, then extracting features from these regions with a CNN that is pre-trained on an image classification task, and finally using a linear support vector machine to classify each region proposal (Girshick et al., 2014). Faster R-CNN is a model based on the principles of the original R-CNN. Instead of an region-proposal algorithm, it uses a fully convolutional neural network module to produce the region proposals. These proposals are then used by another CNN module that classifies the regions. These modules share convolutional layers, which makes the object detection computationally inexpensive. (Ren et al., 2017.) By adding a fully convolutional module to the Faster R-CNN for predicting object masks, He et al. (2017) created a model known as Mask R-CNN, which is a very popular tool for instance segmentation tasks. Semantic segmentation tasks are often solved with fully convolutional neural networks. U-net, which was created by Ronneberger et al. (2015), is a very widely used fully convolutional model, and as it is used in this thesis, it will be discussed in greater detail next.

U-net

U-net is a fully convolutional neural network model that was originally designed for semantic segmentation of biomedical images. The model consists of a contracting path and an expansive path. The contracting path is made up of blocks that have two consecutive convolutional layers with kernel sizes of 3×3, and a 2×2 downsampling max pooling layer with a stride of 2. The convolutional layers are followed by nonlinear activation functions, and the first convolutional layer of each block increases the number of convolutional kernels used. The expansive path consists of an equal number of blocks. These blocks first use an operation known as up-convolution to upsample the input and undo the effects of the corresponding max pooling layer in

the contracting path. The output of the up-convolutional layer is then concatenated with the output of the corresponding contracting block. The block finally uses two normal convolutional layers with 3×3 kernels. The first convolutional layer reduces the number of kernels, and each are again followed by a nonlinear activation function. After the desired number of contracting and expansive blocks are stacked on top of each other, a 1×1 convolution is used to map the final output to the correct number of classes. (Ronneberger et al., 2015.)

Performance metrics for image segmentation tasks

The segmentation accuracy of a model is often evaluated and compared to the performance of other models by using certain metrics that try to quantify the goodness of the predictions.

Pixel accuracy is a simple metric that tells the percentage of correctly classified pixels:

$$\text{PA} = \frac{\sum_{i=1}^{K} p_{ii}}{\sum_{i=1}^{K} \sum_{j=1}^{K} p_{ij}}$$

where K is the number of classes and p_{ij} is the number of pixels belonging to class i that are predicted to belong to class j. (Minaee et al., 2020.)

Precision and recall are metrics that are commonly used for evaluating many kinds of machine learning models, and they are sometimes used for binary image segmentation tasks as well:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

and

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

where TP is number of true positives, FP is the number of false positives, and FN is the number of false negatives. (Minaee et al., 2020.)

Dice similarity coefficient (DSC) and intersection over union (IoU), which is also known as Jaccard index, are some of the most popular metrics for image segmentation tasks:

$$\text{DSC} = \frac{2|X \cap Y|}{|X| + |Y|}$$

and

$$\text{IoU} = \frac{|X \cap Y|}{|X \cup Y|}$$

where X and Y are the true and predicted segmentation maps. These two values are positively correlated with each other. (Minaee et al., 2020.)

3 Related work

3.1 Medical image segmentation with neural networks

Over the last couple of years, deep convolutional neural networks have been applied to a large number of medical image analysis tasks. These tasks include classification, localization and segmentation of organs, lesions or other objects of interest, as well as many other types of tasks. The CNN models have been successfully applied to many different medical imaging procedures such as MRI, computed tomography scans and regular X-ray images. (Litjens et al., 2017.) In this thesis, the focus will be on the segmentation of prostate cancer lesions from magnetic resonance images.

Segmentation of the whole prostate from magnetic resonance images by using CNNs has been a popular area of research for years. One contributor to the popularity has been The Prostate MR Image Segmentation challenge PROMISE12, for which a quite large set of segmented prostate MR images were made freely available for research. (Litjens et al., 2014.)

Next, some of the research more directly related to the topic of this thesis will be reviewed. However, as noted by Litjens et al. (2014), the results of different studies can be very difficult to compare for several reasons. First of all, most models are not freely available and are typically very difficult to implement based on the information available in the research papers. Secondly, the MR images between

research groups can greatly differ from each other as variables such as the acquisition protocol, magnetic field strength and scanner type can have a large effect on the image appearance. (Litjens et al., 2014.)

3.2 Segmentation of prostate cancer lesions

Several studies have used CNNs to segment the cancerous lesions from prostate MR images over the years. Schelb et al. (2019) used a modified U-net to segment both T_2w images and ADC maps into three classes: clinically significant prostate cancer lesions (Gleason score $\geq$ 7), prostate tissue, and background. The segmentation performance of the model was evaluated with DSC, and the segmentation of clinically significant lesions received scores of 0.37 for ADC maps, 0.34 for T_2w images, and 0.35 for the combination of these. The study included prostate MR images of 312 men which were divided into a training set of 80% and a test set of 20%.

Kohl et al. (2017) approached the problem with adversarial training, in which the segmentation network competes with another neural network that tries to produce real-looking fake images. The used segmentation network was a modified U-net model. The data set included T_2w images, ADC maps, and DWIs with a b value of 1500 s/mm^2 for 152 patients, of which 55 had a Gleason score of 7 or higher. The DSC of the clinically significant lesions was found to be 0.41 $\pm$ 0.28.

In the studies of Dai et al. (2019), Mask R-CNN was used to segment prostate tissue as well as prostate cancer lesions. For the segmentation of the lesions, T_2w images and ADC maps of 120 men were used. These were laid on top of each other with a registration algorithm and fed into the network. The resulting DSC for the lesion detection was 0.46 $\pm$ 0.20.

Simultaneous automatic segmentation and classification of prostate cancer lesions from magnetic resonance images has seen little if any published research that is directly comparable to this study.

4 Materials and methods

4.1 Data set

For this study, the axial T_2w images and ADC maps as well as the corresponding prostate gland and prostate cancer lesion masks of the MRI data set introduced by Jambor et al. (2017) were used. The magnetic resonance images were obtained from 162 of the 175 men initially enrolled in the study between March 2013 and February 2015. All these men had a suspicion of prostate cancer as their prostate-specific antigen levels were between 2.5 and 20.0 ng/ml and/or they had had an abnormal digital rectal examination. (Jambor et al., 2017.)

The images were obtained using a 3T MR scanner (Verio, Siemens, Erlangen, Germany) with surface coils. The imaging time varied between 14 and 17 minutes per patient, but this value includes capturing T_2w images in the sagittal plane as well as DWIs with higher b values of 1500 and 2000 s/mm^2. The ADC maps used in this study were based on the axial DWI data obtained with b values of 0, 100, 200, 300 and 500 s/mm^2. The acquisition voxel size was $2.0 \times 2.0 \times 3.0$ mm^3 without intersection gaps. The transmission repetition time was 5543 ms and the transmission echo time was 80 ms. For the axial T_2w images used in this study, the voxel size was $0.6 \times 0.6 \times 3.0$ mm^3, the transmission repetition time was 6400 ms and the transmission echo time was 101 ms. (Jambor et al., 2017.)

Based on the images, prostate cancer lesion masks were delineated and classified

using the biparametric MRI Likert scoring system by a reader with 5 years of experience in prostate MRI. Targeted biopsy was performed for patients with suspicious lesions based on the Likert scores, and systematic 12-core biopsy was performed for all patients. These samples were analyzed separately by two genitourinary pathologists with over 5 years of experience in genitourinary pathology, and Gleason scores were assigned for each patient. (Jambor et al., 2017.)

4.2 Analysis platform

Most of the computationally expensive image analysis in this study was performed on an NVIDIA TITAN V graphics processing unit using version 2.0 of TensorFlow (Abadi et al., 2015).

4.3 Data preparation

The used data set contained multiple T_2w image and ADC map slices of 162 patients. For each of these patients, there were 0–3 prostate cancer lesion masks, which had been assigned a Gleason score based on targeted biopsy and 12-core systematic biopsy. In the cases where one of the Gleason scores obtained by performing biopsy was higher than the Gleason score of the dominant lesion, the Gleason score of that lesion was replaced with the higher score obtained with biopsy.

After assigning the Gleason scores for each lesion mask, the scores were grouped using the standard ISUP Gleason grade grouping system. The distribution of the highest Gleason group of each patient is displayed in Table 4.1.

The patients were split into stratified training and test sets with 70% and 30% of the patients respectively. The split was done on the level of patients instead of single prostate image slices in order to avoid leaking information from the training set to the test set, and the stratification ensured similar distributions of Gleason score

Table 4.1: Number of cases per ISUP Gleason grade grouping system. Each case is assigned to the group according to its highest Gleason group score.

ISUP grade group	Number of cases
Not assigned	63
1	14
2	37
3	23
4	10
5	15

groups in both sets. When training the models, 20% of the training set samples were further split into a separate validation set.

4.4 Prostate segmentation

The popular U-net architecture (Ronneberger et al., 2015) was chosen as the basis of the deep convolutional neural network models that were used for the image segmentation tasks. Before delving into the more difficult task of prostate cancer lesion segmentation, the model architecture was validated by segmenting the entire prostate glands.

The original sizes of the ADC images (128×128 pixels) and the T_2w images (256×256 pixels) were cropped to 64×64 and 128×128 pixels respectively by removing the areas of tissue surrounding the prostate from the images. This was done mainly in order to reduce training time. The values of the images were scaled to the closed interval [0, 1] by simply dividing the values by 4095 which is the maximum value of the 12-bit images.

Separate models were created for ADC maps and T_2w images. The general architecture of the tested models is displayed in Figure 4.1. For T_2w images, the

model architecture consisted of a contracting path with 4 blocks, a single middle block, and an expansive path with 4 blocks. Each of these blocks was made up of two 2-dimensional convolution layers with a kernel size of $3 \times 3.$, followed by a batch normalization operation. The blocks of the contracting path included a 2×2 max pooling operation and a dropout layer after the convolutions. The blocks of the expansive path first performed an up-convolution to upsample the input. After this, the result was concatenated with the corresponding feature map of the contracting path and a dropout operation was performed before feeding the result to the two 2-dimensional convolution layers. The number of network neurons was increased by a factor of 2 at each block of the contracting path, and respectively decreased by a factor of 2 at the blocks of the expansive path. A 1×1 convolution followed by a sigmoid activation function was performed at the final layer. All other convolution layers used a rectified linear unit as the activation function, and the initial values of the convolution layers were drawn from a Gaussian distribution with a mean of 0 and variance computed as suggested by He et al. (2015). The ADC models were identical to the T_2w models except that the contracting and expansive paths had only 3 blocks each because of the smaller image size.

As the segmentation problem is essentially a two-class classification problem, binary cross-entropy was used as the loss function. However, when calculating the loss, the pixels of the prostates were weighted by the ratio between background pixels and prostate gland pixels in the training set. This was done because the images contained significantly more background areas than pixels belonging to the prostate glands. The used optimization algorithm was Nesterov-accelerated Adaptive Moment Estimation (Nadam), first introduced by Dozat (2016).

As a simple form of data augmentation, the input images were flipped horizontally with a 50% probability when training the model. A few models with varying number of neurons were trained. Not a lot of time was spent on the optimization

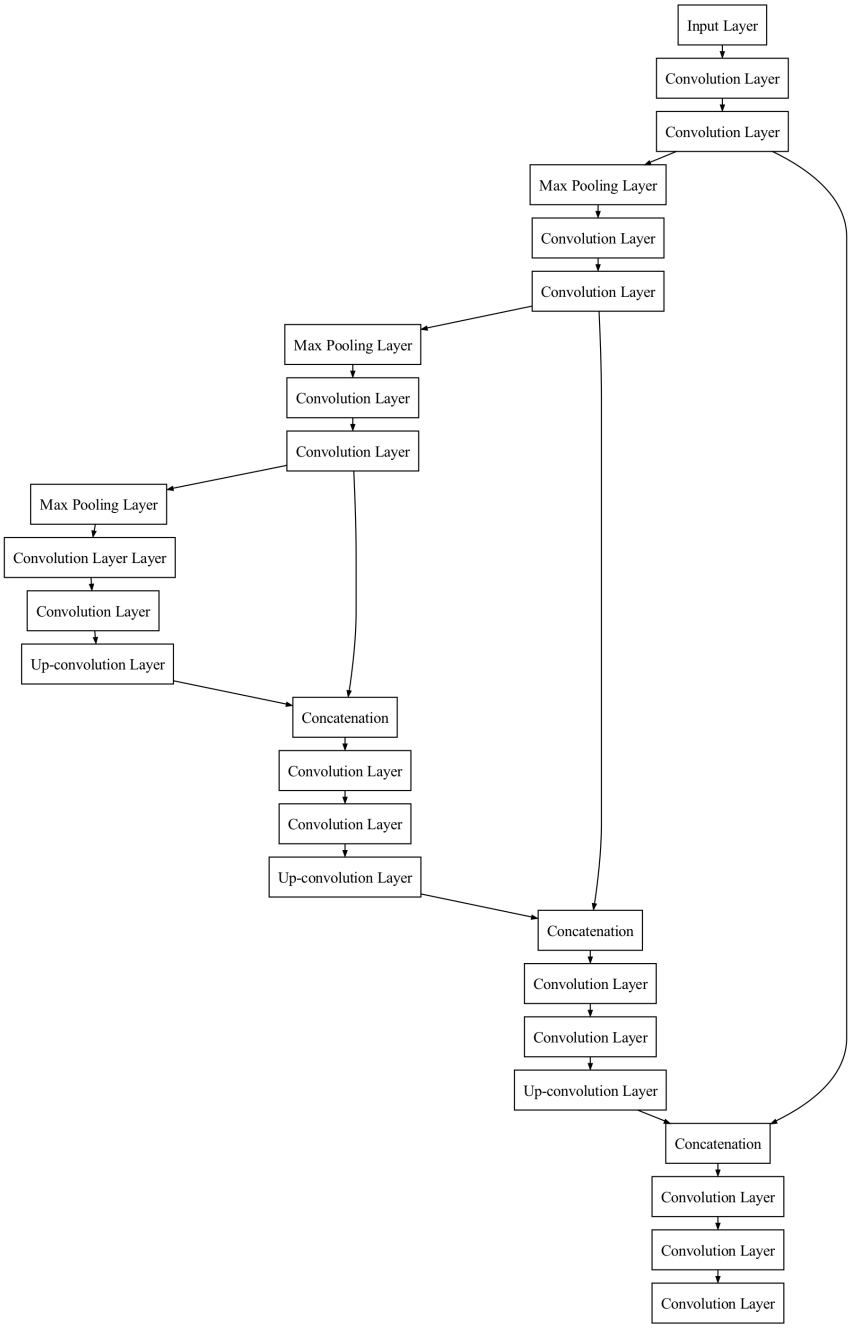

Figure 4.1: The general structure of used U-net models. The batch normalization and dropout layers are not shown in the graph to keep it simpler.

of these models because the purpose of this experiment was simply to validate the usability of the U-net architecture. The performance of the trained models was evaluated on the validation set before finally evaluating the best models using the separate test set. The used metrics included DSC, IoU, percentage of correctly classified prostate gland pixels, and percentage of correctly classified background pixels.

4.5 Binary lesion segmentation

As the next step before moving to actual multi-class prostate cancer lesion segmentation, a slightly easier task of binary lesion segmentation was attempted by building onto the models created for prostate gland segmentation. Because some of the lesions with lower ISUP grades can be difficult to detect, group 3 was chosen as the cut-off value; thus, the task became finding lesions that belong to ISUP grade group 3, 4, or 5.

The images were preprocessed using the same steps as described in the prostate segmentation section, and the model had the same structure as well. However, for this task, introducing small random changes to the pixel intensities was tested as a data augmentation step in addition to random horizontal flipping of the images. The random values were drawn from a zero-centered normal distribution with a standard deviation 0.02. An advanced data augmentation technique known as elastic distortion (Simard et al., 2003) was also tested since it deforms the images in a more natural way than uniform shearing, rotation or translation. In order to combat overfitting, adding both L_1 and L_2 regularization to all convolutional and up-convolutional layers of the U-net model was assessed.

In addition to training the models with regular two-dimensional images, incorporating three-dimensional information was attempted by feeding groups of three adjacent image slices as the inputs. This was done by choosing the slices below and

above the central slice, and if the central slice was the first or last of the entire stack, it was duplicated to fill the missing place. Ideally in order to use all available spatial information, the entire stack of slices should have been fed to the model at once but because different cases included different number of slices, this was not possible.

The same weighted binary cross-entropy loss function was used as in the prostate segmentation task. It was anticipated that the function could have problems with sometimes incorrectly classifying especially ISUP grade 2 lesions as positives, and these false positive cases could penalize the model although even trained professionals could struggle with classifying the images correctly with the available information. To combat this issue, a customized Dice loss function was tested as well:

$$\text{Dice loss} = \frac{2\sum(X \odot Y \odot M) + 10^{-6}}{\sum X + \sum Y + 10^{-6}}$$

where X and Y are the true and predicted segmentation maps, and M is a binary mask where ISUP grade 2 lesions have value 0 and all other pixels have value 1.

The best models were selected based on their DSC and IoU performance on the validation set, and subsequently these models were evaluated on the test set using the same metrics.

4.6 Multi-class lesion segmentation

As the final task, detection and automatic classification of the magnetic resonance prostate images was attempted. The goal of this experiment was to detect cancerous lesions and assign an ISUP group correctly to the detected areas.

The structure on the tested models was generally the same as in the previous section with the small change of using a 5-dimensional output layer instead of one that produces a single output value. Instead of only training the models from scratch, using the parameters of the best-performing binary lesion segmentation models as

Table 4.2: The used ordinal encoding system of the lesion masks compared to regular one-hot encoding. The benefit of ordinal encoding is that it preserves the ordinal nature of the lesion groups.

ISUP grade group	One-hot encoding	Ordinal encoding
Not assigned	00000	00000
1	10000	10000
2	01000	11000
3	00100	11100
4	00010	11110
5	00001	11111

the starting point was tested as well.

Rather than of using one-hot encoding for lesions masks, which is the default option in most multi-class segmentation applications, the ordinal nature of Gleason score groups was preserved by converting the masks to use ordinal encoding, shown in Table 4.2.

The same data augmentation methods were used as for the binary segmentation task. In addition to these, some models were trained with simpler data augmentation techniques such as rotating, shifting, shearing and zooming the image by a relatively small amount.

The tested loss functions included a modified Dice loss function which summed the Dice scores of all five ISUP grade groups, as well as variations of weighted binary cross-entropy, some of which gave different weights to the edges of the lesions.

Because the task included in essence five different optimization tasks, one for segmentation of each ISUP grade group, the end result depended very heavily on the possible weighting of the different subtasks. In real-world applications it could be beneficial to give higher weights to clinically significant lesions at the cost of decreasing performance for non-significant lesions, but in this case such weighting

was not done because the possible applications of the study results did not have any specific requirements.

The performance of the models was evaluated with DSC and IoU as well as pixel-level precision and recall metrics, this time by calculating the values separately for each ISUP grade group.

5 Results

5.1 Prostate segmentation

In order to validate that the U-net architecture works for segmentation tasks with the prostate images, the entire prostate glands were segmented. The training progression of the models was evaluated by investigating how the binary cross-entropy loss and several metrics evolved over the training duration. The progression of the loss function value is shown in Figure 5.1, and the development of calculated DSC metric is displayed in Figure 5.2.

After using the validation set DSC to select the models with the best performance for both ADC and T_2w segmentation tasks, the final models were evaluated on the separate testing set. The evaluated metrics are displayed in Table 5.1. An example of model predictions is shown in Figure 5.3.

Table 5.1: Test set performance of the U-net models for segmenting prostate glands from ADC and T_2w images.

Metric	ADC	T_2w
Dice similarity coefficient	0.869	0.923
Intersection over union	0.768	0.864
Correctly predicted gland pixels	87.4%	94.2%
Correctly predicted background pixels	99.1%	98.9%

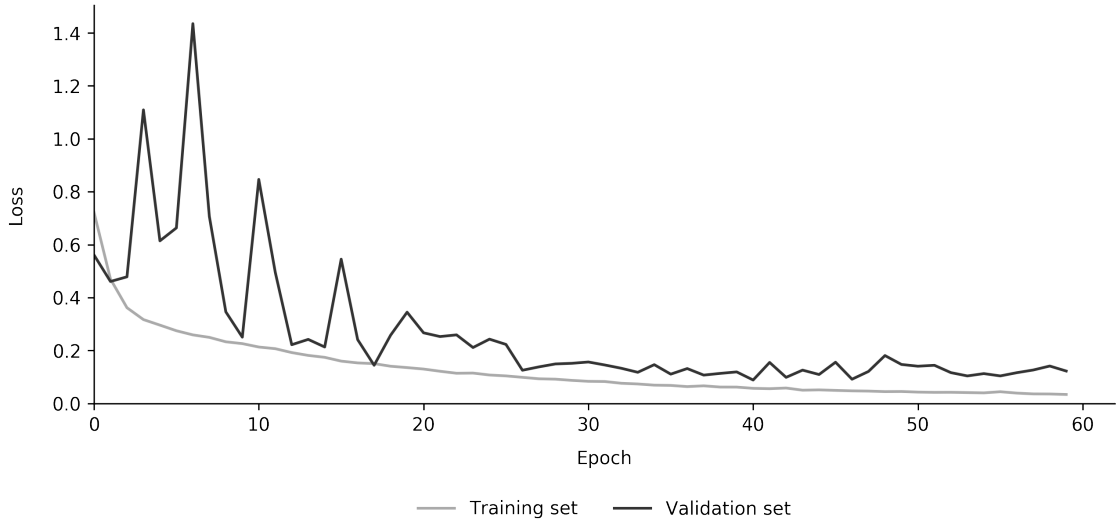

Figure 5.1: The development of weighted binary cross-entropy loss function value over the training duration of a model that segments prostate glands from ADC maps. The loss function is evaluated separately on the training and validation sets.

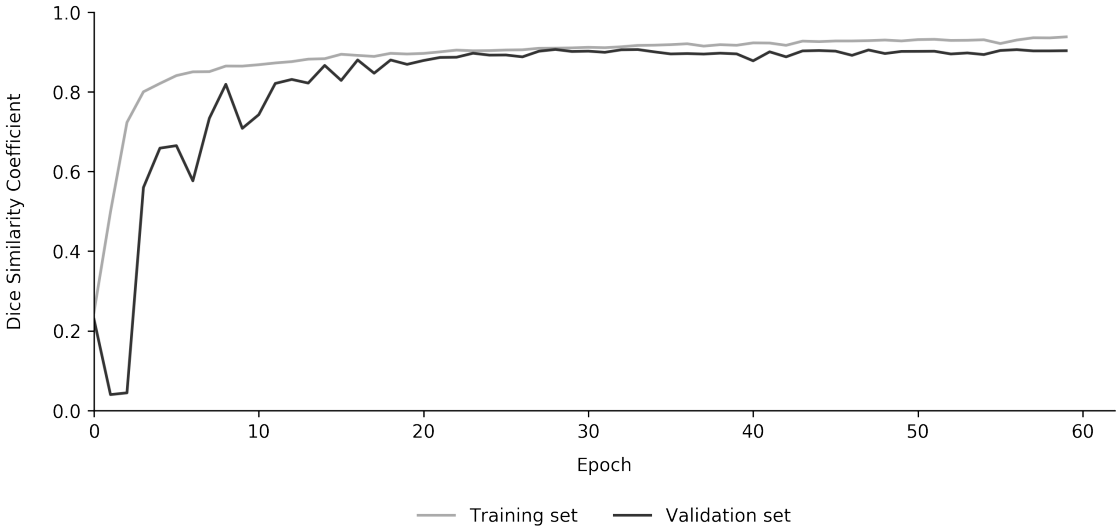

Figure 5.2: The development of Dice similarity coefficient metric over the training duration of a model that segments prostate glands from ADC maps.

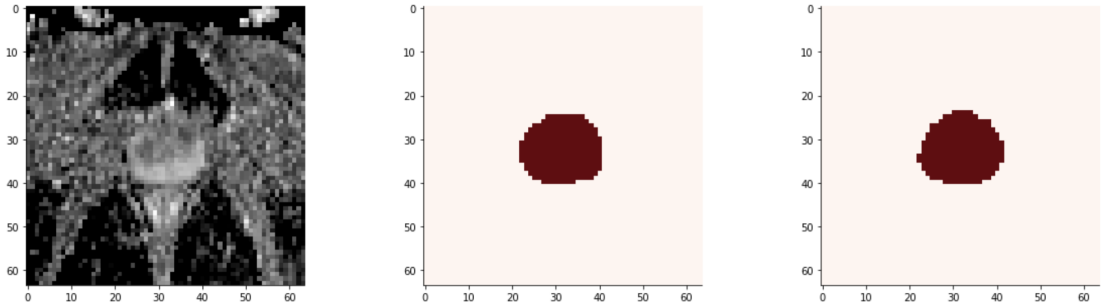

Figure 5.3: An example of prostate gland segmentation results. The original ADC test set image is on the left, the true prostate mask delineated by a professional is in the middle, and the mask predicted by the model is on the right.

5.2 Binary lesion segmentation

As the next step after prostate segmentation, segmentation of prostate cancer lesions belonging to ISUP grade group 3, 4 or 5 was attempted.

By comparing the validation set metrics, it was determined that best performance was reached by models that used three-dimensional images instead of two-dimensional ones, and Dice loss instead of weighted binary cross-entropy. In addition, using elastic transformation for data augmentation and L_1 and L_2 regularization for overfitting reduction proved to be efficient methods for improving the performance. However, as seen in Figure 5.4, the validation set performance could not reach the training set performance even with the models that used very heavy regularization. The best-performing model combined all these techniques and reached a DSC of 0.389 and IoU of 0.241 on the validation set. When evaluated on the test set, DSC was measured at 0.342 and IoU at 0.206.

Interestingly, the identical models had huge problems with T_2w images. While the training set DSC stabilized at around 0.6, the models struggled with the validation set: the DSC seemed to get stuck bouncing between 0 and 0.2 with all

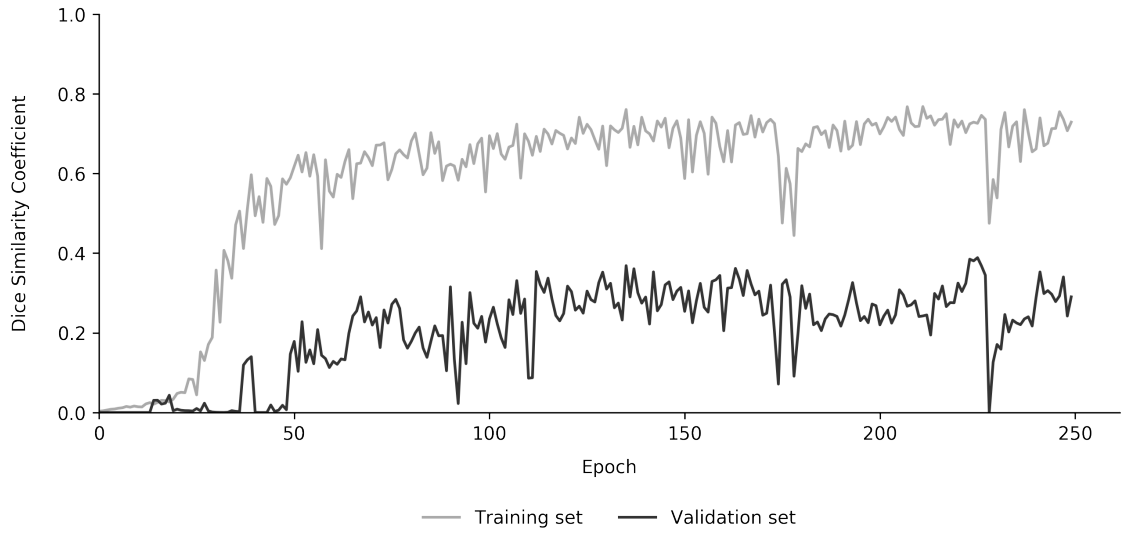

Figure 5.4: Dice similarity coefficient measured on the training and validation sets over the training duration of a model that segments prostate cancer lesions with ISUP grade group 3 or higher.

models.

5.3 Multi-class lesion segmentation

The main goal of this study was to create a deep learning model which can both segment prostate cancer lesions and at the same time assess the clinical significance of the found lesions by assigning an ISUP grade group to them.

The final ADC model that was chosen based on its combined DSC performance on the validation set used the parameters of the best-performing model from the last section as its starting point. The model was trained using the simple loss function that summed the five different Dice scores. The training progress of this model can be seen in Figure 5.5. The validation set DSC values reached averages of around 0.4 but the variance between epochs was very high.

Because the final model displayed significant increase of variance towards the

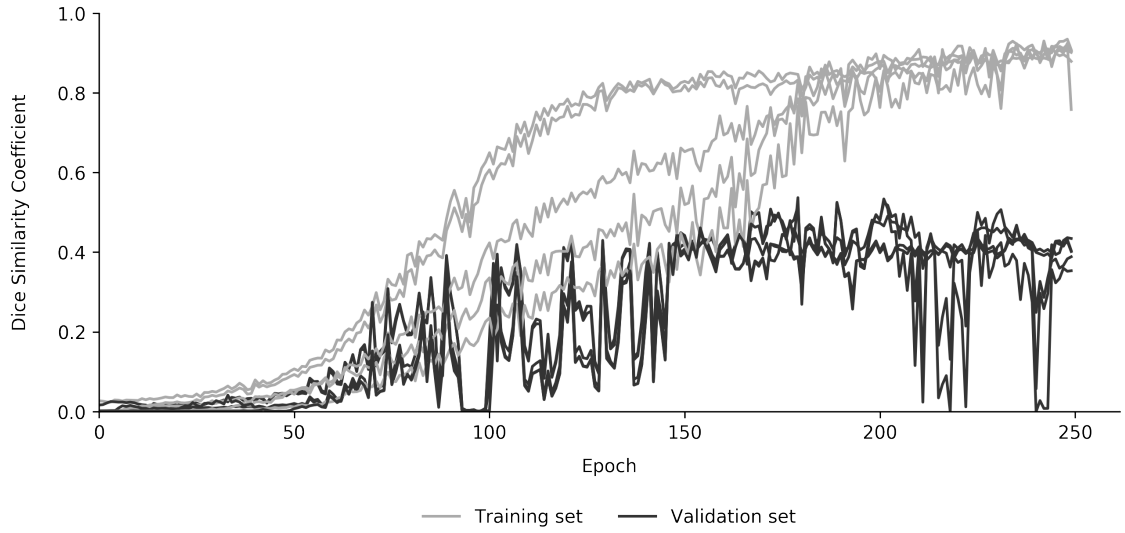

Figure 5.5: Dice similarity coefficient values measured on the training and validation sets over the training duration of a model that segments prostate cancer lesions and assigns ISUP grade groups to them. The five different lines represents represent values for lesions that belong at least to ISUP grade group 1 to 5.

Table 5.2: Dice coefficient factors, intersection over union, pixel-level precision and pixel-level recall of the final ADC segmentation and classification model as measured on the test set. Model version A represents the performance after the first 150 training epochs and version B the performance after all 250 epochs.

	Version A				Version B			
ISUP group	DSC	IoU	Precision	Recall	DSC	IoU	Precision	Recall
≥ 1	0.297	0.174	0.628	0.195	0.389	0.242	0.523	0.310
≥ 2	0.319	0.190	0.628	0.214	0.398	0.249	0.539	0.316
≥ 3	0.388	0.241	0.549	0.300	0.268	0.155	0.576	0.175
≥ 4	0.379	0.233	0.406	0.355	0.180	0.099	0.566	0.107
5	0.234	0.133	0.199	0.285	0.049	0.025	0.114	0.031

end of the training process, the model was evaluated on the test set at two different training points: version A of the model represented the performance after the 150 epochs and version B the highest performance after all 250 training epochs, as measured on the validation set. The DSC, IoU, pixel-level precision, and pixel-level recall values can be seen in Table 5.2. The performance of model A was much more balanced while model B managed to improve the performance on the first two ISUP grade groups while greatly sacrificing predictive capability of the groups 3, 4 and 5.

Just as in the binary segmentation task, the models trained on the T_2w images were not adequate for reliable analysis.

6 Discussion

6.1 Prostate segmentation

The deep learning models that were created for segmenting whole prostate glands from ADC and T_2w images achieved excellent performance. For example, the DSC of T_2w predictions matched some of the best models described by Gillespie et al. (2020). The highest reported DSC of 0.943 in that study is only a bit higher than the one achieved here. Similarly, Zabihollahy et al. (2019) managed to reach a mean DSC of 0.925 for T_2w images, and a mean DSC of 0.911 for ADC images. Overall, it seems that the trained models were able to reach performance that is very close to what others have attained despite the fact that very little effort was spent on model optimization. Consequently, this means that the performance could probably be pushed a tiny amount further if some time is invested into finding the absolutely best architecture and parameters, but drastic improvements are unlikely.

One noteworthy observation is that the performance of the T_2w model is slightly higher than that of the ADC model. This could be due to at least two factors. First of all, it could be that the prostate glands are simply a bit easier to find from the T_2w images. For example, the contrast between the gland and the background seems to be greater in the T_2w images than in the ADC maps. The second possible reason could be the higher resolution of T_2w images in this dataset. At the achieved accuracy levels, most mistakes made by the model are at the very edges of the

gland, which can be seen in Figure 5.3 if carefully inspected. As the ADC images have lower resolution, the proportion of gland pixels that are on the edge is slightly larger than in the case of T_2w images, meaning that single pixel-level mistakes have a larger impact on the DSC and IoU scores. The smaller resolution also means that the professional who has delineated the glands has also likely made relatively larger mistakes as classification of the pixels located at the outlines of the glands is very unlikely to be exactly precise.

To conclude, the results of the prostate segmentation confirm that the deep learning model architecture based on the U-net is indeed suitable for segmenting the prostate images of the used dataset.

6.2 Prostate cancer lesion segmentation

When compared to the results of prostate gland segmentation, it is obvious that the capability of the models that perform prostate cancer lesion segmentation is much lower. For ADC maps, the best binary segmentation model reached a DSC of 0.342 on the test set, which is relatively close to the results of comparable published models: 0.37 (Schelb et al., 2019), 0.41 $\pm$ 0.28 (Kohl et al., 2017), and 0.46 $\pm$ 0.20 (Dai et al., 2019).

Interestingly, version A of the final model that performed both lesion segmentation and classification managed to improve the DSC to 0.388 for lesions that belong to ISUP grade group 3 or higher. Similarly, version B managed to segment lesions belonging to group 2 or higher with a DSC of 0.398. While these values are slightly higher than in the case of the binary segmentation model, they still fall into the same band as the previously published results.

The results of the multi-class segmentation task indicate that the models had difficulties optimizing their parameters for segmenting all different groups at the same time. It is possible that the version A of the final model had more well-

rounded performance than version B because after a certain point in training the model could no longer perform easy optimizations without making compromises. Increasing model size could have prevented this but training of the model was very slow even at its current size.

Both the versions A and B of the final model reached performance levels, as measured by DSC, that are comparable to the binary segmentation literature for two ISUP grade groups at the same time. However, the DSC values for the other groups were noticeably lower. Because version A excelled at segmenting lesions that belong to at least group 3 or 4, and version B had its highest performance for groups 1 and 2, these two model versions could in theory be combined in order to obtain an improved model that incorporates the strengths of both models.

The pixel-level recall values reveal how greatly the models struggled finding the real prostate cancer lesions. Even for the best-performing groups, only around one third of the true cancer lesion pixels were found. This means that the majority of the lesions would likely go undetected if relying on the model, especially as glancing over the result images indicates that the models either get a large portion of the lesion pixels correct or none at all instead of reliably finding a small portion of pixels in every lesion.

On the other hand, the pixel-level precision values are significantly higher except for ISUP grade 5 lesions. This means that the number of false positive results should not cause enormous problems.

Overall, it is obvious that the current performance of the lesion segmentation models is nowhere near sufficient enough to replace a trained human. While it is possible that the models could assist a professional, the negative impact of all missed lesions would most likely be bigger than the positive impact of marginally faster lesion detection.

6.3 Possible limitations of the study

By examining the magnetic resonance images visually, it is clear that the task of prostate cancer lesion segmentation can be very difficult for untrained humans. In many cases the lesions may show up as only subtle changes in the images, and especially exact detection of the edges of the lesions can be difficult. The task is made more demanding by the fact that not all lesions look similar in the images. As noted by Weinreb et al. (2016), different cancer lesions can cause a subset of several indicative changes in the images, and these effects depend on which zone of the prostate is affected. However, similar changes are also caused by numerous different non-cancerous conditions such as benign prostatic hyperplasia, hemorrhage, and cysts, which means that avoiding false positive predictions can be next to impossible with only image data available.

Since all patients of this study had a suspicion of prostate cancer due to an elevated prostate-specific antigen level, it is possible that some of the patients without prostate cancer had other diseases which might cause difficulties for the machine learning models. It is not known how big of a problem this really is.

Because the used data set was classified by a professional who had access to all available information, it is not known what portion of the lesions would even be possible to detect from the images using only either ADC or T_2w images. For example, according to Weinreb et al. (2016), tumor volume should be estimated based on ADC images if the lesion is located in the peripheral zone, and conversely the estimate should be based on T_2w images if the lesion is in the transition zone. There could possibly even be cases where a lesion shows up in only one imaging mode.

It should also be noted that the Gleason scores and consequently the ISUP grade groups were based on biopsy results, which means that the values may not directly correspond to the information available in the magnetic resonance images. This,

combined with the fact that lesion severity is in reality a continuous metric instead of a discrete one, makes exact classification of lesion groups very difficult. This might have potentially had a large effect on the multi-class segmentation results.

6.4 Future improvements

In order to develop lesion segmentation models with better performance, several steps could be undertaken. First, the amount of data is of paramount importance. Deep learning models generally require large amounts of training data even for tasks that are easy for humans, and the task of lesion segmentation from magnetic resonance images seems more difficult than regular object detection from normal photographs. While the data set used in this study had around 1600 images for the training of ADC models and around 2000 images for the training of T_2w models, these numbers are very low considering that only a small portion of these images included positive cases, and most of those few positive cases were not independent as they usually had intercorrelation with other images from the same patient.

While acquiring images of new patients is time-consuming, creating training set images artificially with data augmentation is relatively quick and easy. The study used fairly simple data augmentation, but more complicated methods could provide samples that better resemble real prostate images. Creating such methods would require considerable amounts of prostate MRI knowledge since it is not clear for a non-expert what kind of differences are possible in the structure of the prostate or in the properties of the magnetic resonance images.

Another possible improvement could be incorporating information about the prostate zones into the model. As noted by Weinreb et al. (2016), the structure of the prostate affects how the cancer lesions appear in the magnetic resonance images. If the model does not receive the information about the structure explicitly, it must deduce the relationships from the data, which requires more time and training

samples.

For the model to be able to generalize to all new cases, it must have seen training examples that contain the necessary information. In this study, the training and test set were split from the data by stratifying based on the ISUP grade groups. The data should probably also be stratified based on the prostate zones of the lesions. For example, according to Weinreb et al. (2016), cancer rarely occurs in the central zone of the prostate. It is important that these cases are also included in the training set because identifying such lesions could otherwise be difficult in the test set.

One of the greatest limitations of this study was the inability to combine information from the ADC and T_2w images into one model. As mentioned by Weinreb et al. (2016), different imaging techniques are more useful than the others in different cases. Because of this, it would be very important to include input images from both methods to the same model. This is not a simple task when the images do not align with each other because of different voxel sizes. There exists so called image registration software which could be used to align the ADC and T_2w images with each other but such applications were not available for this study.

It could be worth investigating another interesting hypothesis: rather than analysing just the structure of the images, it might be more meaningful to compare the structure of the prostate with the corresponding area on the other side of the vertical axis. At least as a non-professional, searching for the lesions from the images is mostly based on comparing the image with its other side which is assumed to be mostly symmetrical. It should be analyzed whether the professionals do this as well, and if this is the case, the deep learning model could be structured in such way that information is incorporated simultaneously from same regions both on the left and the right side of the vertical axis.

7 Conclusions

The aim of this study was to develop machine learning models that could help doctors with the analysis of magnetic resonance images of the prostate by segmenting and classifying prostate cancer lesions automatically, thus saving time and potentially finding cases of prostate cancer that would otherwise go undetected. The achieved results are not yet at a level that could provide meaningful aid in real-world applications. While the models could occasionally find cases that the doctors would not, the number of missed lesions is too high and false positives remain common at the same time. Since the models do not find most lesions, the doctors could more easily believe falsely that there are no lesions present in images.

It was shown that a model with the same structure can reach very good performance for segmentation of entire prostate glands. This suggests that the most important limitation of the prostate cancer segmentation was the lack of data. Lesion segmentation is a much more complex task than prostate gland segmentation both for humans and machines because the physiological location influences how the lesions show up in the images, there are a myriad of possible differences between individual cases, and the changes are more subtle in the images. Deep learning models are notorious for requiring large amounts of independent data in order to generalize well, and thus it is unlikely that a well-performing model could be trained without acquiring a significantly larger dataset. Several possible methods of incorporating domain knowledge into the models in order to improve their performances were dis-

cussed about in the previous section. It is unclear how much these would help but they could improve the performance until a certain point.

Overall, the task of automatically segmenting and classifying prostate cancer lesions at a level that provides meaningful aid for the professionals remains an open problem waiting to be answered.

References

Abadi, M., Agarwal, A., P., B., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., ... Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. https://www.tensorflow.org/

Bengio, Y. (2012). Practical recommendations for gradient-based training of deep architectures.

Bergengren, O., Pekala, K. R., Matsoukas, K., Fainberg, J., Mungovan, S. F., Bratt, O., Bray, F., Brawley, O., Luckenbaugh, A. N., Mucci, L., Morgan, T. M., & Carlsson, S. V. (2023). 2022 update on prostate cancer epidemiology and risk factors—a systematic review. *European Urology*. https://doi.org/https://doi.org/10.1016/j.eururo.2023.04.021

Center, M. M., Jemal, A., Lortet-Tieulent, J., Ward, E., Ferlay, J., Brawley, O., & Bray, F. (2012). International variation in prostate cancer incidence and mortality rates. *European Urology*, *61*, 1079–1092.

Choromanska, A., Henaff, M., Mathieu, M., Arous, G. B., & LeCun, Y. (2014). The loss surfaces of multilayer networks.

Ciresan, D., Meier, U., Masci, J., & Schmidhuber, J. (2011). A committee of neural networks for traffic sign classification. *Proceedings of the International Joint Conference on Neural Networks*, 1918–1921.

Clevert, D.-A., Unterthiner, T., & Hochreiter, S. (2015). Fast and accurate deep network learning by exponential linear units (ELUs).

Cohen, R. J., Shannon, B. A., Phillips, M., Moorin, R. E., Wheeler, T. M., & Garrett, K. L. (2008). Central zone carcinoma of the prostate gland: A distinct tumor type with poor prognostic features. *Journal of Urology*, *179*, 1762–1767.

Dai, Z., Carver, E., Liu, C., Lee, J., Feldman, A., Zong, W., Pantelic, M., Elshaikh, M., & Wen, N. (2019). Segmentation of the prostatic gland and the intraprostatic lesions on multiparametic MRI using Mask-RCNN.

Dozat, T. (2016). Incorporating Nesterov momentum into Adam. *Proceedings of 4th International Conference on Learning Representations, Workshop Track.*

Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, *12*, 2121–2159.

Epstein, J. I., Allsbrook, W. C., Amin, M. B., Egevad, L. L., & the ISUP Grading Committee. (2005). The 2005 international society of urological pathology (ISUP) consensus conference on gleason grading of prostatic carcinoma. *American Journal of Surgical Pathology*, *29*, 1228–1242.

Epstein, J. I., Egevad, L., Amin, M. B., Delahunt, B., Srigley, J. R., Humphrey, P. A., & the Grading Committee. (2016). The 2014 international society of urological pathology (ISUP) consensus conference on gleason grading of prostatic carcinoma definition of grading patterns and proposal for a new grading system. *American Journal of Surgical Pathology*, *40*, 244–252.

Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, *36*, 193–202.

Gillespie, D., Kendrick, C., Boon, I., Boon, C., Rattay, T., & Yap, M. H. (2020). Deep learning in magnetic resonance prostate segmentation: A review and a new perspective.

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 580–587.

Gleason, D. F. (1966). Classification of prostatic carcinomas. *Cancer Chemotherapy Reports*, *50*, 125–128.

Gleason, D. F., & Mellinge, G. T. (1974). Prediction of prognosis for prostatic adenocarcinoma by combined histological grading and clinical staging. *Journal of Urology*, *111*, 58–64.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning* [http://www.deeplearningbook.org]. MIT Press.

Gori, M., & Tesi, A. (1992). On the problem of local minima in backpropagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *14*, 76–86.

Greene, D. R., Fitzpatrick, J. M., & Scardino, P. T. (1995). Anatomy of the prostate and distribution of early prostate cancer. *Seminars in Surgical Oncology*, *11*, 9–22.

Greene, D. R., Wheeler, T. M., Egawa, S., Dunn, J. K., & Scardino, P. T. (1991). A comparison of the morphological features of cancer arising in the transition zone and in the peripheral zone of the prostate. *Journal of Urology*, *146*, 1069–1076.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction*. Springer.

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. *2017 IEEE International Conference on Computer Vision (ICCV)*, 2980–2988.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. *ICCV*, 1026–1034.

Hinton, G. (2012).

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, *2*, 359–366.

Hubel, D. H., & Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *Journal of Physiology*, *160*, 106–154.

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, 448–456.

Jambor, I., Boström, P. J., Taimen, P., Syvänen, K., Kähkönen, E., Kallajoki, M., Perez, I. M., Kauko, T., Matomäki, J., Ettala, O., Merisaari, H., Kiviniemi, A., Dean, P. B., & Aronen, H. J. (2017). Novel biparametric MRI and targeted biopsy improves risk stratification in men with a clinical suspicion of prostate cancer (IMPROD trial). *Journal of Magnetic Resonance Imaging*, *46*, 1089–1095.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning: With applications in R*. Springer.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization.

Kohl, S., Bonekamp, D., Schlemmer, H.-P., Yaqubi, K., Hohenfellner, M., Hadaschik, B., Radtke, J.-P., & Maier-Hein, K. (2017). Adversarial networks for the detection of aggressive prostate cancer.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, *43*, 59–69.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems 25* (pp. 1097–1105). Curran Associates, Inc.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*, 436–444.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. In D. Fleet, T. Pajdla, B. Schiele, & T. Tuytelaars (Eds.), *Computer vision – eccv 2014* (pp. 740–755). Springer International Publishing.

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

Litjens, G., Toth, R., van de Ven, W., Hoeks, C., Kerkstra, S., van Ginneken, B., Vincent, G., Guillard, G., Birbeck, N., Zhang, J., Strand, R., Malmberg, F., Ou, Y., Davatzikos, C., Kirschner, M., Jung, F., Yuan, J., Qiu, W., Gao, Q., ... Maan, B. (2014). Evaluation of prostate segmentation algorithms for MRI: The PROMISE12 challenge. *Medical Image Analysis*, *18*, 359–373.

Maas, A. L., Hannun, A. Y., & Ng, A. Y. (2013). Rectifier nonlinearities improve neural network acoustic models. *ICML Workshop on Deep Learning for Audio, Speech and Language Processing*.

Mariotto, A. B., Yabroff, K. R., Shao, Y. W., Feuer, E. J., & Brown, M. L. (2011). Projections of the cost of cancer care in the United States: 2010-2020. *JNCI: Journal of the National Cancer Institute*, *103*, 117–128.

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the idea immanent in nervous activity. *Bulletin of Mathematical Biophysics*, *5*, 115–133.

McNeal, J. E. (1981). The zonal anatomy of the prostate. *The Prostate*, *2*, 35–49.

McNeal, J. E., Redwine, E. A., Freiha, F. S., & Stamey, T. A. (1988). Zonal distribution of prostatic adenocarcinoma - correlation with histologic pattern and direction of spread. *American Journal of Surgical Pathology*, *12*, 897–906.

Minaee, S., Boykov, Y., Porikli, F., Plaza, A., Kehtarnavaz, N., & Terzopoulos, D. (2020). Image segmentation using deep learning: A survey.

Mitchell, T. M. (1997). *Machine learning*. McGraw-Hill.

Mottet, N., van den Bergh, R. C., Briers, E., Van den Broeck, T., Cumberbatch, M. G., De Santis, M., Fanti, S., Fossati, N., Gandaglia, G., Gillessen, S., Grivas, N., Grummet, J., Henry, A. M., van der Kwast, T. H., Lam, T. B., Lardas, M., Liew, M., Mason, M. D., Moris, L., ... Cornford, P. (2021). EAU-EANM-ESTRO-ESUR-SIOG guidelines on prostate cancer—2020 update. part 1: Screening, diagnosis, and local treatment with curative intent. *European Urology*, *79*, 243–262.

Nair, V., & Hinton, G. (2010). Rectified linear units improve restricted Boltzmann machines. *Proceedings of ICML*, *27*, 807–814.

Nesterov, Y. (1983). A method for solving the convex programming problem with convergence rate $O(1/k^2)$. *Soviet Mathematics Doklady*, *27*, 372–376.

Polyak, B. (1964). Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, *4*, 1–17.

Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*, 1137–1149.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In N. Navab, J. Hornegger, W. M. Wells, & A. F. Frangi (Eds.), *Medical image computing and computer-assisted intervention – miccai 2015* (pp. 234–241). Springer International Publishing.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*, 533–536.

Santurkar, S., Tsipras, D., Ilyas, A., & Mądry, A. (2018). How does batch normalization help optimization? *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 2488–2498.

Schelb, P., Kohl, S., Radtke, J. P., Wiesenfarth, M., Kickingereder, P., Bickelhaupt, S., Kuder, T. A., Stenzinger, A., Hohenfellner, M., Schlemmer, H. P., Maier-Hein, K. H., & Bonekamp, D. (2019). Classification of cancer at prostate MRI: Deep learning versus clinical PI-RADS assessment. *Radiology*, *293*, 607–617.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, *61*, 85–117.

Schröder, F. H., Hugosson, J., Roobol, M. J., Tammela, T. L. J., Ciatto, S., Nelen, V., Kwiatkowski, M., Lujan, M., Lilja, H., Zappa, M., Denis, L. J., Recker, F., Berenguer, A., Määttänen, L., Bangma, C. H., Aus, G., Villers, A., Rebillard, X., van der Kwast, T., ... ERSPC Investigators. (2009). Screening and prostate-cancer mortality in a randomized European study. *The New England journal of medicine*, *360*, 1320–1328.

Simard, P., Steinkraus, D., & Platt, J. (2003). Best practices for convolutional neural networks applied to visual document analysis. *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.*, 958–963.

Sontag, E., & Sussmann, H. (1989). Backpropagation can give rise to spurious local minima even for networks without hidden layers. *Complex Systems*, *3*, 91–106.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*, 1929–1958.

Stanzione, A., Imbriaco, M., Cocozza, S., Fusco, F., Rusconi, G., Nappi, C., Mirone, V., Mangiapia, F., Brunetti, A., Ragozzino, A., & Longo, N. (2016). Biparametric 3T magentic resonance imaging for prostatic cancer detection in a biopsy-naive patient population: A further improvement of PI-RADS v2? *European Journal of Radiology*, *85*, 2269–2274.

Steyn, J. H., & Smith, F. W. (1982). Nuclear magnetic resonance imaging of the prostate. *British Journal of Urology*, *54*, 726–728.

Sung, H., Ferlay, J., Siegel, R. L., Laversanne, M., Soerjomataram, I., Jemal, A., & Bray, F. (2021). Global cancer statistics 2020: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA: A Cancer Journal for Clinicians*, *71*, 209–249.

Tedeschi, E., Palma, G., Canna, A., Cocozza, S., Russo, C., Borrelli, P., Lanzillo, R., Angelini, V., Postiglione, E., Morra, V. B., Salvatore, M., Brunetti, A., & Quarantelli, M. (2016). In vivo dentate nucleus MRI relaxometry correlates with previous administration of gadolinium-based contrast agents. *European Radiology*, *26*, 4577–4584.

Weinreb, J. C., Barentsz, J. O., Choyke, P. L., Cornud, F., Haider, M. A., Macura, K. J., Margolis, D., Schnall, M. D., Shtern, F., Tempany, C. M., Thoeny, H. C., & Verma, S. (2016). PI-RADS prostate imaging - reporting and data system: 2015, version 2. *European Urology*, *69*, 16–40.

Wiederanders, R. E., Stuber, R. V., Mota, C., O'Connell, D., & Haslam, G. J. (1963). Prognostic value of grading prostatic carcinoma. *Journal of Urology*, *89*, 881–888.

Zabihollahy, F., Schieda, N., Krishna Jeyaraj, S., & Ukwatta, E. (2019). Automated segmentation of prostate zonal anatomy on T2-weighted (T2W) and apparent diffusion coefficient (ADC) map MR images using U-Nets. *Medical Physics*, *46*, 3078–3090.