PhD degree in Systems Medicine (curriculum in Computational Biology)

European School of Molecular Medicine (SEMM),

University of Milan and University of Naples “Federico II”

Settore disciplinare: MED/36

Hybrid Deep Learning and Radiomics Models for Assessment of Clinically Relevant Prostate Cancer

Lars Johannes Isaksson

Division of Radiation Oncology, IEO European Institute of Oncology IRCCS, Milan, Italy,

Department of Oncology and Hemato-Oncology, University of Milan, Milan, Italy

Tutor: Prof. Barbara Alicja Jereczek-Fossa

Division of Radiation Oncology, IEO European Institute of Oncology IRCCS, Milan, Italy,

Department of Oncology and Hemato-Oncology, University of Milan, Milan, Italy

PhD Coordinator: Prof. Saverio Minucci

Anno accademico 2021-2022

Thesis advisor: Professor Barbara-Alicja Jereczek-Fossa Lars Johannes Isaksson

Hybrid Deep Learning and Radiomics Models for Assessment of Clinically Relevant Prostate Cancer

Abstract

Precision medicine holds the potential to revolutionize healthcare by providing every patient with personalized treatments and decisions tailored to his or her individual needs. This might be enabled by the large influx of potentially diagnostic information from new sources such as genetics and modern imaging techniques, provided the relevant information can be extracted. One such framework that has started to demonstrate promise in radiology, especially in the assessment of cancer, is radiomics; the practice of characterizing images by extracting a substantial amount of quantitative mathematical descriptors. This success has largely been enabled by artificial intelligence (AI) and machine learning developments that are capable of handling the big data arrays. Using radiomics, researchers have been able to build prediction models capable of assisting and informing doctors in important decisions such as risk assessment or the choice of treatment. But even though radiomics has shown promise in preliminary studies, there is still a long way to go before radiomics and related AI applications can become routine tools in clinics. The road from patient admission to release is long, and all its intricate steps need to be studied in detail to establish the AI models' benefits and safety.

Deep learning is an incredibly powerful AI technique that has revolutionized many areas of science and industry such as recommender systems and protein folding. The technique has demonstrated particular capabilities in image analysis, such as the ability to drive cars autonomously and generate realistic-looking images from scratch. However, the recent advances in deep learning have largely been segregated from the radiomics domain, even though they can synergize with radiomics by performing complementary tasks such as image segmentation and denoising. There is considerable potential for DL and radiomics to cooperatively reinforce each other that so far has been majorly unexplored.

This thesis investigates the application of radiomics and deep learning in the context of prostate cancer. It focuses on the clinical perspective of where machine learning implementations are most likely to have a beneficial real-world impact. A key contribution is the deployment aspect: the models are not simply proofs of concept but are conceived and applied in a practical scenario, from patient admission to treatment decision. The specific areas studied include automatic organ segmentation in medical images, automatic quality assurance of segmentations, image processing, and radiomic feature analysis. Finally, a comprehensive study is performed on predicting essential pathological variables with AI, which so far has not been studied previously. Taken together, the methods outlined in this thesis constitute a concrete pathway of how AI can be used to bolster the steps along the patient's clinical trajectory. Successful applications of these methods hold the potential to reduce the workload of clinicians and improve patient outcomes.

Contents

Acronyms

ADC	Apparent Diffusion Coefficient
AI	Artificial Intelligence
ARVD	Absolute Relative Volume Difference
AUC	Area Under (the receiver operating) Curve
BiFPN	Bidirectional Feature Pyramid Network
CCC	Concordance Correlation Coefficient
CNN	Convolutional Neural Network
Conv	Convolution
CT	Computed Tomography
cv	coefficient of variation
DCE	Dynamic Contrast Enhanced
DL	Deep Learning
DWI	Diffusion-Weighted Images
EPE	Extraprostatic Extension
FLOP	Floating Point Operation
FTT	Feature-Tokenizer Transformer
GAN	Generative Adversarial Network
GBDT	Gradient Boosted Decision Trees
GP	General Practitioner
GT	Ground Truth
HD95	95th percentile Hausdorff Distance
IBSI	Image Biomarker Standardisation Initiative
iPSA	initial Prostate-Specific Antigen
ISUP	International Society of Urological Pathology
LR	Learning Rate
MAE	Mean Absolute Error
MBConv	Mobile inverted Bottleneck Convolution
MCC	Matthews Correlation Coefficient
ML	Machine Learning

MLP	Multilayer Perceptron
mp	multiparametric
MRI	Magnetic Resonance Imaging
MSD	Mean Surface Distance
MSE	Mean Squared Error
MT	Multitask
N	Lymph node status
NDM	Normalized Distance between Means
NMD	Normalized Mean Distance
NMI	Normalized Mean Intensity
OLS	Ordinary Least Squares
PET	Positron Emission Tomography
PI-RADS	Prostate Imaging Reporting & Data System
PReLU	Parametric Rectified Linear Unit
PSA	Prostate Specific Antigen
QA	Quality Assurance
ReLU	Rectified Linear Unit
ROI	Region of Interest
SHAP	SHapley Additive exPlanations
T	Tumor stage
TL	Transfer Learning
TPE	Tree-structured Parzen Estimator

Listing of figures

All contents of this dissertation are the result of my work except where specific reference is made to the work of others. Where stated, the work is a derivative of work published by myself and my collaborators in peer-reviewed journals.

Lars Johannes Isaksson
October, 2022

Acknowledgments

THIS WORK would not have been possible without the help, encouragement, and guidance from my supervisor Barbara-Alicja Jereczek-Fossa and my co-supervisor Paul Summers. I am grateful for the path you've put me on, and that you've let me pursue the most interesting problems and challenges I've encountered along the way. The gratitude extends to all our co-workers and collaborators, who have shown me the value of working together as a team of people from very different backgrounds.

I would like to give a special thanks to Sean Carrol, Ben Goertzel, and Max Tegmark, whom I've never even met, but somehow have managed to show me that being a real scientist is not just about publishing papers and doing experiments. Science communicators and inspirational characters like you are, perhaps irrationally, an enormous part of what has kept me pursuing this path through the years. I do not believe people like you are sufficiently recognized for your endeavors or even acknowledged most of the time. Coming from a purely non-academic background, I would probably not even have known that this path was a viable option was it not for you, and I am eternally grateful for enlightening me.

Last but not least, I would, of course, like to thank my parents for giving me the freedom to roam around throughout my upbringing and later life, both literally and figuratively, sometimes regardless of whether I would end up in a good or a sticky situation. Curiosity has always been a core value for me, and I am grateful you've let me embrace it. Even though we've been far apart at times, we will always be close at heart.

It is now conceivable that our children's children will know the term cancer only as a constellation of stars.

William Jefferson "Bill" Clinton, 2000

1

Introduction

1.1 Prostate cancer

1.1.1 Prostate cancer & precision medicine

Prostate cancer is the second most common cancer (second to breast cancer) and the fifth leading cause of cancer-related deaths in men according to the world health organization[66]. The American National Cancer Institute estimates a total of 268,490 new cases in 2022 in the United States, with a fatality rate of almost 13 % (34,500 deaths)[105]. Traditionally, the detection and diagnosis of prostate cancer have been done through the testing for the blood serum prostate-specific antigen (PSA) biomarker and histology results from ultrasound-guided prostate biopsy. Although biopsies have led to an increased prostate cancer detection rate, the technique still suffers from a high false negative rate since small tumors are unlikely to be detected[44]. New developments in imaging technologies, particularly magnetic resonance imaging (MRI), have made it possible to assess patients more accurately due to the additional information these images provide. In addition to the standard anatomical T2-weighted MRI sequences, multiparametric MRI (mp-MRI) can give complementary physiological insights into metabolic and extracellular activity via dynamic contrast-enhanced (DCE) perfusion and diffusion-weighted imaging (DWI). With this increased staging and risk stratification accuracy, better treatment decisions can be made which can lead to a significant decrease in over- and undertreatment of patients[74,88,99,108].

The influx of new and better diagnostic information leads to an increased diversity between patients in terms of their data profiles. A potential future can be envisioned where the treatment decisions are not based on staging, but on the complex network of patient data—this is the promising potential of personalized/precision medicine. In personalized medicine, all of your relevant data is considered when making a decision so as to tailor the outcome perfectly to what your body is most likely to respond to positively. Of course, this will require a great deal of knowledge about the subject matter and comprehension of the large body of data, which is why artificial intelligence (AI) is seen as a great tool to aid in this endeavor. In contrast to humans, AI systems are known to handle vast arrays of data with ease. In fact, AI is already underway to assist doctors in real clinical situations, for in-

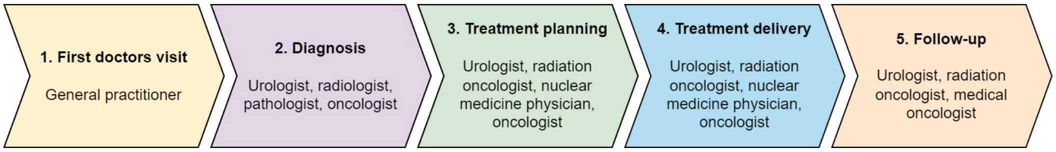

1. First doctors visit
General practitioner
2. Diagnosis
Urologist, radiologist, pathologist, oncologist
3. Treatment planning
Urologist, radiation oncologist, nuclear medicine physician, oncologist
4. Treatment delivery
Urologist, radiation oncologist, nuclear medicine physician, oncologist
5. Follow-up
Urologist, radiation oncologist, medical oncologist

Figure 1.1: The different steps in the prostate cancer pathway that a typical patient goes through and what type of practitioners are involved.

stance as decision support systems and image analysis tools. But there is still a large amount of work to be done and questions to be answered before AI can be reliably immersed into clinical practice.

1.1.2 The patient pathway

Before digging into the weeds of artificial intelligence (AI) and machine learning, it will be useful to conceptualize how the current practices look from the patient's perspective. How does one detect prostate cancer? What happens after treatment? Etc. This section will cover just that. An overview of the different steps a typical prostate cancer patient goes through, from admission to release, can be seen in Figure 1.1. This will be a useful frame of reference for discussing the applications of AI that will ensue in the following section.

A patient's first visit is typically with a general practitioner (GP). If prostate cancer symptoms like urination difficulties, weak urine stream, or blood in the urine are ascertained, the GP may initiate a blood test to measure PSA and a digital rectal exam to check for enlargement of the prostate. If there is sufficient reason to suspect prostate cancer, the patient is referred for diagnosis.

At the diagnosis stage, the patient goes through prostate biopsy, from which the pathologist infers the ISUP (International Society of Urological Pathology) grade group, and if suspect, prostate MRI imaging, which leads to the PI-RADS (Prostate Imaging Reporting & Data System) score from a radiologist. The improvement of prostate MRI over the past decade has seen an inversion of this order, with the radiological examination now recommended first, and biopsy being limited to those cases where there is sufficient suspicion of tumor, ideally using the MRI data to guide needle placement in the biopsy procedure. These staging evaluations, along with the related information such as the age, prostate volume, and PSA level, go into the radiology and pathology report, which is then used to diagnose the patient and ultimately guide the treatment decision, sometimes in a multidisciplinary consultation with all involved clinicians.

If the cancer is in early development and the risk related to the patient is thought to be low, active surveillance (e.g. no treatment, but regular re-staging visits) is a common option. When treatment is warranted for more advanced disease, the options include hormone therapy, chemotherapy, prostatectomy, and radiotherapy, the most common of which being external beam radiation therapy. After the treatment has been decided, the delivery plan and tracking schedule need to be set. A common practice for radiotherapy is to divide the total dose delivery over a number of sessions, usually over multiple weeks, in what is called fractionated radiotherapy. A key component in this path is to decide the dose distribution, both in terms of its spatial location and temporal spacing.

After the treatment has finished, a follow-up visit is scheduled where the clinicians assess how successful the treatment was, if the cancer is in remission, and the future prognosis of the patient. In case of disease progression, further treatment is proposed.

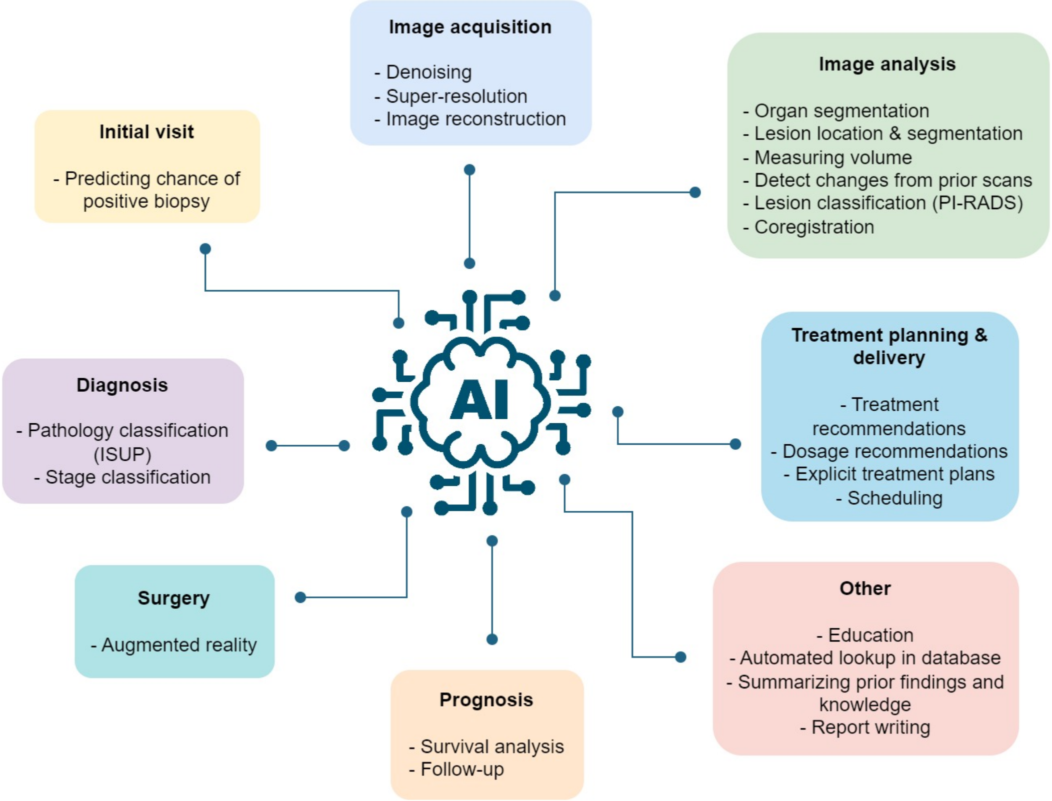


Image acquisition
- Denoising
- Super-resolution
- Image reconstruction
Image analysis
- Organ segmentation
- Lesion location & segmentation
- Measuring volume
- Detect changes from prior scans
- Lesion classification (PI-RADS)
- Coregistration
Initial visit
- Predicting chance of positive biopsy
AI
Diagnosis
- Pathology classification (ISUP)
- Stage classification
Treatment planning & delivery
- Treatment recommendations
- Dosage recommendations
- Explicit treatment plans
- Scheduling
Surgery
- Augmented reality
Other
- Education
- Automated lookup in database
- Summarizing prior findings and knowledge
- Report writing
Prognosis
- Survival analysis
- Follow-up

Figure 1.2: An overview of some of the potential applications of AI in clinics.

1.1.3 The promising potential of AI in prostate cancer

An illustrative overview of some of the ways AI can be integrated into and help with the clinical workflow described in the previous section can be seen in Figure 1.2. Not all of the applications mentioned in this section are equally relevant in real-world scenarios and some of them are further away than others, but it is clear that the number of possible AI applications in the prostate cancer journey is enormous.

At the first meeting with the GP or urologist, the interaction is fairly basic, but AI could still help with deciding whether a biopsy is relevant. In this context, an AI could assess all of the symptoms the patient is experiencing and combine them with the digital rectal exam and PSA level to come up with a recommendation. It could even cross-reference the symptoms and blood levels with other conditions to see if there are other potential pathologies worth examining further.

If prostate cancer is suspected and the patient goes on to imaging, AI can be used to improve the images in various ways. Improving the image quality by increasing the resolution and decreasing the noise are two straightforward examples. Depending on the imaging modality, there may also be technical applications in the acquisition process, such as the reconstruction algorithm for MRI images (the raw data in MRI is in Fourier space, and must therefore be transformed into an understandable 2D image).

There are a lot of applications for AI in the image analysis step since this is where most of the clinical assessments are grounded. AI can locate and segment organs and lesions and thus also accurately measure their volumes and potentially perform classification (e.g. give the PI-RADS score that is later used to determine the patient's overall status). Organ segmentation is also essential to estimate the damages to surrounding tissues if radiotherapy is prescribed. AI can coregister the different images acquired as part of the multiparametric PI-RADS compliant MRI examination such that they align perfectly, which simplifies their comparison and prevents the need to segment the organs in multiple images. If the patient has been scanned previously on earlier visits, AI can help with accessing these and compare them to the new scans, e.g. by highlighting important differences or providing high-level conclusions.

In the diagnosis stage, AI can effortlessly integrate all the collected information and provide estimates of the interesting endpoints such as the ISUP grade group (directly from the pathology images), tumor stage, or risk class. If the AI is powerful enough to predict the pathology results from the other data accurately, a biopsy could even be prevented, which would prevent potential complications from the surgery.

The first and most straightforward role of AI in the treatment stage is of course to assist in the treatment decision by recommending different treatments and ideally providing pros and cons of the different options. The treatment delivery can also be assisted by AI, for instance by providing treatment plans and dose recommendations. Another application is to estimate the potential side effects and other complications such as damage to surrounding organs in the case of radiation therapy.

If prostatectomy is the preferred treatment, AI could help in the operation e.g. through augmented reality. In such a case, the AI could display MRI images in the surgeon's peripheral vision to provide information on critical structures like the location of lymph nodes. It could even help in training an inexperienced surgeon by providing visual cues to guide the incisions.

After treatment, or if active surveillance is determined to be the best course of action, it is useful to schedule a new appointment to establish if the condition has improved and if additional treatment should be pursued. AI can of course assist with this task as well. It is also useful to provide a survival analysis to estimate the life span of the patient, which is already a vital part of the treatment decision. If the patient is very old, for example, long-term effects are unlikely to be a concern since the patient is more likely to die of other causes, which can influence the decision.

There is a long list of other potential uses of AI: educating new personnel, automatic database lookup & organization, summarizing prior patient data or knowledge about the condition, writing reports, communicating with and consolidating patients and relatives, triage, etc. The examples provided here are just some of the most pertinent ones.

1.2 Mathematical modeling & machine learning in healthcare

1.2.1 What is AI?

AI is the general term adopted to describe essentially everything computer-related that has to do with things we normally associate with intelligence: performing a complex task, playing a game, drawing conclusions, etc. Machine learning (ML) is a subset of AI that refers to the branch of artificial intelligence that concerns the study of giving computers the ability to learn rather than being explicitly programmed. The term "learning" should however be considered colloquially since the procedure is very different from how we imagine human agents learn. Notably, an ML model's parameters are fixed after they have been learned, which means that a deployed model is unable to learn from its mistakes unless it is explicitly updated. Deep learning (DL) is, in turn, a subset of ML that refers to a specific strategy for building ML models, namely with so-called artificial neural networks (to be discussed below). As the name suggests, these models were inspired by the neurons in the brain; the idea being to connect simple artificial neurons together to make a more powerful network, much like the brain does. The phrase "deep" simply refers to networks with many layers of artificial neurons in them.

This section will briefly introduce some of the basic concepts within the ML field and a few of the more advanced notions that will be recurrent throughout the discussions in this thesis. In particular, it covers model training and model validation, what constitutes a learning algorithm, overfitting & underfitting, and regularization.

1.2.2 Building a mathematical prediction model

There are many different ML algorithms and models, but let us start with a simple illustrative example. The task is to build a model that predicts the outcome variable $\mathbf{y}$ as close as possible given some input parameters $\mathbf{X}$. Consider $\mathbf{y}$ as a vector of numbers that represents some characteristic (e.g. whether a patient has a disease or not) in a set of samples (patients) and $\mathbf{X}$ as a matrix storing the available data for the samples. For a model M, we can denote the predicted vector as $\hat{\mathbf{y}}$ and thus $M(\mathbf{X}) = \hat{\mathbf{y}}$. In linear regression, the outcome is modeled as a linear combination of the input variables: $\hat{\mathbf{y}} = \mathbf{Xb}$. The "learning" here amounts to finding the parameters $\mathbf{b}$ that minimize the (squared) difference between the predicted $\hat{\mathbf{y}}$ and the real $\mathbf{y}$, or in other words:

$$\arg\min_{\mathbf{b}} ||\mathbf{y} - \mathbf{Xb}||_2^2 \tag{1.1}$$

Recalling that the derivative of a function is zero at the point of its minimum (or maximum) allows us to find these parameters (the $\mathbf{b}$-values) by differentiating the above expression with respect to $\mathbf{b}$ and equating the resulting expression with 0. This results in $(\mathbf{y} - \mathbf{Xb})^\top \mathbf{X} = 0$, or equivalently, $\mathbf{b} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$[55]. This is called the ordinary least squares (OLS) estimator, and the parameters can generally be acquired easily because there is a simple unique solution. However, in most real-world cases, there is no *exact* solution ($\hat{\mathbf{y}} \neq \mathbf{y}$) because the outcomes are not perfectly predictable from the variables.

When the parameters $\mathbf{b}$ have been found, the model is said to have been learned. In general, learning simply constitutes finding appropriate internal parameters. For more complex models, e.g. ones that can model non-linear relationships between variables, it is generally not possible to set up a simple analytical expression like (1.1). In such cases, one must resort to other learning algorithms, the most popular being the gradient descent algorithm, which iteratively finds better and better parameters.

The difference $\mathbf{y} - \hat{\mathbf{y}}$ measures how far our model is from the real outputs, and since we obtain the $\hat{\mathbf{y}}$ estimates from the samples the model was trained on, this is called the training error. As an estimate of how good the model is in practice, however, the training error is not a good measure, since it is biased towards performing well on that particular set of data (which may or may not be representative of the general population). To estimate how good the model is in reality, we need to collect new samples and measure its error on them. This is called the test error—it is a measure of how good the model is at generalizing to unseen data.

Accurately measuring a model's generalization performance can be tricky because it requires new data that the model has not seen before. To circumvent this, the data is usually split into one training set, on which the model is trained, and one test set, on which the model is tested. However, this raises a major problem because we would like the model to use as much training data as possible in order to achieve high performance, and we would like the test data to be as large as possible in order for it to accurately represent the true data distribution. These problems are exacerbated when there is a limited amount of data available, which is a particularly pressing concern in medical research. How, then, does one properly split the data into training and test sets? The standard procedure is cross-validation, where the data is repeatedly split into different train/test splits and the model is retrained and reevaluated on each split. An especially popular cross-validation strategy is K-fold cross-validation, where the data is divided into K equally sized folds, and only one fold is the test set at a time. Five-fold cross-validation thus uses one-fifth of the data for testing and four-fifths for training, then retrains the model five times on the five different (but overlapping) training sets. An obvious downside of this is that K models have to be trained, which entails additional computational costs and time investments. Another thing to note is that the K parameter strongly influences the learned parameters and the produced error estimates. An optimal K cannot be known, but both five- and ten-fold cross-validation are commonly accepted.

What happens if the test error is much larger than the training error? In this case, the model performs poorly at predicting the outcome of unseen samples, which probably means that it will be useless in practice. This often happens when the training set is very small or when too much noisy or irrelevant data is used in building the predictions. The terms used to collectively describe these scenarios are *overfitting* and *underfitting*, respectively. More generally, underfitting is when a model fails to adequately capture the structure of the data, and overfitting is when a model fits a particular set of data so closely that it fails to capture a general representation of the underlying data mechanisms (see Figure 1.3). Striking a good balance between overfitting and underfitting is arguably the single most important problem in machine learning.

To combat overfitting, researchers and developers make use of regularization techniques, which is a collective term for strategies that reduce the complexity of a proposed solution or generally decrease its generalization error. One of the motivations behind reducing the complexity lies in the fact that more predictor variables and higher order polynomials have a stronger ability to "curve around" individual data points as demonstrated in Figure 1.3. Two common regularization techniques are L1 and L2 regularization, in which a penalty

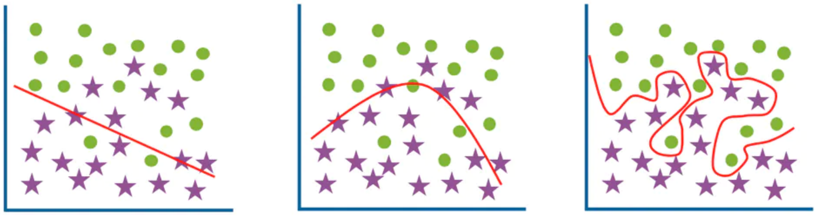


Figure 1.3: A simple illustration of the overfitting and underfitting problem on a classification task where the objective is to separate green dots from purple stars. An underfit model (left) is so simple that it fails to capture the interesting behavior in the data. An overfit model (right) has very low training error, but will likely produce a high error on unseen data because it follows the spurious and random behavior in the data too closely. A good fit (middle) captures the true data distribution fairly well. Image modified from IBM cloud (accessed Aug 2022).

term is added to the objective function in (1.1) that penalizes large values in the **b** parameters: $+\lambda|\mathbf{b}|$ for L1 or $+\lambda\mathbf{b}^2$ for L2 (λ is a hyperparameter). This leads the model to not rely on individual variables too much (the penalty is higher for higher values of b_i) and at the same time not include variables with too small a contribution (b_i will approach zero when $|y_i - X_i b_i| < \lambda b_i$).

Various forms of randomness in the training procedure can also have a regularizing effect. In artificial neural networks, an exceptionally important regularization technique is called dropout. When using this technique, a random subset of the nodes in the network is disregarded at the beginning of each new training step, which makes the network rely less on outputs from individual neurons. There are, of course, many other regularization techniques as well, and multiple regularization techniques can even be used together.

1.3 Deep learning

1.3.1 Basic principles of deep learning

Let us first conceptualize the basic principles behind deep learning (DL). At a very basic level, a DL model can be thought of as a system that takes in a particular input and produces an output of desired shape and form, such as a simple scalar or a vector. For example, if we want to build a model that can estimate a person's age by looking at a picture, we would set up the model such that it takes the image as input, and produces a single number as output. The key principle is that if we know the true output for a given input (e.g. the real age of a person in a picture), and let the model produce its own output, we can calculate how far apart the model's estimate is from the real value. If we then let the model update its internal parameters by taking a small step in the direction of the true output, the error of the model will be smaller. Repeating this process many times for many different inputs eventually results in a model that produces outputs that are very close to the desired outputs. This iterative process of minor improvement lies at the heart of supervised DL modeling.

The idealized training procedure mentioned above requires a few properties and features to work properly:

1. A model **architecture** (the "black box"): the architecture should be such that the output of the model will be of the desired type, which is ensured by building the model (especially the output layer) in a particular way. Notice that the above description of the learning procedure can be realized in many different models and not just DL models—the criterion for being a DL model is the particular building blocks

that are used, namely artificial neurons. After building a model and defining its internal parameters, the mathematical operations that the model performs are in fact precisely determined, which means that we know exactly how the model computes its output. The term “black box” comes from the fact that these mathematical operations are often too abstract to interpret in a semantically meaningful way, which adds a layer of unintelligibility.

2. A **loss function** is what makes it possible to calculate the distance between the model’s output and the real output. The choice of loss function depends heavily on the problem at hand, and there are often standard loss functions to use for common problems (although, you are free to use or create your own). For example, when the problem is to predict a real number, the mean squared error is a natural choice. The only constraint on the loss function is that it has to be differentiable because this is what makes it possible to know the direction in which to take the step (that is, to update the model’s parameters).

3. **Labelled data**: if we don’t know the true value of the output for the data we train our model on, we cannot compute the loss function. This type of training, where the true labels are known a priori, is called supervised learning because it’s akin to a supervisor telling the model what the right and wrong answers are.

4. A **training algorithm** that updates the model’s parameters, coupled with an optimizer that decides how big a step it should take. The standard training algorithm in supervised learning is called backpropagation (or simply backprop). This algorithm is so called because it starts by computing the gradients with respect to the error for

the final layer, then iteratively propagates the error backward to compute the gradients for earlier layers. The optimizer is different and there are many to choose from. In practice, however, it is very common to start with the Adam optimizer because it has historically performed well on a wide variety of tasks and also possesses attractive properties in terms of its speed and memory consumption.

The above description portends what may be called the central dogma of deep learning: large DL models are data-hungry and require large amounts of data to work well. If only a few samples are given, the step variety will be low, which will limit the model's ability to cover the parameter space. The fundamental assumption of the training procedure is that the model will generalize to unseen data just from learning to minimize the error in the training set. It turns out that this is often the case in practice, so long as the new data comes from a similar distribution. Exactly how much data is needed for a model to work well is impossible to say a priori because it is heavily dependent on both the problem and the quality of the data.

1.3.2 Advanced principles of deep learning

With the conceptual principles out of the way, we can dig deeper into how the DL components work and how models are constructed. The fundament of the original artificial neural network is the perceptron (Figure 1.4)[119]. Loosely inspired by the biological neuron, the perceptron takes in a set of inputs (analogous to the dendrites) and "decides" whether or not it should produce an output based on the cumulative input signal (analogous to the action potential). This output signal may then be the input to one or more other perceptrons (analogous to the axonal connections), and so on. Mathematically, the output o_j for

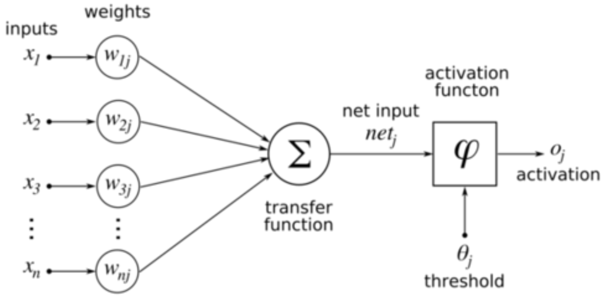


Figure 1.4: The perceptron—the basic building block of the standard artificial neural networks, proposed by Rosenblatt in 1958[119]. A set of inputs $x_1, \ldots, x_n$ are multiplied by their respective weights $w_{1j}, \ldots, w_{nj}$ and then summed into a net input net_j. The output, o_j, of the perceptron is calculated by passing the net input net_j together with the bias term θ_j through the activation function φ. The output is also described in (1.2).

neuron *j* can be formalized by

$$o_j = \varphi(\theta_j + \sum_i x_i w_{ij}) \tag{1.2}$$

where *x* are the different inputs, w_{ij} is the weight associated with output *i*, and θ_j is the perceptron's so-called bias term. The weights and biases are the primary parameters that are learned in the training procedure. Note that we can choose whichever activation function we want and are not limited to the (mostly) binary activation patterns of real neurons, but in practice, simple activation functions are preferred because the derivative of the function needs to be computed during training, which can be computationally expensive for advanced functions.

The simplest architecture one can build in this way is the multilayer perceptron (MLP)[120], where a set of layers, each containing a predefined number of perceptrons, are stacked on top of one another. Each node in a layer is typically connected to every node in the previous layer, which is to say that the network is densely connected. The weights determine the strengths a node assigns to different input signals, which means that it can effectively learn to ignore some of its inputs by setting the corresponding weights to zero. The final layer, which computes the ultimate output of the model, varies in size and shape depending on the problem: if the problem is regression, the output is a single node and if the problem is multiclass classification, the output layer has one node for each class (each representing the probability for the corresponding class), and so on. A consequence of the denseness property of the MLP, which is inherited by virtually all modern DL architectures (save for sparse networks), is that the number of parameters grows polynomially with the number of nodes and layers. This means that most DL models are vastly overparameterized; it is not uncommon to see networks with hundreds of billions of parameters in the current literature.

The objective of the training algorithm is to update the weights and biases in such a way that the error of the next iteration will be less. If we know the gradients of the loss function with respect to the weights, this can be achieved by taking a small step in the negative gradient direction (since the gradient of a function indicates the direction in which the function increases). A step in this context simply means updating the parameter's value. There are two problems here: 1) the gradients are not directly accessible—only the gradient with respect to the output can be directly calculated (assuming the loss function is differentiable), and 2) the layered structure of the network means that only the last layer's derivative is directly dependent on the loss—the other layers' gradients are dependent on the values of their superseding layers. Both of these problems can be solved by applying the chain rule: instead of calculating $\frac{\partial L}{\partial \mathbf{w}}$, we can rewrite it as $\frac{\partial L}{\partial \mathbf{w}} = \frac{\partial L}{\partial o_j}\frac{\partial o_j}{\partial \mathbf{w}}$ for the output neuron j and its weights $\mathbf{w}$ (the bias term can be considered as a part of $\mathbf{w}$ whose input is always 1). But from (1.2) and Figure 1.4, we can see that $\frac{\partial o_j}{\partial \mathbf{w}} = \frac{\partial \varphi(\sum x_i w_{ij})}{\partial \mathbf{x}}\frac{\partial \mathbf{x}}{\partial \mathbf{w}} = \frac{\partial \varphi(\sum x_i w_{ij})}{\partial \mathbf{x}}\mathbf{w}$ and hence

$$\frac{\partial L}{\partial \mathbf{w}} = \frac{\partial L}{\partial o_j}\frac{\partial \varphi}{\partial \mathbf{x}_j}\mathbf{w}_j. \tag{1.3}$$

Since all the factors above can be calculated from the input $\mathbf{x}_j$ and output o_j of the j'th layer, we can calculate $\frac{\partial L}{\partial \mathbf{w}}$ and hence know the direction we should update the weights in. Finally, by noting that every element in $\mathbf{x}_j = \mathbf{o}_{j-1}$ is the output of a previous neuron (provided layer j is not the first layer), we can calculate this quantity for every weight in the network. This is what constitutes the backpropagation algorithm. When training a network in practice, one multiplies $\frac{\partial L}{\partial \mathbf{w}}$ with a step size parameter called the learning rate, which controls how fast/stringent the learning should be.

1.3.3 Computer vision with deep learning

When images are to be analyzed with computers, it is convenient to think of them as arrays of numbers; a 256-by-256-pixel image is simply a matrix with 256 rows and 256 columns. If there are colors in the image, each different color is represented in a separate channel—standard convention uses just three channels, since most colors can be mixed with different proportions of red, green, and blue—so a normal color image would be a simple 3D matrix. Analyzing matrices like this is perfectly feasible with MLPs, but a few obvious issues arise:

1. In a standard fully connected MLP layer, each of its nodes/neurons is connected to every input, meaning that a single neuron in the example above will have $256 \times 256 = 64,536$ connections for a relatively small 256-by-256 greyscale image. It is easy to see that the number of connections grows rapidly for larger images and wider layers.

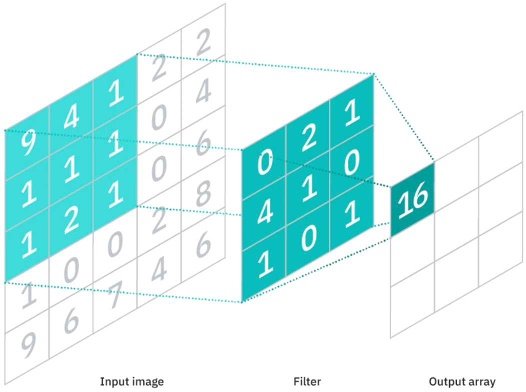


Figure 1.5: A functional illustration of a 3×3 convolutional filter. Each element in the filter is multiplied by the corresponding element in the local patch in the input image. The final intensity value of the output image is calculated by adding all the resulting elements together; in this case $2 \cdot 4 + 1 \cdot 1 + 4 \cdot 1 + 1 \cdot 1 + 1 \cdot 1 + 1 \cdot 1 = 16$. The next pixel value in the output image is calculated by sliding the filter one step to the right and performing the same multiplication on the new local patch. The values in the filter are learned through training.*

2. The connections are static, so a specific weight will always be connected to a specific location in the image. This means that the activation patterns will depend on where in the image objects are located, which is often not a desired property. If we build a network to detect cats, we want the detection to work regardless of where in the image the cat is located.

3. There is often a tremendous number of local correlations in images. Most of the pixels in a black cat tend to be black, most of the pixels in a blue sky tend to be blue, etc. Fully connected layers disregard these correlations completely since neighboring weights don't communicate with one another.

Convolutional neural networks circumvent these problems by using convolution filters that slide across the image and calculate output values by applying the convolution to local patches (see Figure 1.5). Notably, the output is also a matrix, which makes sequential application of convolutional layers straightforward. As an example of what this can look like,

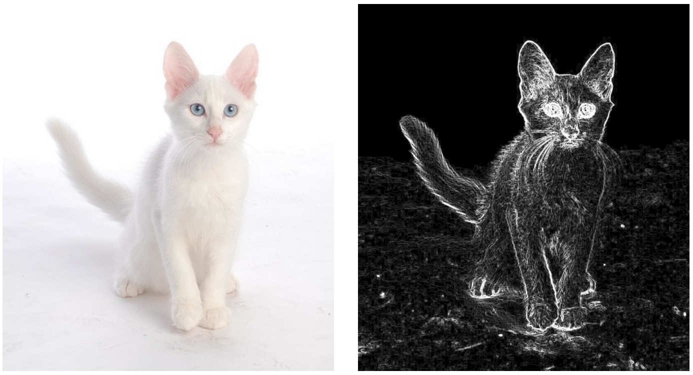


Figure 1.6: When applied to an image (left), the Schaar convolutional filter produces an image where the edges are highlighted (right). This filter has a clear intuitive interpretation (edge detection), but a general convolution is typically more abstract.

consider the Schaar filter shown in Figure 1.6. The output of this filter is high when there are large variations in the local patch, which effectively makes it detect edges in the image. In CNNs, the filter parameters are learned through gradient descent and backpropagation which enables the network to automatically extract useful information for the task at hand. The beauty of this approach lies in the fact that sequential filter applications act as a successive abstraction mechanism, where early layers extract low-level abstractions (edges, shapes, textures, etc.) and deeper layers extract high-level abstractions (objects, animals, emotions, etc.). Thus, a well-built and properly trained CNN can learn to do everything from object detection to sentiment analysis (e.g. detect if a person looks sad or happy).

1.4 Radiomics

1.4.1 The hope and history of radiomics

The discipline of radiomics has recently emerged as an attempt to improve the quality and reliability of the current clinical workflow, particularly in radiation oncology where multiparametric MRI images are nowadays routinely acquired. The fundamental hypothesis in radiomics is that useful information that is not readily apparent to human observers can be mined directly from medical images. This information can then be used to build predictive ML models and inform doctors. Information is extracted by calculating many predefined quantitative features from the image matrix, such as the entropy, energy, or standard deviation of the pixel intensities, and even more features can be extracted by calculating features on various image derivatives, such as co-occurrence and size zone matrices, 3D representations of organs, or filtered versions of the original image, leading to the collection of large amounts of data. The initial successes of radiomics in retrospective studies have established the field as one of the most promising approaches to quantitative imaging in radiology and oncology.

The purpose of the radiomic features is to extract useful and ideally complementary information from the images with the potential to provide diagnostic power. For example, it is conceivable that some features could reflect underlying characteristics of the physiology of the tumor such as genomic and proteomic expression patterns. This hope is partially supported by evidence: it has been demonstrated that MRI findings can be associated with protein expression in brain glioblastoma[59], that CT-imaging traits can be correlated with histopathologic markers and gene expression in liver cancer[85,121], and that MRI phenotype of breast cancer can be associated with multiple genetic mechanisms[161]. If true, clinicians could gather proxies of these types of biomarkers without the need for conducting complex genomics or proteomics assays. It is also likely that radiomic features could consolidate the radiologists' assessments since the images they are extracted from already constitute a significant part of the clinical evaluation—it is for example well known that morphological tumor characteristics and intensities in diffusion-imaging are highly relevant to the clinical prognosis.

Although radiomics started to emerge as a tool for clinical radiology in 2012[87,84], the usage of quantitative features to analyze images is far from new. Similar methods have long been used for industrial applications in agriculture, mining, industrial inspection, security, image retrieval, aerial photography, and satellite image analysis[115]. In fact, texture-based mathematical image analysis date back all the way to the 1970s[53,41]. But the data-driven nature of radiomics offers no inherent insight into the biological underpinnings of the observed relationships between image characteristics and disease status[38], and, as such, the models and relationships must be very thoroughly validated before they can be formally established. Thus, there is a long way to go before radiomic models can be reliably deployed in real-world clinical applications.

1.4.2 A look at the radiomic features

The standard radiomic features can be divided into different groups based on what they attempt to describe: first-order statistics, shape-based (2D and 3D), and different gray-level matrix categories. In this section, we describe the categories further and give a few examples of their features. It is worth noting that different extraction programs and packages may support different features and their implementations may differ even for the exact same feature. To combat this, a large community-wide effort called the Image Biomarker Standardization Initiative (IBSI) was started in 2020 with the goal of standardizing the definition and extraction of the features and related procedures[162]. It is fair to say that every serious radiomics extraction software today strives to comply with the IBSI standard.

The first and arguably most intuitive group of features is the first-order statistics features, which encompass statistical properties of the intensities in the image such as the minimum, mean, standard deviation, and interquartile range. Some other examples include (letting $\mathbf{X}$ be the set of N voxels, $\mathbf{P}$ the first order histogram, N_b the number of non-empty bins in $\mathbf{P}$, $\mathbf{p} = \mathbf{P}/N$, and ε an arbitrary small value):

$$energy = \sum_{i}^{N} x_i^2 \tag{1.4}$$

$$entropy = -\sum_{i}^{N_b} p_i log(p_i + \varepsilon) \tag{1.5}$$

$$mean\ absolute\ deviation = \frac{1}{N}\sum_{i}^{N} |x_i - \bar{\mathbf{X}}| \tag{1.6}$$

Generally speaking, these do not have any simple interpretations over and above the statistical ones, but some may be straightforwardly related to important features in the image. For instance, high-intensity values in a diffusion image are associated with higher metabolic activity, which will be indirectly reflected by a high mean intensity within the tumor.

The shape features describe properties of the morphology of the region of interest and are therefore independent of the intensity values in the image. Some simple examples include the volume V, surface area A, and maximum diameter. Other examples include:

$$sphericity = \frac{\sqrt[3]{36\pi V^2}}{A} \tag{1.7}$$

$$compactness = \frac{V}{\sqrt{\pi A^3}} \tag{1.8}$$

$$flatness = \sqrt{\frac{\lambda_1}{\lambda_n}} \tag{1.9}$$

where λ_1 and λ_n are the first and last principal components of the shape of the region, respectively. The first principal component is simply the vector from the center of the region to its furthest point, and the last component is the shortest radial vector orthogonal to it. As such, flatness measures how flat a region is by taking the ratio between these two; a flat object (e.g. a single-slice region) will have a large flatness, and a round object will have a flatness close to one[*] [143]. Likewise, sphericity measures the region's resemblance to a sphere, and compactness can be interpreted as a measure of how compact a region is.

Gray-level matrix features are features calculated from different matrices constructed from the gray levels of the histogram (a gray level in this context is a non-empty bin in the histogram). The matrices differ in the way that their values are determined. Most of the time, these can be interpreted as quantifying various characteristics of the texture in the images[6], although exactly *how* they relate to semantically meaningful concepts is not generally known. This thesis will not delve deeper into their interpretations or how these matrices differ in practice, but additional information can be found in the IBSI manual or in the documentation of various radiomics software packages (e.g. PyRadiomics[143]). Five different matrices are commonly constructed:

- Gray Level Co-occurrence Matrix (GLCM). The values are determined by the number of times different gray levels occur together in close vicinity.
- Gray Level Run Length Matrix (GLRLM). The values are determined by the number of times connected regions of a specific grey level and length occur.

[*] To prevent division by 0, the inverse flatness is often computed instead, which ranges between 1 and 0.

- The Gray Level Size Zone Matrix (GLSZM) counts the number of zones in which all spatially connected voxels share a common grey level. The $(i,j)^{\text{th}}$ element in the matrix equals the number of times a zone with gray level i and size j appears in the image.

- Neighbouring Gray Tone Difference Matrix (NGTDM). Describes the average difference between pixels of a specific grey level and their mean neighborhoods.

- Gray Level Dependence Matrix (GLDM). The values are determined by the number of voxels within a certain vicinity that are dependent on the center voxel of the vicinity (dependence is defined as being within a certain predefined threshold value).

All features above are routinely extracted from the original images, but it is conceivable that features calculated from processed images (e.g. with the Schaar filter shown in Figure 1.6) could also reflect diagnostic information. Software packages thus provide the option to calculate features from various image derivatives. Common options include:

- Wavelet images. For 3D images like MRI and CT, these filters can be applied independently in the three directions, which means that there are eight different combinations of high- (H) and low- (L) pass filters: HHH, HHL, HLL, etc.

- Laplacian of Gaussian. This is effectively an edge detection filter similar to that of Figure 1.6. The image is acquired by first applying the Gaussian filter and then the Laplacian filter to the same image.

- Square. All intensity values are squared.

- Root. All intensity values are passed through the square root function.

- Logarithm. All intensity values are passed through the natural logarithm function.
- Exponential. All intensity values are passed through the exponential function (e^x).
- Gradient. Intensity values are replaced with the local gradients in the corresponding pixel positions.
- Local binary pattern (in 2D and 3D). Intensity values are replaced by the local binary pattern as calculated by different levels of spherical harmonics.

1.4.3 Radiomics in the clinical domain

The typical radiomic workflow is consists of a series of steps that can be roughly divided into three main groups: image preparation, radiomic feature extraction, and analysis (see Figure 1.7). This section will go through the steps in detail and cover some of the core considerations that go into them.

The first step after acquiring the images is to delineate the tumor such that the radiomic features can be calculated within the pathological tissue. This may be done routinely in medical practice in order to e.g. arrive at reasonable dose distributions for the treatment, which means that there is not necessarily any additional clinical segmentation workload for conducting retrospective radiomic trials. If not, a medical expert is recruited to segment the tissue which is usually done with a computer mouse in some proprietary software.

The last step in the image preparation group is to process the images such that quantitative analysis can be carried out robustly. This differs depending on the imaging modality and what part of the body is being examined. For instance, MRI images can have very different intensity value ranges even for the same scanner and protocol, which means that

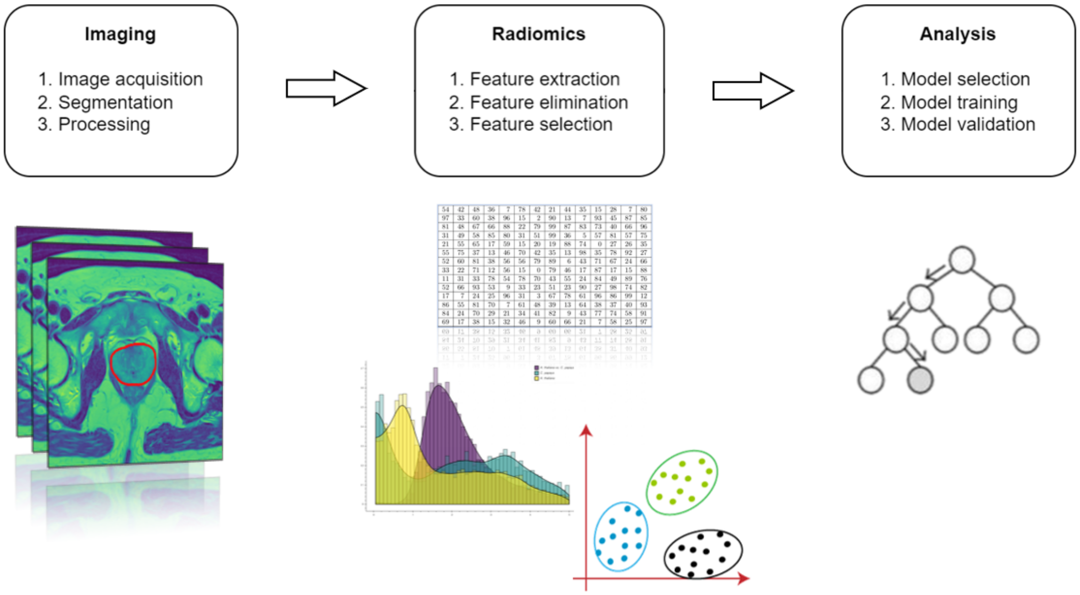


Figure 1.7: The workflow of radiomics starts with the medical images. After the images have been acquired, segmentation is performed such that the radiomic features can later be extracted from the tissue of interest. It can sometimes be desirable (particularly in MRI) to process the images such that the intensity values are consistent within the population. After the features have been extracted, the feature set is pruned, e.g. by removal of identical and low-variance features. The feature set is often further reduced by some form of feature selection (e.g. clustering). The final step is the analysis part, where ML models are built and validated.

different patients need to have their intensity values normalized to a homogeneous distribution (if not, this may introduce unnecessary variation). CT and PET images tend to be more similar, but there may be other reasons to perform preprocessing, especially if the dataset contains images from different sites. Differences in scanner type and scanning settings typically introduce an array of variations that need to be taken care of. Furthermore, it is often desirable to standardize all images, e.g. to zero mean and unit variance, if they are to be analyzed with a DL model.

Extracting the features is tantamount to deciding what type of features to extract and what type of images to calculate features from (i.e. the different processed images mentioned in the previous section). In some cases, e.g. when using a simple linear prediction model or when the purpose is to analyze the impact of individual features, it may be useful to only extract a small subset of features, but another common approach is to extract as many features as possible and rely on feature selection to simplify the analysis. It is not uncommon to end up with 1000-2000 features in total.

When a large set of features are extracted from a single image, it is not surprising that some features show a high degree of similarity to one another. Feature elimination aims to eliminate some of those features in order to simplify the subsequent mathematical analysis. This can be done in a variety of different ways, but since the features are similar by assumption, the choice of which one of the similar features to keep and which to discard tends to not be very impactful. A straightforward approach is to remove all features that have a correlation above a specific threshold value with at least one of the non-eliminated features.

Feature selection refers to selecting the features to use in the mathematical analysis, which is essential to combat the curse of dimensionality, and also leads to faster training times and more robust models. A very common type of dimensionality reduction technique used for feature selection is any type of clustering algorithm, particularly k-means or hierarchical clustering. By clustering the data into a smaller set of internally similar clusters, one can select one representative feature from each cluster instead using the full features set. This procedure also implicitly eliminates features that are very similar to each other. When clustering the data, one needs to select the parameter that determines the number of clusters.

The analysis part of the workflow usually starts with model selection. Researchers sometimes start with a specific model in mind suitable for the problem at hand, in which case model selection refers to selecting the best hyperparameters for it, but model selection also refers to selecting the best type of model from a set of candidates. For instance, there is a vast array of different classification models to choose from, and it is not possible to know which one will perform the best a priori. The standard way of selecting a model is to train and validate the candidates on the available data and simply select the one with the lowest validation error, which means that the selection, training, and validation steps are effectively merged into a single pipeline. This can be done fairly straightforwardly with some type of cross-validation.

1.4.4 Limitations of radiomics

Since the dawn of radiomics in 2012, the number of yearly radiomic studies has grown to roughly 10,000 in 2021 according to Google scholar. Crucially, the field is still in its infancy, and much work remains to be done before practical deployment and testing can become a reality. An important step in this direction will be the conduction of prospective trials, but such studies are much harder to organize and perform and require an elevated level of rigor in their study design. In order to perform reliable prospective studies (as well as retrospective studies for that matter), radiomics researchers need to establish the foundations needed to build, train, and evaluate reliable radiomic models.

One of the primary concerns surrounding radiomics has long been the issue of reproducibility and repeatability issue of the radiomic features[141,160,9,154]. In practice, this has become somewhat of an umbrella term for a number of underlying issues relating to reproducibility, where numerous sources of errors are sometimes conflated. The foremost problem is that many features are known to be irreproducible in the sense that features calculated from repeated scans of the same patient can return different values. This phenomenon has even been demonstrated in so-called phantom studies, where a toy-like inanimate test subject is scanned multiple times. Another source of variation lies in the segmentation procedure since different experts will produce (hopefully slightly) different segmentations on the same image, which will inevitably lead to differences in the feature calculations. But even in cases when some features have been shown to possess predictive power in one set of patients, the same features do not necessarily need to be predictive in another population. This type of dissimilarity is more akin to the reproducibility crisis that has been a known phenomenon in sociology for decades.

The reproducibility issues mentioned above are a legitimate cause for concern, but various strategies exist to alleviate the problems. Even though many features are irreproducible, some features are not. One can then either explicitly select reproducible features (e.g. those who have been shown to be stable in prior studies) or perform a similar analysis oneself. It is also possible to make the calculation of features more robust by performing image homogenization, e.g. by normalizing the intensities with respect to the intensities in relatively well-behaved tissues. In principle, a well-constructed model validation pipeline will also diminish the problems since the models will need to perform well on the validation set in order to be selected.

The next major limitation is the size of the data sets. It is not surprising that radiomics studies suffer from small datasets given that sharing patient data is generally disincentivized and collecting high-quality data can be hard, even if done internally. This severely hinders the conclusions that can be drawn. According to a recent review investigating 57 different studies in prostate cancer radiomics, the median sample size was just 102 patients (132 mean and 498 max)[37]. To put this in perspective, the median sample size in Wikipedia's list of datasets for machine-learning research for human biological data is 1176 (81,163 mean 1,223,009 max). In image analysis, the famous ImageNet[33] dataset, which has been criticized for being too narrow (albeit for very different reasons), has 1,281,167 images. The small sample size and large number of predictor variables lead to a related problem sometimes called the "small n large p problem". The problem this refers to is that, when the ratio $\frac{p}{n}$ is large enough, the probability of finding a spurious relationship (e.g. one that is due to chance rather than an underlying causality) between a predictor variable and the target approaches 1.

1.4.5 Why is radiomics hard?

With the widespread success of ML in other areas and the wide range of potential applications for AI in clinics as we saw in Figure 1.2, one might wonder why hospitals are not currently crawling with AI algorithms and prediction models. It would seem that the medical community has not yet started to widely deploy ML algorithms and methods to aid doctors and nurses in their everyday work. This section will cover some of the factors that contribute to these circumstances and what can and should be done to suppress them as well as possible. Three key interconnected factors play a large role: data, validation, and approval. Broadly speaking, it is not possible to thoroughly validate the models without large high-quality data sets, and without thorough validation, it is not possible to obtain approval from administrators and regulators. Without approval, it is very hard to deploy models and collect new high-quality data.

The tendency of the radiomics field to conduct studies on small data sets brings along a host of different problems. As noted earlier, it severely limits the conclusions that can be drawn, since small data sets are unlikely to be representative of the general population they are drawn from. But small data sets also make it easier for researchers to build overly optimistic models, e.g. by overfitting. They make it so that the variance of the model fitting procedure becomes very large, which in turn increases the bias in the model selection procedure.

A conceivable solution to the small data problem is to collect data from external public sources (if available) but even this poses its own limitations. It is not strictly clear that external data would actually help, since they may be sufficiently different from the internal data that it could hamper the overall performance on the internal data set. Even small differences in the acquisition protocol and scanner settings could lead to significant downstream effects, even if the qualitative differences are invisible to human observers.

Dataset differences also render black-and-white comparisons of different studies virtually meaningless, because there may be large variations in the distributions and difficulties within them. In fact, it is often not even possible to determine whether the results from a study are good or bad due to a lack of detailed descriptions of the code implementation. Even if the report describes a perfect training procedure, it is possible that the implementation suffers from fatal mistakes such as information leakage. The lack of large public benchmark databases of test samples, ideally with concealed labels so as to prevent "training-on-the-test-set", further exacerbates the problem.

Some researchers have also raised concerns about the clinical utility of some radiomics studies. When conducting research for the healthcare domain, the foremost question always ought to be whether the findings would be helpful. It is no use to build a prediction model for shoe size, for example, if this would not be interesting for the clinicians. Furthermore, shoe size can be easily obtained simply by asking the patient or, in extreme cases, by measuring the feet. In addition, even if a prediction model is good, it doesn't necessarily follow that it is better than what is currently used in practice. It is always useful to compare ML models with clinical practices if possible.

With the above issues in mind, there is no silver bullet that solves all problems. Ideally, studies and models should be designed while considering and being open about their limitations. A few main take-home messages can be devised:

- Putting effort into collecting more data and improving its quality is more likely to yield positive outcomes than conducting small proof-of-concept studies, especially for datasets smaller than a few hundred samples.

- The focus should be on implementing and using more stringent and rigorous validation procedures, such as nested and repeated cross-validation, rather than achieving high AUC or accuracies. The performance in a study has no inherent value unless its robustness can be guaranteed and compared with other models on the same data.

- The clinical utility should be established before models are trained. It doesn't matter how good a model is if it is not something that could help or inform clinicians.
- For clinical implementation to become a reality there needs to be good communication between researchers, administrators, doctors, and policymakers. The decision to start practical trials is not usually on those who know the most about the models, but it is up to the model developers to be truthful about the promises and limitations.

1.5 Thesis structure

This thesis is divided into three main chapters, each tackling a separate part of a patient's path: (I) Image acquisition & segmentation, (II) Image processing & radiomic feature variability, and (III) Computer-aided cancer assessment. A detailed description of the chapters follows:

- Chapter 2: Image acquisition & segmentation. This chapter tackles various aspects of organ segmentation in medical images, which is one of the first challenges that clinicians face when a patient is admitted for prostate cancer care. The defining question of this chapter is whether we can automate the segmentation process, and how to do this in the best way possible.
- Chapter 3: Image processing & radiomic feature variability. In this chapter, the issue of image normalization and its effects on the radiomic features is studied. This

critical step is often disregarded in medical imaging literature but is necessary for performing robust analyses.

- Chapter 4: Computer-aided cancer assessment. This chapter focuses on building and comparing prediction models for the assessment of the pathological status of prostate cancer patients. A deep dive into the predictions and behavior of the models is presented in a way that is relevant for both clinicians and model developers.

The thesis ends with a concluding chapter where the main findings are summarized and contextualized in terms of how they fit into the current clinical practice.

2

Image acquisition & segmentation

2.1 Background

Segmentation of anatomical structures in medical images is a vital step in many clinical domains including radiology, pathology, ophthalmology, dermatology, and microscopy[114,127,107,136]. For instance, accurate delineations of neighboring organs are crucial for calculating dose and estimating the risk of normal tissue complications in radiotherapeutic treatment planning. Typically, the regions of interest (ROIs) are manually or semi-automatically hand-drawn by trained medical personnel in treatment planning systems, but there is a tremendous incentive for streamlining or automating this process since it is incredibly time-consuming (especially for less strictly defined regions such as tumors) and requires trained experts. Furthermore, segmentations from human experts suffer from poor reproducibility and/or accuracy[11,126,131,147]. The emerging discipline of radiomics and other computer-aided prediction frameworks has further increased the need for accurate and consistent segmentations since the calculation of the radiomic features requires ROIs. This begs the question of whether we can use AI to segment organs and ROIs instead.

Computer-assisted segmentation is nothing new: atlas-based algorithms with applications in segmentation have been around since 2004[32], and the concept of fuzzy connectedness which creates fuzzily-connected regions in images existed even before that[142] (1996). Atlas-based segmentation is currently implemented as an option in clinical treatment planning software such as Raystation from RaySearch Laboratories. An *atlas* in this context refers to a reference image (or multiple images in the case of a multi-atlas) that can be used as a basis to distort similar images to align with it. If a segmentation map is known for the atlas, one can then apply the segmentation to the distorted image, then revert the distortion for both the image and the segmentation map to achieve a final segmentation. More recently, ML engineers have focused on segmentation with DL models, with a major driving force being its application to self-driving cars.

The current state-of-the-art models for automatic image segmentation are all DL-based. These are typically characterized by the heavy use of convolutional operations, which enable the models to successively and automatically extract relevant features at different resolutions and locations in the images. Since its introduction in 2015, the U-net DL segmentation model has become the most widely used model for medical both image segmentation[118] and other domains. At present, virtually all current high-performing DL-based models are evolutions of the U-net that have incorporated various modifications, such as flattened operations[75], bottleneck convolutions[57], or attention mechanisms[90]. However, instead of common trends appearing, increasingly different approaches and models are seeing use. Moreover, even for a given segmentation task, reports on different segmentation techniques are hard to compare, since they typically use different data sets, and suitable public benchmark medical image data sets are often not available or readily accessible. A foreseeable problem is thus that engineers in clinical institutions will become perplexed when faced with the choice of which segmentation model to implement because the space of possibilities is simply too large.

Even though DL models have become the de facto ruler of automatic segmentation models, the data size problem remains an issue. As discussed previously, in what may be called the central dogma of DL, DL models tend to require huge amounts of data to work well[*]. In the original U-net publication, the authors used the "EM stacks" segmentation challenge dataset of ISBI 2012, which contains just 30 images (512 × 512 pixels) from serial section transmission electron microscopy. But each such image contains hundreds of cells to be segmented, which translates to some ~6000 signal-generating structures. Comparing this to a typical dataset of about 100 prostate MRI scans (each with roughly five slices of prostate tissue), the difference in the signal-per-image ratio is clear. In principle, it is con-

[*] It is still not well understood why DL models are so data-hungry. Some researchers have suggested that the overparameterization of DL models is the key factor that enables them to interpolate *smoothly*, which in turn leads to their superior performance. Learning this many parameters, as the argument goes, is the ingredient that demands such large amounts of data.

ceivable that traditional models like atlas-based segmentations could be more effective for the simple prostate geometry in the small-data regime, but the magnitude of such an effect has not been studied.

One of the key methods in the seminal U-net paper was the extensive use of data augmentation, which is a term used to describe techniques that artificially enlarge the data set by transforming the input into novel-looking samples. Common techniques in image analysis include rotation, flipping images up/down/left/right, zooming images in or out, color adjustments, and elastic deformation. The aim when augmenting the images in the training set of the model is to reduce its generalization error. Although the augmented images are normally obviously just slight variations of the original, the technique is surprisingly effective for AI models, presumably because they are not learning concepts in a semantically meaningful way as humans do. Since medical datasets are extraordinarily small compared to other applications such as urban scenes used for autonomous driving (the popular cityscapes dataset contains 25,000 annotated images[27]), data augmentation may play a decisive role in clinical scenarios. To achieve clinically acceptable segmentation performance with few data samples, it will be critical to explore the possible ways to enhance the training procedure, e.g. by effective use of data augmentation.

Most contemporary research in medical image segmentation focuses on developing and applying automatic segmentation procedures to reduce the workload of clinicians, speed up the delineation process, and improve the segmentation quality. As the performance of these models has improved, institutions are looking to start experimenting with these models in clinical practice. In the course of research focused on developing and validating the models, however, several aspects of model deployment have been left unaddressed, including model drift, underspecification (that is, when pipelines can return many predictors with equivalent training performance, but with very different deployment performance)[30], dataset & model biases, and quality assurance. Indeed, the authors of a recent survey of AI in radiation oncology[61] argued that there is an unmet need for guidance on the implementation and use of AI models in clinical practice.

Quality assurance (QA) stands out as one of the key steps in the deployment of AI algorithms that so far has been largely left out in contemporary medical imaging research[28,147,17,122,62]. In general, it refers to the practice of monitoring the output, performance, and user experience of a deployed method or model to ensure that it is working as intended. This is of particular importance in medical contexts, where patient outcomes may be jeopardized. Previously, the role of humans in this step has been mostly subsumed, but there appears to be no principled reason why this cannot be carried out by AI algorithms as well. Despite the potential benefit of AI and ML for QA being recognized[145], there is surprisingly little literature on the topic, particularly in the field of image segmentation.

If automatic segmentation can reduce the workload of clinicians and improve both the speed and accuracy of segmentations, why is manual segmentation still the method of choice in clinical institutions? This chapter delves deeper into what it takes to train autosegmentation models and how to best do this with a limited amount of data. Our contribution can be roughly divided into three parts:

1. Data augmentation: How can limited data best be leveraged to train segmentation models to an acceptable standard? Can data augmentation be leveraged in a particular way suitable for small clinical datasets?

   To this end, we will investigate whether we can use a new data augmentation technique called mixup in conjunction with standard augmentation methods to improve the segmentation performance of deep learning models. The work presented in this section is based on results published in Isaksson 2022[70].

2. Automatic segmentation: How do the most common DL architectures and training strategies compare against one another and atlas-based segmentation when it comes to prostate segmentation? Is it possible to solve the auto segmentation problem to a clinically accepted degree in the low data regime?

   To answer these questions, we will train and compare the most common DL segmentation strategies and atlas-based segmentation on a set of 100 MRI images from prostate cancer patients. Most of the material in this section is based on Isaksson 2022[67] and is currently under review.

3. Quality assurance: How do we ensure that automatic segmentation delivers proper delineations? Can we monitor the quality in such a way that we would be aware of when human intervention is required?

   For this purpose, we will design and train a new type of DL model to specifically predict the quality of automatically generated contours. This section is based on material published in Insights Into Imaging (see Isaksson 2022[69]).

2.2 Dataset

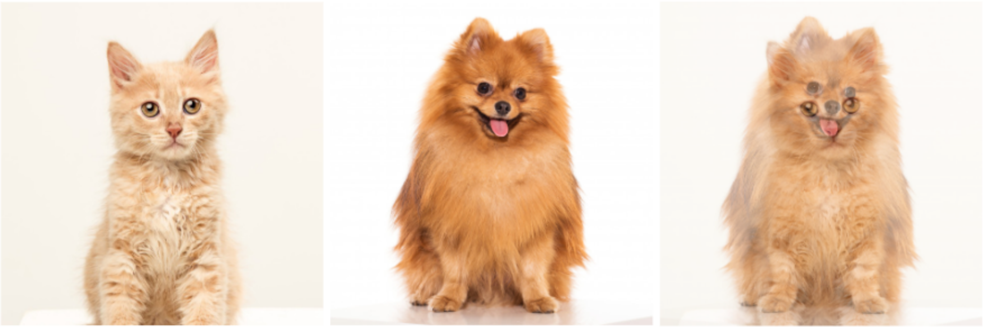
For the work in this chapter, a data set of 100 T2-weighted MRI prostate scans from patients at IEO European Institute of Oncology IRCCS, Milan, Italy was used. All patients gave their consent for use of their data for research and educational purposes, and the use of the data was approved by the local Ethical Committee, which waived the requirement for further consent specific to this study. The images were acquired using a 1.5 T scanner (slice thickness 3.0-3.6 mm, slice gap 0.3 mm, pixel spacing 0.59×0.59 mm, echo time 118 ms, and repetition time 3780 ms). For every image, a ground truth segmentation was established by consensus from two experienced radiologists with more than five years of experience.

The MRI volumes were resampled to a common size of $320 \times 320 \times 28$ using bilinear interpolation in the three cases where the image was larger than 320×320, and zero padding where there were fewer than 28 slices. Each image was corrected with the N4 bias field correction algorithm using the SimpleITK 2.0.2 python package with default parameters. Within each image, the intensities were clipped to the 0^{th} and 99^{th} percentile interval, and subsequently standardized to the $[0, 4033]$ range[*]. For the DL-based applications, each image was also normalized to zero mean and unit variance.

2.3 Mixup data augmentation for segmentation models

Researchers address the generalization problem of deep image processing networks mainly through extensive use of data augmentation techniques such as random flips, rotations, and deformations. This section investigates whether a novel data augmentation technique called mixup, which constructs virtual training samples from convex combinations of inputs, can be used to improve the performance of DL segmentation models. The technique was recently proposed for deep classification networks, where it improved the performance on a variety of datasets, but so far it has not been evaluated for image segmentation tasks.

[*] 4033 was the maximum intensity in our data set.



(a) *Cat* (b) *Dog* (c) $0.5 \cdot Cat + 0.5 \cdot Dog$

Figure 2.1: Example of a simple linear combination of two images. The combination of the two may not be very helpful for teaching humans how to distinguish between cats and dogs, but it may be beneficial for teaching machines to generalize well.[†]

Accordingly, we will train a variant of the U-net architecture to segment the prostate, and compare the results with and without mixup in terms of Dice similarity coefficient and mean surface distance from the reference ground truth segmentation.

2.3.1 The mixup algorithm

The mixup algorithm[156] involves constructing new samples $\hat{\mathbf{x}}$ from different training samples $\mathbf{x}_i$ and $\mathbf{x}_j$ according to $\hat{\mathbf{x}} = \lambda \mathbf{x}_i + (1 - \lambda)\mathbf{x}_j$. For an illustrative example of how this looks, see Figure 2.1. The λ parameter is drawn from a symmetric beta distribution, i.e. $\lambda \sim \beta(\alpha, \alpha)$, where α is a hyperparameter of the transformation. As such, one can draw λ from different distributions by selecting different α-values: for $\alpha = 1$, λ will be drawn from a uniform distribution, and for $\alpha \to \infty$, λ will approach a delta function centered at $\lambda = 0.5$.

[†] Images designed by Freepik.com

2.3.2 Methods

Hyperparameter search

A hyperparameter exploration was first conducted in order to select the best value for the mixup parameter α. For this experiment, we randomly split the data into two equally sized (N=50) training and hold-out test sets. After splitting, the clinical variable distributions of the two sets were tested to be similar (Wilcoxon signed rank test), which allowed us to control for adverse effects stemming from different clinical traits between the two groups. For each value of α (including $\alpha = 0$ for no mixup), the network was trained until convergence eight times with different initializations (see sections below for training and implementation details). The best value of α was selected for the remainder of our experiments.

Quantifying segmentation performance

We performed a random five-fold cross-validation procedure comparing the best value of the mixup parameter α with no mixup. In addition to mixup, the samples were augmented with a standard scheme of random horizontal flips, random zoom and translation (with a scale factor in $[0.5, 1.5]$), and random rotations in the $[-\frac{\pi}{4}, \frac{\pi}{4}]$ range. This enabled us to gauge how well mixup works in conjunction with other data augmentation techniques, similar to how one would do it in practice.

Segmentation performance was quantified by the Sørensen–Dice similarity coefficient (Dice) and mean surface distance (MSD) between the network outputs and the ground truths. In addition, we included the absolute relative volume difference (ARVD), 95th percentile Hausdorff distance (HD95), and sensitivity, which are other common metrics seen in segmentation studies.

The **Dice** coefficient is defined by

$$\text{Dice} = \frac{2|A \cap B|}{|A| + |B|} \tag{2.1}$$

where A and B are the sets of pixels of the structures being compared. As such, the Dice coefficients ranges from 0 to 1, where 1 corresponds to perfectly overlapping objects, and 0 corresponds to any configuration where their intersection is zero.

The **MSD** (measured in pixels) between the surfaces A_s and B_s of A and B, is defined as

$$\text{MSD}(A, B) = \frac{\sum_{a \in A_s} \text{d}(a, B_s) + \sum_{b \in B_s} \text{d}(b, A_s)}{|A_s| + |B_s|} \tag{2.2}$$

using the Euclidean distance from pixel $a \in A_s$ to surface B_s given by $\text{d}(a, B_s) = \min_{b \in B_s} ||a - b||$. Since manual segmentations are drawn and evaluated on a slice-by-slice basis (and since the pixel spacing is spatially anisotropic), we calculated this quantity in 2D (within-slice). This means, however, that the distance will be undefined in slices with at least one empty contour set. To avoid this, we redefine the empty contour as a single pixel in the center of the image.

ARVD is the absolute volume difference calculated relative to the volume of the ground truth (in this case B), defined by:

$$\text{ARVD}(A, B) = |\frac{V_A - V_B}{V_B}|. \tag{2.3}$$

An ARVD of 0.3, therefore, means that the predicted prostate is either 30% larger or 30%

smaller than the ground truth. A perfect score of 0 indicates that the objects have identical volume, but does not necessarily mean that the prostates are well aligned spatially.

HD95: the 95th percentile Hausdorff distance between two geometrical objects A and B is defined by:

$$\mathrm{HD95}(A, B) = \max\left\{\sup_{a\in A}\inf_{b\in B} \mathrm{d}(a, b), \sup_{b\in B}\inf_{a\in A} \mathrm{d}(a, b)\right\} \tag{2.4}$$

where the supremum functions in this case return the 95th percentile values, making the metric more robust to irregularities*. As with the MSD, we calculated this quantity on a slice-by-slice basis (i.e. in 2D).

Network architecture

The architecture used for the segmentation network was a variant of the classic U-net as illustrated in Figure 2.2 with various modifications inspired by high-performing submissions to the PROMISE12 online prostate segmentation challenge[48]. The main features of the network are:

1. All convolutions were applied with 3D $3 \times 3 \times 3$ kernels.
2. Fewer filters were used, and the number of filters at deeper levels of the network was increased by a constant amount (+14) instead of a multiplying factor (typically $\times 2$).
3. Each level performs a single convolutional operation instead of two or more in series.

*For an intuitive understanding of the HD, consider the following algorithm: 1) for every point in A, take the minimum euclidean distance to B; 2) take the maximum from among such distances; 3) repeat steps one and two for object B and its distance to A; 4) the HD between A and B (and, equivalently, B and A) is the maximum of the two values obtained from step two.

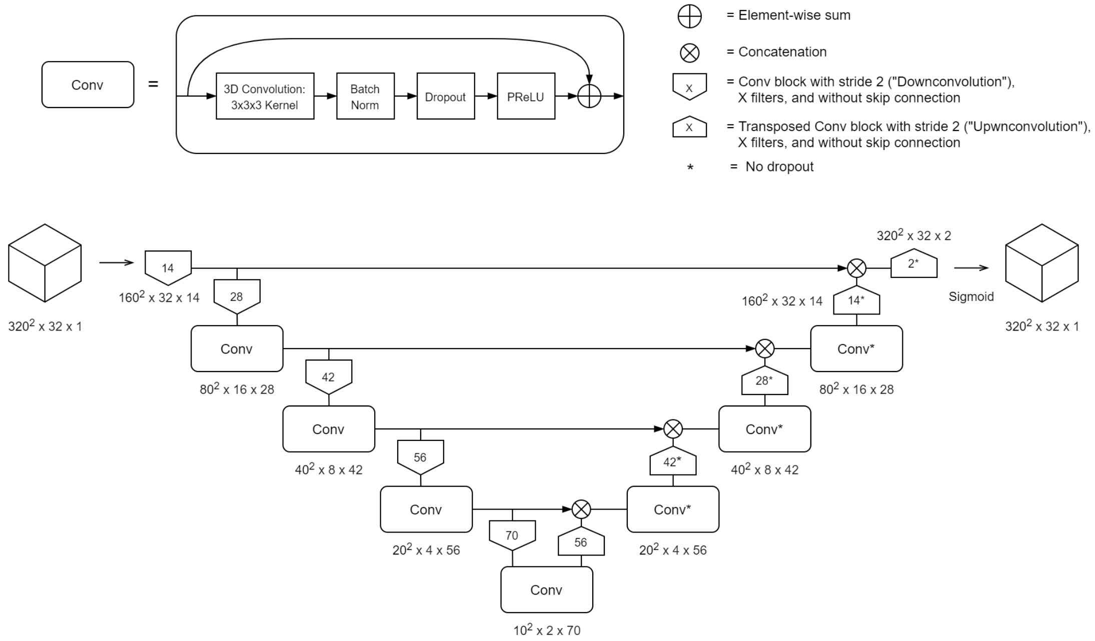


Figure 2.2: Network architecture used for analyzing the effectiveness of mixup data augmentation. The numbers outside the boxes display the tensor size at different levels, where the last number indicates the number of channels.

4. The first operation is strided instead of a normal convolution in order to downsample the images such that the computational resources could be managed more efficiently.

5. A PReLU[56] activation function was used instead of the more common ReLU function.

6. Dropout with a rate of 0.5 was applied in the encoding part of the network.

These design choices were motivated by heuristic comparisons, favoring simplicity and speed over minor performance gains.

Model training

As the loss function, we used the top-k cross entropy[152], which calculates the pixel-wise cross-entropy, but only considers the top k pixels as contributing to the final loss. This allows the network to focus training on hard-to-classify pixels (likely within the edges of the prostate) and also speeds up training. The parameter k was set to 5% of the number of pixels in the images (in this case $k = 143\,360$) times the batch size, which was set to 8.

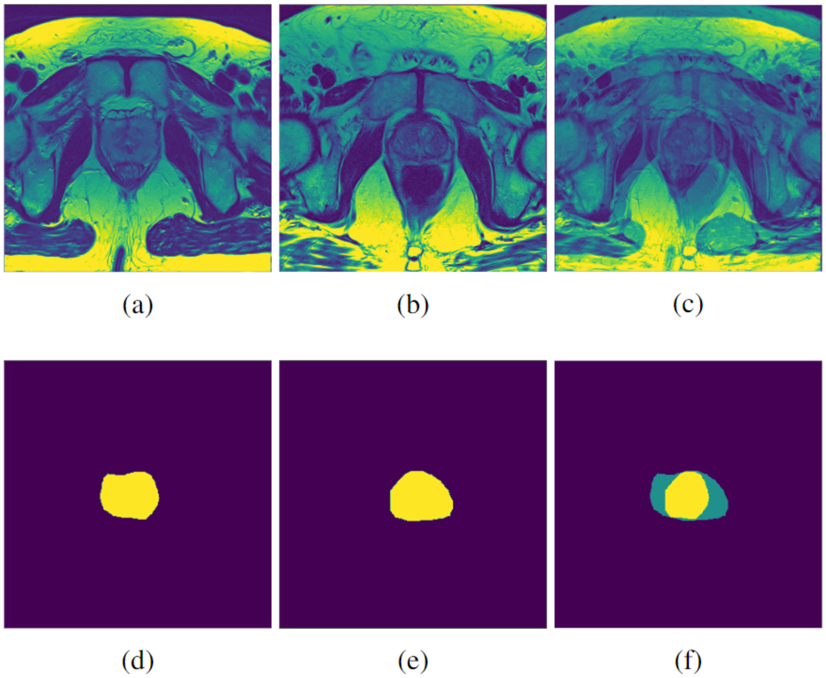
The models were trained for 350 epochs with the Adam optimizer[81] with default learning parameters ($\beta_1 = 0.9, \beta_2 = 0.999$), a learning rate of 0.0005, and extended with the lookahead mechanism[158] (sync period = 6, step size = 0.5) to reduce its variance. The best weights were restored prior to evaluation. The implementation was done in Python 3.7 with TensorFlow 2.3 running on an Nvidia Tesla K80 GPU (16 GB).

2.3.3 Results

A visualization of the mixup technique for two samples in our data set is presented in Figure 2.3. It is clear that the mixup image does not depict a realistic appearance for a prostate MRI, and the target mask of the augmented sample is no longer binary.

The mixup implementation did not introduce any significant computational overhead; less than 0.04 s per batch of 8 images (roughly 0.1% increase).

The parameter search concluded that the best choice for α was $\alpha = 0.5$ in terms of both Dice and MSD (see Figure 2.4). Both $\alpha = 0.5$ and $\alpha = 0.7$ were significant improvements (Mann-Whitney U-test) compared to non-mixup training for both evaluation metrics. Other α values also outperformed non-mixup training, albeit without statistical significance. The mean performance of $\alpha = 0.5$ resulted in a 1.46% increase in Dice and a



(a) (b) (c)

(d) (e) (f)

Figure 2.3: Examples of prostate MRI scans (a, b) from two patients with accompanying prostate masks (d, e). The right images (c, f) show the results of the mixup data augmentation technique ($\lambda = 0.5$) for the two patients. Note that mixup renders the target mask non-binary.

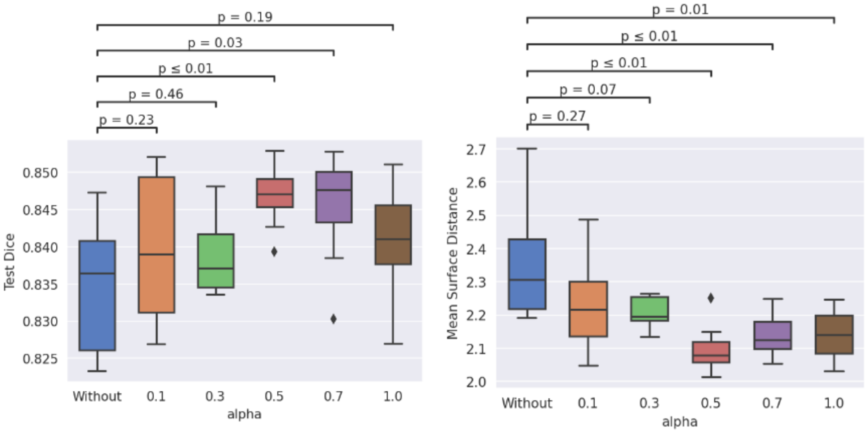


Figure 2.4: Test results of the α parameter search on the holdout test set. Results are aggregated from eight different initializations. A good performance is characterized by a high Dice and low surface distance. p-values indicate the significance of the Mann–Whitney U-test.

10.9% decrease in MSD over non-mixup training.

In the cross-validation experiment with $\alpha = 0.5$, mixup training increased mean Dice by 1.91% ($p = 0.005$) and decreased mean MSD by 10.7% ($p = 0.054$) over non-mixup training (see Figure 2.5). The analysis of the other metrics (absolute relative volume difference, 95th percentile Hausdorff distance, and sensitivity) revealed that mixup training is superior in every case. The differences were significant in every case apart from MSD and absolute relative volume difference (see Table 2.1).

Training curves from the k-fold cross validation are displayed in Figure 2.6. The time to convergence seems to be largely unaffected by the use of mixup, and mixup training seems to generalize better and with lower variance compared to non-mixup training, even though the training loss is very similar.

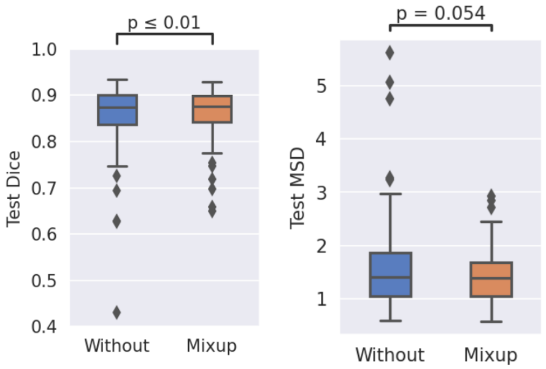


Figure 2.5: Cross-validation results from training without mixup and training with mixup with $\alpha = 0.5$. p-values indicate the significance of the pair-wise Wilcoxon signed-rank test. The mean pairwise differences of Dice and MSD were 0.016 (3.15%) and -0.207 (-13.9%), respectively.

Table 2.1: Mean values of different commonly used performance metrics for the cross-validated test performance. Parentheses indicate the 95% CI and p-values show the significance of the Wilcoxon signed-rank test. ARVD: absolute relative volume difference, HD95: 95$^{\text{th}}$ percentile Hausdorff distance, Sen: sensitivity.

	No mixup	mixup	p-value
Dice	0.839 (±0.03)	0.855 (±0.02)	0.005
MSD	1.93 (±0.57)	1.73 (±0.49)	0.054
ARVD	0.088 (±0.04)	0.066 (±0.03)	0.155
HD95	6.46 (±1.41)	5.78 (±1.45)	0.011
Sen	0.812 (±0.03)	0.832 (±0.03)	0.036

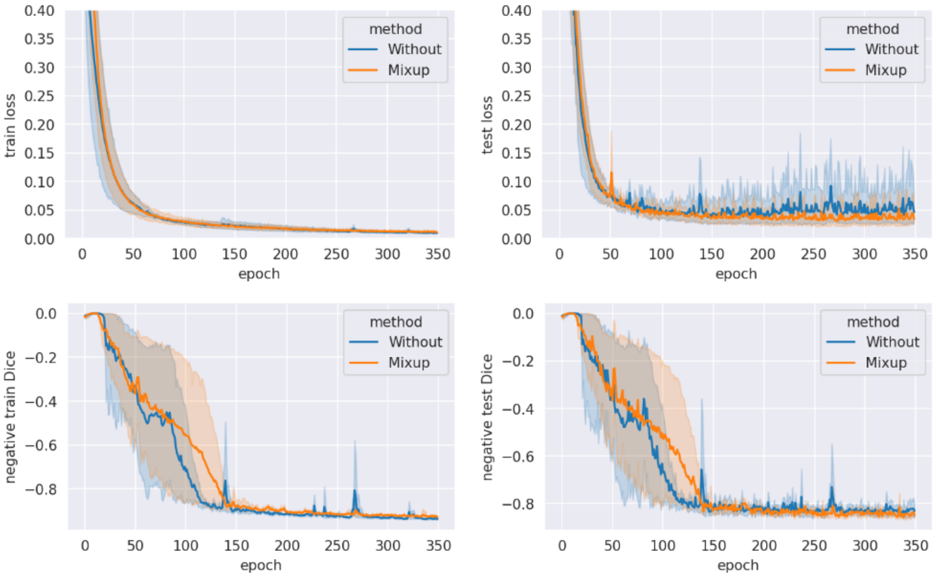


Figure 2.6: Training curves from the five-fold cross-validation without mixup and mixup with $\alpha = 0.5$. The large Dice variance in the early stages of training is a feature of the loss function: the time it takes for top-k pixel loss to generalize globally varies strongly between runs.

2.4 Comparing automatic segmentation models

In this section, we compare four automatic DL segmentation models on a single data set consisting of 100 prostate cancer patients. We also benchmark the DL models against multi-atlas segmentation and a proprietary segmentation algorithm in the Syngo.via clinical imaging software developed by Siemens Healthineers. The models were chosen to cover some of the most essential training strategies (2D vs. 3D, training from scratch vs. transfer learning) and architectures (V-net, GANs, EfficientNet). The following DL architectures were included in the comparison:

- V-net: a 3D evolution of the U-net that uses 3D convolutions instead of 2D. The V-net architecture also incorporated residual connections and replaced the pooling operations (used for up-sampling and down-sampling) with convolutions.

- Transfer-learned U-net: a U-net with encoder weights pre-trained on the ImageNet[33] dataset.
- GAN (generative adversarial network): a GAN extension of the transfer-learned U-net.
- EfficientDet: a version of the EfficientDet object detection architecture modified for segmentation purposes.

2.4.1 Methods

Atlas segmentation

Multi-atlas segmentation is a common automatic segmentation method built on ideas from deformable registration and coordinate transformation[103,26,31]. Even though they have recently fallen out of favor since the introduction of DL, they are still central to many treatment planning software packages, which makes them relatively accessible. The atlas method used herein is embedded in the RayStation 9B treatment planning system commonly used in radiation oncology. The implementation is based on the anatomically constrained deformation algorithm (ANACONDA), which combines image intensity data with anatomical information using contoured image sets. This solution requires the user to define atlases of images and contours, which the algorithm then uses to find a coordinate transform between the new image and the already segmented images in the atlas. If the atlas contains images similar to the new image, the coordinate registration will be more accurate, which will result in a more credible segmentation.

Siemens (Syngo.Via software)

The Syngo.Via medical image reading and post-processing software offers a built-in automatic segmentation method based on DL[130]. Partial details on its implementation have been published in relation to liver segmentation, but further information remains undisclosed to the public. Much of the appeal of this approach is that it is implemented in software that many clinicians are already familiar with and use in their existing clinical routine, which increases its potential for integration and decreases the learning curve.

V-net

The basis of the U-net segmentation model and its successors (including the V-net[104]) is its encoder-decoder architecture: an encoder first distills relevant information about the input image into representative features, then a decoder extracts useful information from these features into a desired structure—in this case, a binary segmentation matrix. The encoder and decoder are in turn built up from serially connected convolution blocks that operate at different resolutions by using intermediate upsampling and downsampling operations. The encoder and decoder are also connected horizontally by skip connections between each resolution. The primary modifications in the V-net design were to use 3D convolutions for volumetric medical images and to incorporate residual convolution blocks. The pooling operations that were used for up- and down-sampling were also replaced with strided convolutions and transposed convolutions (colloquially often called de-convolutions), respectively.

Our V-net implementation is summarized in Figure 2.7. The residual convolutional blocks we used can be formalized as

$$\mathrm{ResConv}(x) = \mathrm{PReLU}(\mathrm{Dropout}(\mathrm{BatchNorm}(\mathrm{Conv3D}(x)))) + x \tag{2.5}$$

where PReLU is the parameterized rectified linear unit activation function[56]. We use non-spatial dropout with a rate of 0.5 and only apply it in the encoder. All the 3D convolutions used a $3 \times 3 \times 3$ kernel size apart form the transposed convolutions (used for upsampling) which used $2 \times 2 \times 2$. The first convolution is a single strided convolution (e.g. not a residual block) with 48 filters that downscales the resolution from $320 \times 320 \times 28$ to $160 \times 160 \times 28$. Every successive level adds (or subtracts in the decoder stage) another 48 filters apart from the last transposed convolution, which uses two filters. We also use concatenations for the horizontal connections. The output layer is a single $1 \times 1 \times 1$ 3D convolution with a sigmoid activation function.

Transfer learning

Transfer learning (TL) is a common learning approach when data or resources are scarce or otherwise tainted (e.g. by poor quality). The idea is to apply pre-trained weights obtained on a much larger dataset and/or a broader task in order to save computing resources and leverage the robustness of previously learned features. Hence, it's a convenient choice in medical image analysis where data is often sparse.

Our TL approach consisted of a U-net decoder stacked on top of an EfficientNetB0 encoder (Figure 2.8) pre-trained on the 2012 ILSVRC ImageNet[33] dataset for images classification. We chose the EfficientNetB0 backbone for its performance and efficiency; its authors demonstrated state-of-the-art performance on ImageNet while being up to 8.4x

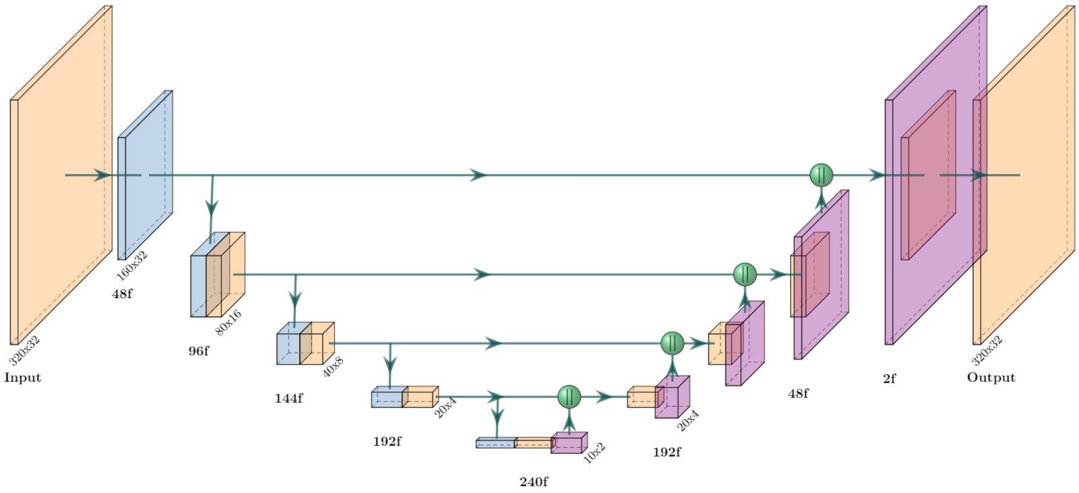


Figure 2.7: Architecture of our V-net segmentation network. Each block represents a residual convolution block (see (2.5)); strided convolutions (downsampling) are blue and transposed convolutions (upsampling) are purple. The output resolution at each level is displayed and the number of filters/channels is displayed with the 'f' suffix. Green spheres represent concatenation. The last block performs 1x1x1 convolution with a sigmoid activation function.

smaller and 6.1x faster than previous models. A key aspect of the EfficientNet architecture is that it processes the images and intermediate features with mobile inverted bottleneck convolution (MBConvs) blocks (see Figure 2.9 for details). Since the ImageNet samples are 2D images, this architecture can also only process 2D inputs, which means that the implementation operates on a slice-by-slice basis. To implement this model, we used the Segmentation Models[155] python package.

To transform the 320×320×28 MRI scans into the 224×224×3 images required of the ImageNet weights, an initial $3 \times 3 \times 3$ convolution maps each slice into three-channelled images. They were then center-cropped into 224×224. The final network output was constructed by concatenating and zero-padding them back to their original $320 \times 320 \times 28$ resolution. The number of filters in the decoder was set to {360, 288, 216, 144, 72} for levels $\{P_7, P_6, P_5, P_4, P_3\}$, and their kernel size was set to 4×4. Other parameters were

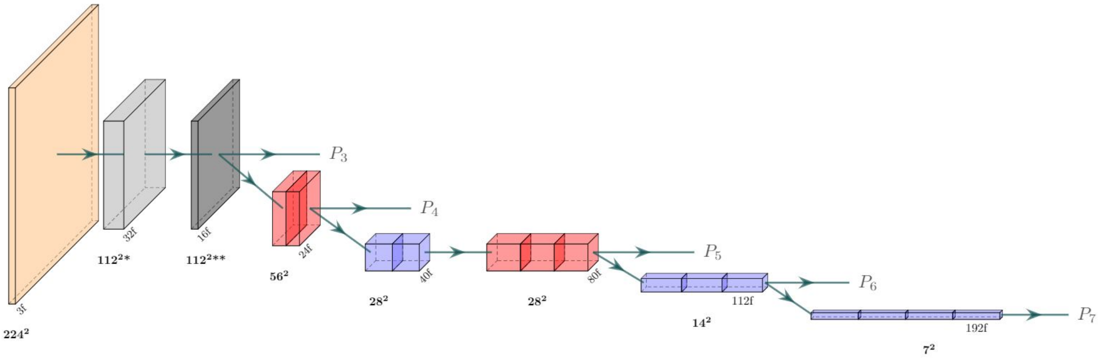


Figure 2.8: Architecture of the EfficientNetB0 backbone used in our transfer learning model. Blue and red blocks represent mobile inverted bottleneck convolution (MBConv) blocks (see Fig. 2.9), with a kernel size of 3x3 and 5x5, respectively. The light gray block represents a strided convolution followed by batch norm and a swish activation, while the dark gray block represents an MBConv block with expansion factor 1 and kernel size 3. The resolution at each level is displayed in bold and the number of filters (output channels) is displayed with the 'f' suffix.

configured to match our U-net implementation.

Generative Adversarial Network (GAN)

GANs are a family of models where two network agents—one generator and one adversarial/critic/ discriminator—compete against each other with a shared objective function. The generator is trained to generate "fake" samples (in our case segmentation masks) with the aim of trying to fool the discriminator into thinking they are real, while the discriminator is trained to distinguish fake/generated samples from real ones. As the generator gets better at generating realistic-looking samples, the discriminator has a harder time identifying them as fake. And when the discriminator improves its discrimination performance, the generator needs to generate more realistic-looking samples in order to keep up, ideally leading to a self-improving feedback loop. One compelling feature of GANs is that their objective function is implicit in the architecture, leading to a type of qualitative optimization.

Our GAN implementation was based on the pix2pix[71] framework for image-to-image

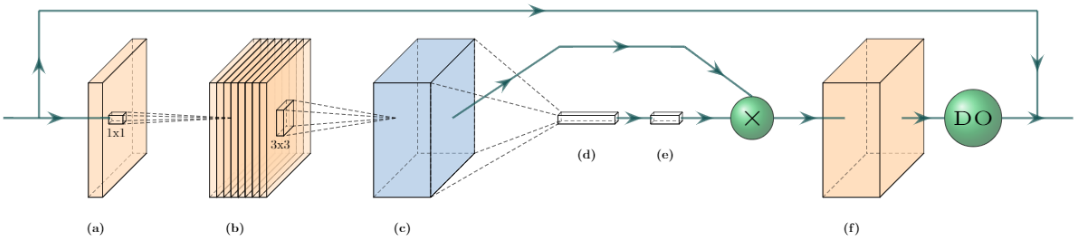


Figure 2.9: The mobile inverted bottleneck convolution (MBConv) block. (a) An initial 1x1 conv block expands the number of input channels according to the expansion factor hyper-parameter. (b) Depth-wise 3x3 conv block over channels. (c) Global average pooling shrinks the tensor along its spatial dimensions. (d, e) A squeeze conv (1x1 conv + swish) and an excitation conv (1x1 conv + sigmoid) first squeeze the channel dimension by a factor of 0.25, then expand it back to its original shape. The output is multiplied by the output tensor from step (b). (f) A final 1x1 conv block with a linear activation maps the tensor to the desired number of output channels, followed by a dropout layer for stochastic depth (dropout rate 0.2)

generation. The model was created by adding a binary classifier (see Figure 2.10) on top of a generating network, for which we chose the same architecture as our TL model. This model was chosen because its fixed encoder weights provide more stability during training (which is one of the primary challenges with GAN training). As in the pix2pix implementation, all convolutions in our discriminator are applied with a $4 \times 4 \times 4$ kernel, and no batch normalization is applied in the first convolutional layer. The dropout ratio was set to 0.5.

Segmentation-adapted 3D EfficientDet

EfficientDet[138] is an architecture intended for object detection that is an extension of the popular EfficientNet[137] model for object classification developed by researchers at Google. While neither architecture is intended for segmentation, the design is innovative and popular enough to warrant exploration in the segmentation regime as well. One of the defining factors of the "Efficient"-models is their relatively high speed, efficiency, and small size,

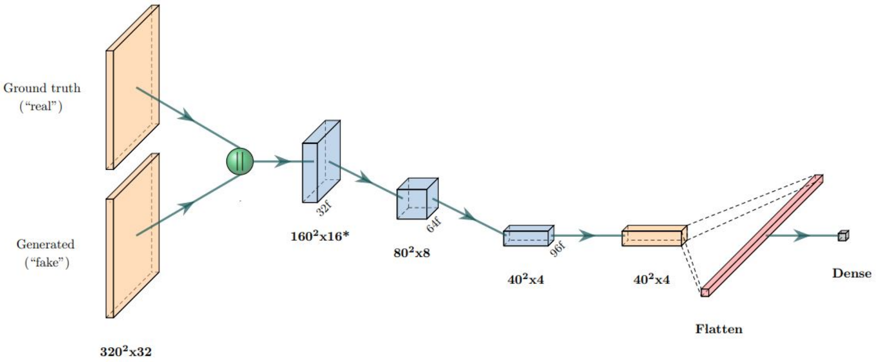


Figure 2.10: Architecture of the GAN discriminator. The ground truth and generated masks are first concatenated, then passed through consecutive strided convolution blocks (blue) and one regular convolution block. All blocks are non-residual Convolution blocks with $4 \times 4 \times 4$ kernels followed by batch norm (apart from the first strided block) and a PReLU activation. The final dense layer uses a linear activation.

which may prove useful in clinical contexts since hospitals usually need to train and deploy their models in-house without access to data centers or powerful GPUs. The original models were reportedly up to 8.4x smaller, 6.1x faster, and used 13x – 42x fewer FLOPs compared to their competition while still maintaining state-of-the-art performance (at the time).

The standard EfficientDet model consists of an EfficientNet backbone (see Figure 2.8) and a series of sequential bidirectional feature pyramid (BiFPN)-layers. The BiFPN layers aggregate features at different resolutions by applying the novel fast normalized fusion technique, which allows the network to attend to individual input features according to a learned relative importance, defined by

$$O = \sum_i \frac{\mathrm{ReLU}(\omega_i)}{\varepsilon + \sum_j \omega_j} \cdot I_i \tag{2.6}$$

where ω_i are the learned weights, $\mathbf{I}$ is the input, and ε is a small value that prevents numerical instability. The shape of $\boldsymbol{\omega}$ determines the type of attention: feature/input attention if it's a scalar, channel attention if it's a vector, or pixel attention if it's a multidimensional tensor. The architecture also incorporates depthwise separable convolution to speed up the network and reduces its memory requirement.

The EfficientDet architecture can be modified for segmentation with just minor changes. While the original model applied multiple BiFPN layers in succession, we noticed that stacking BiFPNs deteriorated the performance in the focused and relatively simple task of prostate segmentation. Instead, our implementation applies the fast normalized fusion technique from the BiFPN layer directly to the outputs of the EfficientNet backbone (see Figure 2.11). This greatly improves both speed and memory requirements further. To further accommodate the network for segmentation and our computational resources, we applied the following changes:

- To adapt the network to 3D, the $N \times N$ convolutions were replaced with $N \times N \times 3$ convolutions in cases where $N \neq 1$
- The number of filters in the P_1 to P_7 levels were set to $\{32, 24, 48, 48, 64, 80, 96\}$ (as opposed to $\{32, 16, 24, 40, 80, 112, 192\}$).
- The fast normalized fusion was applied over channels and inputs (instead of just inputs).
- The expansion factor in the MBConvs was set to 2.0 (instead of 6.0).
- The upscaling was done by a nearest-neighbor resizing followed by a $3 \times 3 \times 1$ anti-aliasing convolution (instead of only nearest-neighbor upscaling)

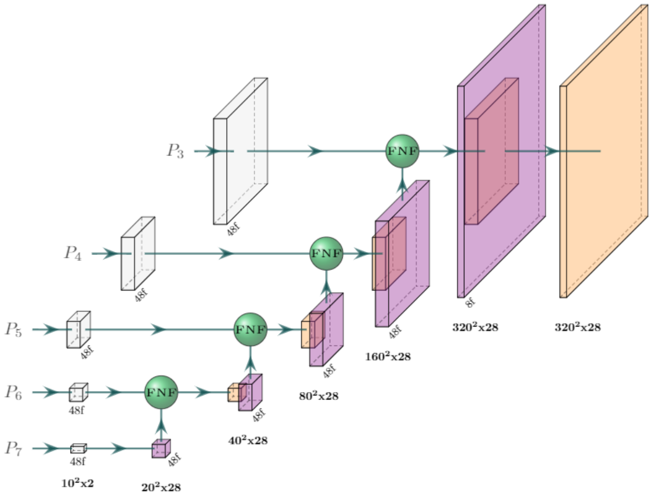


Figure 2.11: Architecture of the segmentation head in the 3D EfficientDet network. The P_3-P_7 output features from the EfficientNetB0 backbone (see Figure 2.8) are first convolved to a common channel dimension of 48 with $1 \times 1 \times 1$ filters (in white), then iteratively added with outputs from lower levels by fast normalized feature fusion (Eq. 2.6). The upscaling (purple blocks) is done by a nearest-neighbor resize followed by a $3 \times 3 \times 1$ anti-aliasing convolution. All convolutions are performed depth-wise and are followed by batch norm and a swish activation function, apart from the last block, which is a single convolution with a $1 \times 1 \times 1$ kernel and a sigmoid activation function.

Experiments

The images were randomly divided into two different training and testing sets on which the models were evaluated: a 70/30 split and a 50/50 split. This allowed us to validate our results with repeated measurements and to test the reliability of the models in terms of training set size. In order to not let particular clinical characteristics confound the results of the study, we checked that the distribution of prostate volume and extraprostatic extension were similar within the training and test sets using the Mann-Whitney U-test.

Each model was trained and evaluated once on the 70/30 train/test-set split and once on the 50/50 split. The evaluation was based on the Dice score, absolute relative volume difference (ARVD), mean surface distance (MSD), and 95th-percentile Hausdorff distance

(HD95) to the reference standard (ground truth), which are all defined in Section 2.3.2.

The following analyses were performed on the segmentation results:

1. We tested how much better the best performing model was compared to the others, and analyzed how significant the differences were (Wilcoxon signed-rank test).

2. We analyzed the performance of the methods in terms of the most poorly segmented patients. Since the segmentations are made on images from real patients, it is of great importance to not jeopardize any downstream implications resulting from unacceptable segmentations.

3. In order to find potential confounding factors that may have a significant impact on the quality of the automatically generated contours, we looked at associations between the results and different clinical variables (age, prostate volume, iPSA, ISUP grade, EPE score, and PI-RADS category). If found, this could serve to indicate whether a given prostate is suitable for automatic segmentation. It may also be useful for constructing better segmentation algorithms in the future. This analysis was made with the Kruskal-Wallis tests (for categorical variables), and Spearman correlation (for continuous variables). This analysis was only performed on the Dice and MSD performance metrics.

4. Lastly, we tested whether the increased training size in the 70/30 split significantly improved the performance of the best model compared to the 50/50 split. This can indicate whether datasets of this size are adequate to train a model, or if larger datasets are needed. The performance increase resulting from adding data will diminish at some point, and understanding this interplay will be important for future

studies, especially because good-quality clinical data is hard to collect. This analysis was done with the Kruskal-Wallis test (since samples were of different size). As an extension of this analysis, we also performed a pair-wise Wilcoxon signed rank test on all patients in the intersection of the 70/30 and 50/50 test sets (a total of 20 patients).

No false discovery rate corrections were applied in the above analyses since we prioritized low type 2 errors (to not discard any potential associations).

Model training

The models were trained with a modified pixel-wise top-k cross-entropy loss function with $k = 143,360$ (5% of the pixels in the $320 \times 320 \times 28$ images), which only considers contributions from the k most poorly segmented pixels. The batch size was set to 2 in all cases due to memory constraints. To fully leverage the available data, we heavily augmented the samples with the following augmentation methods (listed in order of application):

- Mixup[156,70] with $\alpha = 0.5$ (that is, randomly sampling the mixing proportion from a beta distribution with $\beta = 0.5$ for both distribution parameters).
- Horizontal flips with 50% probability.
- Uniform random rotation in the $[-\frac{\pi}{4}, \frac{\pi}{4}]$ range about the depth axis.
- Random bilinear resize and translation with a uniformly distributed scale factor in the [0.7, 1.3] range.
- Elastic deformation.

Since the target mask is not binary after applying mixup, we modified the loss function to penalize the distance from the target (as opposed to the distance from the binary label encoding), i.e. to $\mathcal{L} = -\log(1 - |\mathbf{Y} - \hat{\mathbf{Y}}|)$ from $\mathcal{L} = -\sum_i p(y_i)\log(p(\hat{y}_i))$, where $y \in [0, 1]$ are the target labels and $\hat{y} \in (0, 1)$ are the predicted values.[*]

We used the Adam optimizer with default learning parameters ($\beta_1 = 0.9, \beta_2 = 0.999$) to train all our models (apart from the GAN). The learning rate was decreased on loss plateaus following an annealing schedule of 0.001→0.0005→0.0001.

For the GAN, which uses separate optimizers for the generator and the discriminator, we instead set the parameters to $\beta_1 = 0.5$ and $\beta_2 = 0.999$, with learning rate 0.0001 for the generator and 0.001 for the discriminator. In addition, we used the relativistic GAN loss[76], defined on the discriminator outputs $D(x)$ by $\mathcal{L}_D = -\log(\text{sigmoid}(D(x) - D(\hat{x})))$ for the discriminator and $\mathcal{L}_G = -\log(\text{sigmoid}(D(\hat{x}) - D(x)))$ for the generator, where x and $\hat{x}$ denote the real and generated images, respectively. Since each fake image in this case had a corresponding real image, we did not need to sample $(x, \hat{x})$ pairs in order to implement the relativistic loss (this is not the case for GANs in general, since the fake images are often generated from a latent distribution). To stabilize the GAN training, we added a Dice-loss term (see (2.1)) to $\mathcal{L}_G$ with a relative weight of $5 : 1$ in favor of Dice.

All models were implemented with TensorFlow 2.4 in Python 3.7 and trained on an NVIDIA Tesla V100-SXM2 (16 GB).

[*] Without this modification, the loss for intermediate values would not be minimized when $y = \hat{y}$. For example, $-0.5\log(1) < -0.5\log(0.5)$.

2.4.2 Results

Segmentation performance

The segmentation scores (Dice, ARVD, MSD, and HD95) of the different methods are displayed in Figure 2.12. In all cases, EfficientDet3D was the highest-performing model on average (except in terms of ARVD in the 70/30 split, where it came second after the V-net). The total mean ranks across all scores were: 1.25 for the EfficientDet3D, 2.9 for both TL and V-net, 3.5 for GAN, 4.6 for Atlas, and 6.0 for Siemens. The superior performance of the EfficientDet3D model was significant in most cases (see Figure 2.13) with the exception of ARVD, according to which all DL-based models performed similarly. In terms of Dice, the TL model was the only one to perform at a similar level. For MSD, the GAN and the TL approach (in the 50/50 split) achieved similar performance, and for HD95, only the TL approach (in the 70/30 split) was statistically similar to the EfficientDet3D model.

In terms of performance of the most poorly segmented patients (Figure 2.12), the EfficientDet3D model was generally the best, failing to take first place only in terms of MSD and HD95 in the 70/30 split (where it was second to the TL approach), and in terms of ARVD in the 50/50 split (where it ended up in fourth place). Segmentation masks made by the EfficientDet3D model for its worst-case patient in terms of Dice are compared against the ground truth in Figure 2.14. Each performance metric had a different worst-case patient for the EfficientDet3D. The mean ranks in terms of worst-case performance for the remaining four methods were: 2.5 for V-net, 2.6 for TL, 3.8 for GAN, 4.9 for Atlas, and 5.6 for Siemens.

We further note that:

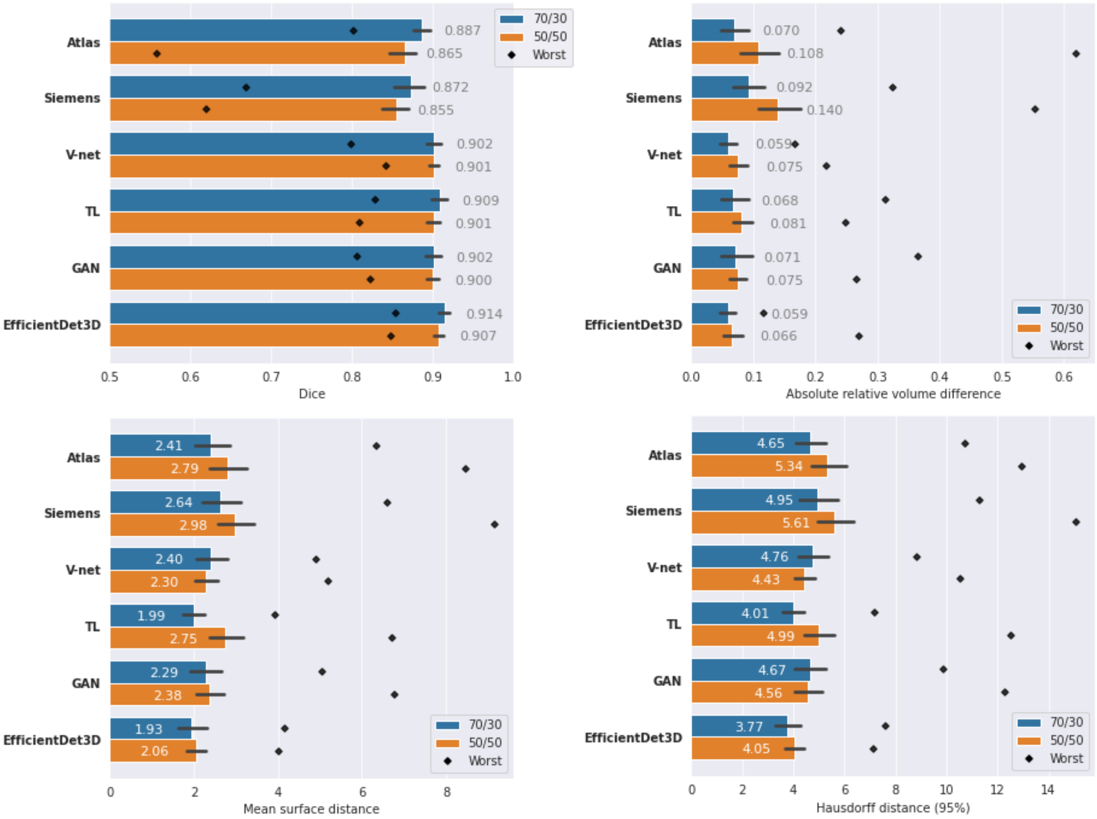


Figure 2.12: Performance of the different segmentation models in terms of Dice, absolute relative volume difference, mean surface distance, and Hausdorff (95 percentile) distance. Numbers represent the mean value, colors represent the dataset split (blue: 70/30, orange: 50/50), black lines represent the standard deviations, and black diamonds represent the result of the worst case. Note that, apart from Dice (top left panel), a lower value means better performance.

- the variance of the models' performance decreased as the mean performance increased

- the mean performances were always higher in the 70/30 split compared to the 50/50 split (with three exceptions), but the differences were not statistically significant

- the DL-based models were less sensitive to performance outliers

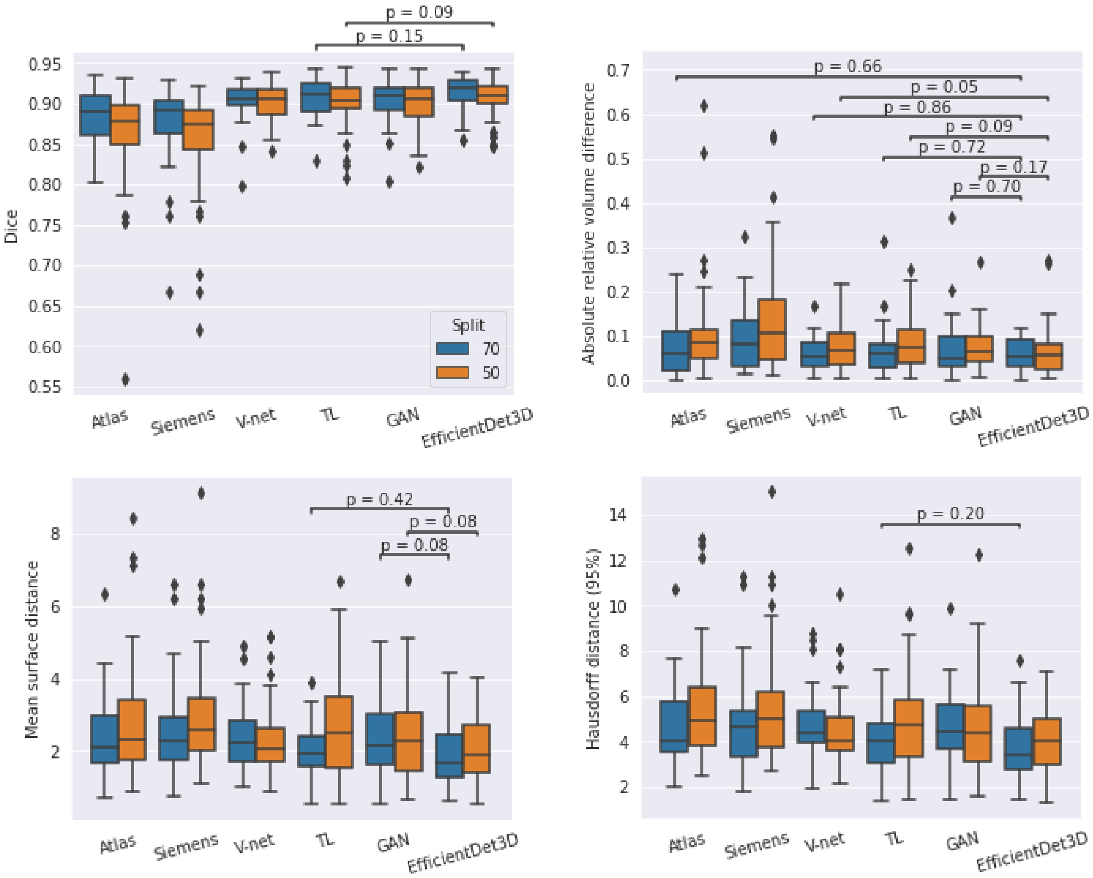


Figure 2.13: Statistical analysis of the segmentation performance resulting from the Wilcoxon signed rank test. Horizontal lines indicate all methods that did not differ significantly from the EfficientDet3D model, which was the best performing model on average. Blue and orange colors represent runs from the 70/30 and 50/50 dataset splits, respectively.

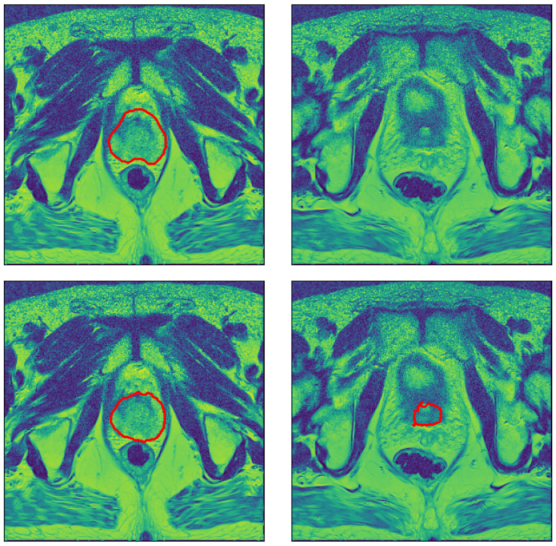


Figure 2.14: Automatic segmentation from the EfficientDet3D model (bottom) compared against the ground-truth segmentation (top) of the worst case patient in terms of the Dice coefficient. The left panels display a slice from the center of the prostate, and the right panels display a slice just outside the base of the prostate, where the model erroneously believes there is prostate tissue.

Association between clinical variables and segmentation results

The full association analysis is presented in Appendix A.1. The only significant relationship was between prostate volume and MSD, and age and MSD, but both relationships were weak (ρ=0.33 and ρ=0.29, respectively). No relationship was consistent between both MSD and Dice.

Effects of dataset size

The comparison between the results from the 70/30 and 50/50 dataset split is presented in Fig. 2.15. Neither test (paired samples Wilcoxon signed rank test and Kruskal-Wallis test) found any significant differences for any of the performance metrics. This analysis was only done for the EfficientDet3D model.

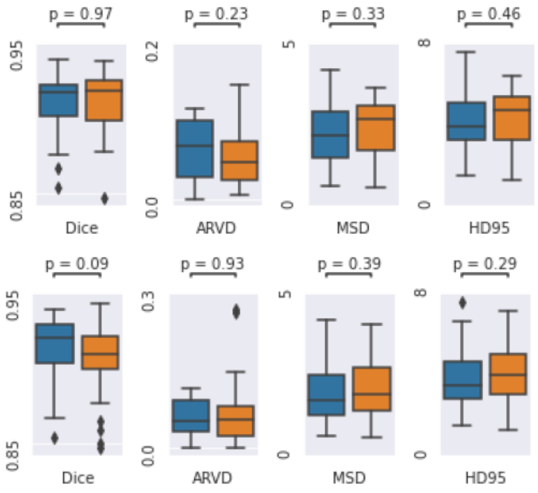


Figure 2.15: Statistical analysis of the differences in segmentation performance between the 70/30 and 50/50 dataset splits of the EfficientDet3D model. Top: Wilcoxon signed rank test on the intersection of the test sets (20 patients). Bottom: Kruskal-Wallis test between the full test sets (30 and 50 patients). Blue and orange colors represent the 70/30 and 50/50 dataset splits, respectively.

2.5 Quality assurance with deep learning

Deploying an automatic segmentation model in practice should require rigorous quality assurance (QA) and continuous monitoring of the model's use and performance, particularly in high-stakes scenarios such as healthcare. Currently, however, tools to assist with QA for such models are not available to AI researchers. In this section, we build a deep learning model that estimates the quality of automatically generated contours and explore some of its clinical implications.

One of the problems with estimating the quality of a segmentation mask is that the output is essentially a collection of predictions—one for each pixel—which invalidates many of the standard certainty approaches for individual predictions. In ordinary classification tasks, the output of the model is typically a vector of probabilities that represents the model's certainty for a given sample. This vector can be used to give users a measure of how confident the model is for a given prediction (although the reliability of this confidence estimation is debated—see the so-called calibration problem). For segmentation models, on the other hand, the output is a pixel-wise collection of probabilities, which makes its direct use ambiguous. For example, it is not obvious whether all pixels should have equal importance, or if perhaps pixels closer to the prostate boundary (where most of the less well-defined pixels are) should be considered more significant. One way to overcome this is to use an ensemble of multiple segmentation models to build an uncertainty prediction model based upon the variance of their predictions, as suggested in[101]. However, model ensembles require drastically more time and resources, particularly in the case of DL. Another approach is to use a Bayesian framework, which inherently models uncertainty[14]. Men et al.[102] tried to solve the problem by studying a binary definition of quality (e.g. errors are present/absent), while other researchers have suggested using rule-based models derived from geometric attributes[5,18], or texture-based[159], volumetric-based[60], and/or shape- and intensity-based[100] features with ML and statistical models. While these methods can improve the quality evaluation of contours, they suffer from the inherent limitations of hand-crafted models, which are often coarse-grained and typically produce inferior results. In this context, DL models, whose superior feature extraction and inference capabilities have been demonstrated in a multitude of other image-analysis domains, may provide invaluable advantages.

Research relating to QA of segmentation models with DL is very limited. Chen et al.[20] conducted a QA study on 680 breast cancer patient CT images based on a ResNet-101. Their approach achieved good performance but used a discrete classification rather than a continuous regression method, which may not be desired in clinical contexts because of its limited flexibility. Two other related approaches are the fields of uncertainty prediction and out-of-bounds detection. In uncertainty prediction problems, the task is to estimate the uncertainty of the network given an input-prediction pair (this is typically studied in regression problems where network outputs are not represented as certainties). Traditional approaches in this field model the uncertainty with a Bayesian framework and typically require either explicit models of the ground truth distribution[63,40,79] or joint training with an uncertainty network[51]. While joint training can be an effective and well-versed approach, it stands to reason that it can not be implemented after training. Instead, out-of-bounds detection simply tries to detect samples that differ greatly from the training distribution. These detectors often rely on artificially created out-of-distribution samples (possibly with a generator neural network[89])[92], which is a difficult problem in itself that introduces another dimension of bias. Due to the scarcity of research directly related to QA, it can be useful to draw inspiration from other domains that predict (continuous) outcomes directly from images. One example is age estimation, where the task is to predict a person's age from pictures of their face (see e.g. Cao 2020[16], Gao 2020[42], Liu 2020[94], or Berg 2021[12]). Notably, all state-of-the-art models in this field are DL architectures. However, a key thing that distinguishes segmentation QA from age estimation is the presumed in-sample interaction between images and contours: it is impossible to tell the quality of a segmentation just by purely looking at the segmentation (or image).

There are three main ways to predict the performance of automatically generated contours: regressing the performance metric directly, predicting some discrete or qualitative measure of performance (e.g. good/moderate/bad), or predicting the ranking of samples ordered by quality (i.e. ordinal regression/classification). The second method is useful when precise ground truth segmentations are not available, or when time constraints limit the annotation quality of the training data since clinicians can allocate samples to qualitative categories faster and more easily than they can produce reliable ground truth segmentations. The third method can be beneficial when only the ordering of samples is important, but the downside is that single-sample inference is not straightforward, particularly for out-of-distribution samples. We opted for the first method, as it is good when many training samples are available, preferably over a wide distribution of ground truth segmentation scores. It allows for the use of distance-based loss metrics, which penalize poor predictions more than good ones.

In this section, we will:

- Build a convolutional DL model to predict the quality of automatically generated contours. This model can be used in QA to give automatic segmentation models a greater sense of transparency.

- Compare the model to a naïve baseline that predicts the segmentation performance from patient characteristics only. If poor performance can be estimated from clinical variables alone, clinicians may be able to leverage this information to exclude subsets of patients on which automatic segmentation methods produce poor results.

- Investigate when the model is able to correctly handle likely failure cases with noisy, empty, or misaligned predictions. Identifying failure cases is a crucial step in QA and analyzing this behavior may provide insights into potential model biases and model behaviors.

2.5.1 Methods

Predicting contour quality

We framed the problem of quality assurance as a regression problem, where the input to the model is an image-contour pair and the output is a measure of quality (see Figure 2.16 for an illustrative overview). The specific quality metric we chose was the Dice coefficient (2.1) because it is mathematically well-defined, easily interpreted, bounded, and widely used within the medical imaging community.

To measure the performance of the quality prediction models we used mean absolute error (MAE) between the predicted Dice and the target Dice values, as well as the Spearman rank correlation between them. The Spearman correlation measures how correct the order of an ordered set is, and ranges from 1 (all samples are placed in correct order) to -1 (all samples are placed in the opposite order). A random placement has an expected rank correlation of zero. As such, it gives an intuitive understanding of how well the algorithm can tell good contours from bad ones. Here, the use of rank correlation will be implied whenever correlations are mentioned.

Baseline quality prediction model

The baseline model tries to predict how well an arbitrary segmentation algorithm would perform on a patient given only the clinical variables for that patient. The rationale for this is to examine whether any clinical variables are predictive of how hard it is to segment a given prostate. Because this model makes no use of images, it cannot distinguish between different segmentations of the same patient, but it can still be useful as an analytical tool

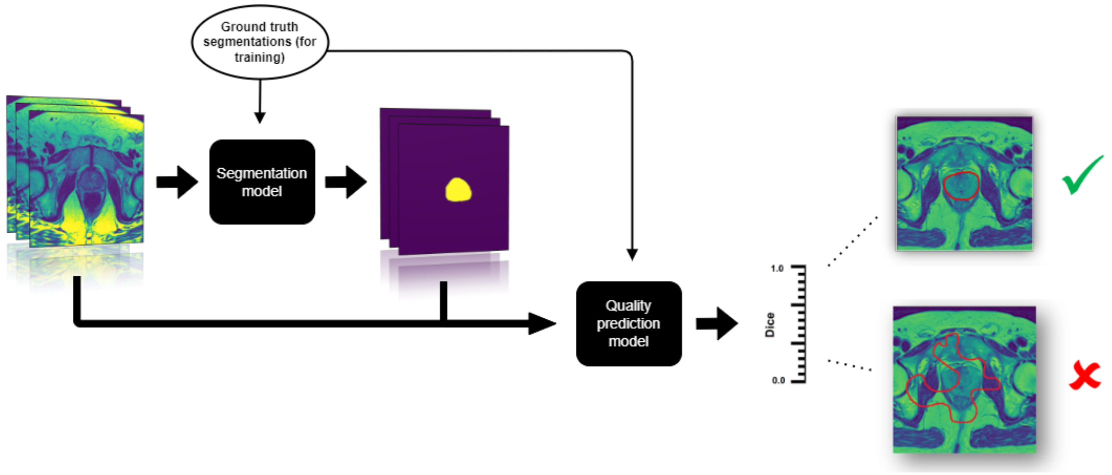


Figure 2.16: Overview of the problem of predicting the quality of organ segmentations. First, a segmentation model takes images as input and produces segmentation maps. Then, our quality prediction model takes both the images and the segmentation maps as input and produces an estimate of the quality of the segmentation—in this case the Dice similarity coefficient. Good contours have a high Dice value and poor contours have a low Dice value. Note that we need ground truth segmentations in order to calculate the true Dice value and train the quality prediction model with supervised learning. In this study, we used 60 ground truth segmentations and 80 automatically generated masks along with heavy data augmentation to train the quality prediction model.

and a benchmark. As the baseline model architecture, we chose a gradient boosted decision tree model implemented in CatBoost[117] v.1.0.3 with Python 3.7. For the 20 images with two different segmentations, we used the mean value of the Dice coefficients as the target value. The following clinical characteristics were available for each patient for use in the this model: age, prostate volume, ISUP grade, PI-RADS score, iPSA, and risk class

To train this model, we first perform a 64-step parameter search with the Optuna[3] Python package with default settings to find suitable parameters. The search space is displayed in Table 2.2. Each parameter set was evaluated by its mean absolute error after eight repeated random 5-fold cross-validations. The best model was then further evaluated with 64 repeated random 5-fold cross-validations.

In order to gauge the usefulness of the baseline model, we compared its performance

Table 2.2: Parameter space searched by the Optuna parameter search for the baseline CatBoost model.

Parameter	Values
n_estimators	{1, 256}
max_depth	{1, 6}
l2_leaf_reg	$[10^{-3}, 10]^*$
random_strength	[0.1, 3]

*log-uniform prior
[]: continuous interval
{}: integer interval

against a naïve baseline that predicts the mean Dice value for all samples.

Quality prediction network

Since this network was trained to predict contour quality, it needs not only an image and a ground truth segmentation but also a generated segmentation mask for which to predict the quality (alternatively, it needs an image, a contour, and a target Dice value). As such, it was trained on the test sets from the previously trained segmentation models for a total of 50+30=80 training samples. This means that the patients in the overlap of the two test sets (20 patients, to be precise) are included twice and considered independent samples (i.e. each of these 20 patients has one image-contour sample from the 70/30 test set and one image-contour sample from the 50/50 test set, but the images in both samples are identical).

The DL model we trained to predict segmentation quality was a modified EfficientDet[138] architecture (see Figure 2.17). Our modifications included adaptation to 3D convolutions as well as an expansion-factor reduction (from 6 to 2) and a custom regression head. The regression head consisted of serially connected fast normalized fusion nodes (see (2.6)) followed by batch normalization, PReLU, a single-channel convolution, and a final sigmoid activation function. The EfficientNetB0 backbone used the default filter parameters of 32, 16, 24, 40, 80, 112, and 192 channels for the P1 to P7 levels, respectively. Our BiFPN blocks were repeated three times and used 64 filters each.

To reduce the memory consumption of the model, the images were center-cropped from 320×320×28 to 160×160×28 voxels. The images were also normalized by linearly mapping the 0th and 99th percentiles to the [0, 255] range, after which the 100th percentile values were appended. The MRI images and segmentation maps were concatenated on the channel dimension to form 4D tensors of shape 160×160×28×2 for each sample. These 4D tensors were used as the input to the model.

The network was trained for 200 epochs with MSE loss and batch size 2. We used the Adam optimizer with a learning rate of 0.002, which was reduced to 0.0002 after 120 epochs. Validation of the model was done with a random 5-fold cross-validation.

To give the network the ability to interpolate outside the narrow range of target Dice values typical of prostate segmentation, we used an elaborate data augmentation scheme to generate novel contours. At each epoch, one of the two samples had its corresponding contour switched with another contour randomly chosen from the training set such that each batch consisted of one real and one "fake" image-contour pair. The fake contour was then scaled by a random factor in [0.55, 1.8]. After this procedure, we also applied standard data augmentation two both samples (in order): horizontal flips, uniform in-plane rotation (in $\pm\frac{\pi}{12}$) uniform 2D x and y translation (in ±10%), uniform zoom (in ±10%), and elastic deformation. This procedure also eliminates bias that could be introduced by only using

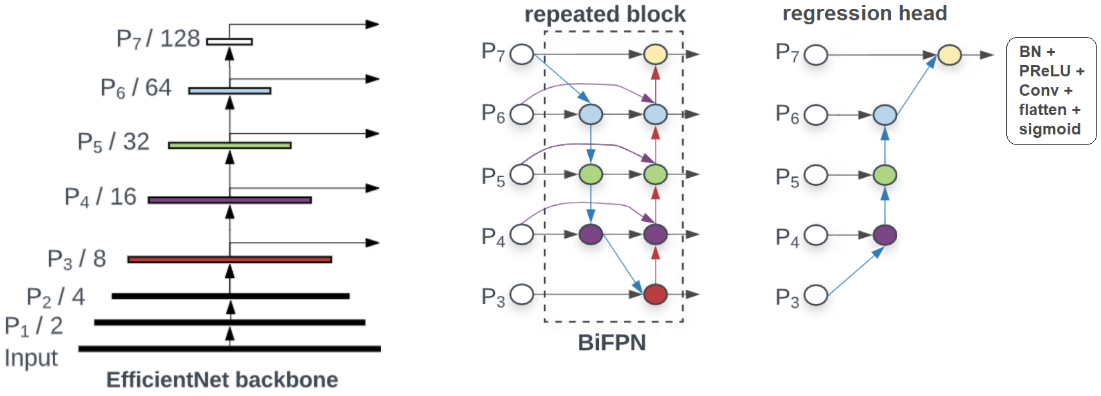


Figure 2.17: Quality assurance network architecture. An EffcientNetB0 backbone is connected to three repeated bidirectional feature pyramid blocks (BiFPNs). The regression head consists of serially connected fast normalized fusion nodes and finally BatchNorm (BN), PReLU, a single-channel convolution, and a sigmoid activation. Numbers indicate the resolution at each level relative to the input.

contours from a single segmentation model.

Failure case studies

We evaluated how well the model predicts the quality of different variations of failed contours, for which the predicted Dice score ought to be low. The following failure modes were investigated (see Figure 2.21 for illustrations):

1. empty contours (every pixel in the array is zero—no prostate tissue has been identified),

2. uniform binary noise (each pixel in the array is randomly assigned a value of zero or one),

3. filled matrix of ones (every pixel in the array is one—the whole image has been identified as prostate tissue),

4. shifted ground truth masks (the ground truth segmentation is randomly shifted uniformly by ±50% in the *x*- and *y*-direction).

These cases were constructed from the 16 patients in the test set at each validation fold, such that each failure case generated 80 independent samples in the course of the cross-validation procedure.

In addition, we evaluated the predictions on the 16 unseen ground truth segmentations at each validation fold (for a total of 80 ground truth images). This allowed us to test the model performance on the opposite end of the domain, where all target values are 1.

We also defined a global accuracy score to indicate how well the model performed across all test samples (80 from the standard test set, 320 failure case samples, and 80 ground truth samples). This is useful because the MAE is not always indicative of how helpful the model's predictions are. For example, if there is a segmentation with a true Dice value of 0.0, and the model predicts a Dice value of 0.5, the contour would still be flagged as "poor quality", because both 0.0 and 0.5 Dice are considered bad. This means that the prediction is qualitatively correct, even though the MAE of 0.5 is very large. For a predicted Dice value $\hat{y}$ and target Dice value y, we defined a failed prediction as either:

- $(\hat{y} < 0.75) \wedge (y > 0.8)$, i.e. a predicted Dice value of less than 0.75 when the target Dice value is larger than 0.8, or

- $(\hat{y} > 0.8) \wedge (y < 0.75)$, i.e. a predicted Dice value larger than 0.8 when the target Dice value is less than 0.75.

Table 2.3: Average mean absolute error (MAE) and rank correlation of the baseline CatBoost and naïve models, which only utilize clinical variables to predict segmentation quality. The naïve method predicts the mean target value for all samples. Parentheses indicate standard deviation. The values are aggregated from 64 repeated 5-fold cross-validations.

	MAE	Corr
CatBoost	0.016 $(\pm 3 \cdot 10^{-4})$	-0.16 (± 0.02)
Naïve	0.016 (± 0)	n/a

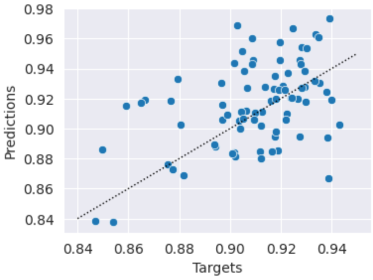


Figure 2.18: Predicted vs. target Dice values of the deep network shown in Figure 2.17. The dotted line indicates perfect $x = y$ predictions

2.5.2 Results

The performance of the CatBoost and naïve baseline models are summarized in Table 2.3. The MAEs were 0.016 for both models, and the prediction-target correlation for the CatBoost model was -0.155 ($p = 0.55$). A deeper analysis of the relationship between the clinical variables and segmentation performance is presented in Appendix A.1.

The mean absolute error of the Dice value predictions was 0.020 ±0.026 (2.2% mean absolute percentage error), and their correlation with the target values was 0.423 (Figures 2.18 and 2.19). The maximum absolute error was 0.066 (equivalent to a 7.3% deviation from the target). The time required to generate predictions was 0.02s per patient on an RTX 3090 GPU. Characteristic training curves of the network are shown in Figure 2.20.

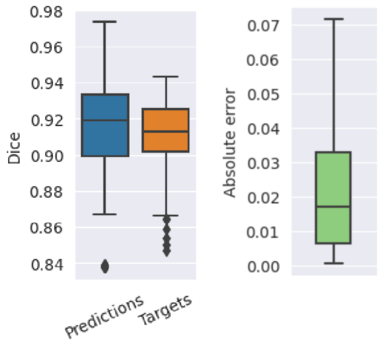


Figure 2.19: Predicted Dice values, targets, and the respective absolute error of the quality prediction deep learning network. The mean absolute error is 0.02 and the correlation between the predicted and target values is 0.42

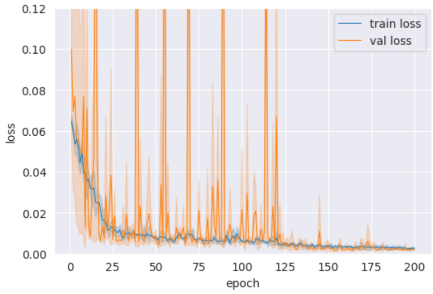
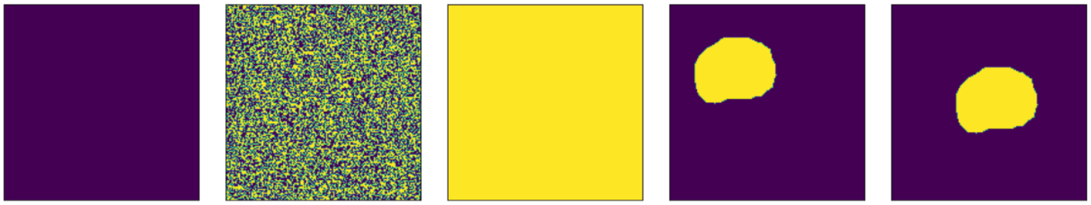


Figure 2.20: Characteristic training curves of the deep quality prediction network. The plot is an aggregate of all five validation folds. The validation loss often spikes in early training, which then disappears after the learning rate reduction at 120 epochs



	Empty	Binary noise	Ones	Shifted GT	GT
MAE	0.32 (±0.074)	0.13 (±0.055)	0.18 (±0.16)	0.23 (±0.12)	0.10 (±0.037)
Corr	n/a	0.21	0.17	0.52	n/a

Figure 2.21: Performance of the quality prediction model (MAE and rank correlation of predicted Dice scores, including the 95% confidence intervals) on different cases of failed segmentations: completely empty contours, pure noise, matrices full of ones, and shifted ground truths (GTs). The performance on real GT segmentation maps is also shown. All results are aggregated over 5 different validation splits for a total of 80 samples each

The results of the predictions on the constructed failure cases are shown in Figure 2.21 together with an example segmentation from each type of failure. The least successful cases were the empty contours (0.317 MAE), followed by the shifted GT segmentations (0.233 MAE), the all-ones segmentations (0.182 MAE), and the binary noise (0.126 MAE). The shifted GT cases, which had the second-worst MAE, had the best correlation with the target values: 0.522. The empty and GT contours have undefined correlations since their target values are all zeros and all ones, respectively.

In terms of overall accuracy, only two out of the 480 cases were misclassified, amounting to a 99.6% accuracy. The first case was a segmentation full of ones with a predicted Dice value of 0.81 and target Dice value of 0.11, and the second case was a shifted GT mask with a predicted Dice value of 0.82 and target Dice value of 0.72.

2.6 Discussion

Overview

The automatic segmentation methods evaluated here reached Dice scores of up to 0.914, which is on par with, if not better than, that for human agents, for which studies have reported Dice coefficients of 0.90[93], 0.83[150], 0.859[11], and 0.82[126]. Even our worst-performing DL-based model achieved an average Dice score of 0.900, which indicates that the technique is relatively robust regardless of the underlying architecture. On the other hand, the performance of the medical software algorithms yielded significantly inferior performance with notable obvious mistakes. An exceptionally thorough recent review[80] covering 100 different papers on prostate cancer segmentation confirms that our performance is similar to that of the best performing algorithms on other similar-sized datasets (although care should be taken when comparing studies involving different data sets). This review further demonstrates the width of the different models that can achieve these performances.

Even with a dataset of just 50 patients, the Dice performance of the best and worst DL models were 0.907 and 0.900, respectively. This can be partially attributed to the mixup augmentation, which improved the Dice performance by 0.016 or 3.15% ($p = 0.005$). This indicates that segmentation models might be trainable up to a clinically acceptable degree with very little data, contrary to the widespread assumption that DL models need enormous datasets. It should be noted, however, that the average test score is not the quintessential metric of model performance. For a model to be useful in clinics, it is equally important to not instill false confidence when the segmentation is bad, which might happen if the model is given an atypical sample (e.g. outside of the training distribution). This is why quality assurance of deployed DL models is essential.

The DL quality prediction model for QA accurately predicted the Dice scores of automatically generated prostate contours with a MAE of 0.020. This amounts to a mean deviation from the true Dice values of only 2.2%. In particular, none of the errors were larger than 0.066 (7.3% deviation from the target), which indicates a high degree of robustness and reliability. The moderate correlation of 0.42 between the predicted and target values suggests that the model is also able to correctly infer quality differences (i.e. which ones are better/worse) between contours, even when the differences are minor. These results provide a solid framework for a practically useful segmentation pipeline, but more conclusive results with bigger data need to be produced before automatic segmentation can become a reality in clinics.

Ablation studies and unsuccessful experiments

Due to the vast landscape of possible design choices when building DL models, it is practically impossible to explore most of the design space, particularly in low-recourse scenarios. Our general approach in this project was to favor simplicity and ease of implementation over minor performance gains while taking cues from top-performing submissions in segmentation challenges (e.g. the PROMISE12 prostate segmentation challenge[93]) and other computer-vision areas (e.g. classification and object detection). This is not a substitution for rigorous experiments, but it provided direction when resources were scarce. Several different architectures and parameters were explored initially, but were abandoned when their results were not promising or interesting.

For the segmentation architecture, this included residual refinement blocks, group dilated convolutions, decomposed convolutions, and pyramid attention. We also experimented with mixed precision training and alternative loss functions such as intersection-over-union loss, boundary distance loss, and hybrid losses as well as various learning rate schedules. For preprocessing and augmentation, Gaussian smoothing, noise addition (uniform, Gaussian, salt and pepper), label smoothing, cropping, and bias-field addition were experimented with but did not consistently influence the results in a meaningful way. Most of the parameter choices were selected similarly, but special attention was devoted to the number of convolutional filters and the dropout rates.

Overall, the segmentation performances of the less performant DL-based models (U-net, TL, and GAN) were similar (total mean ranks of 2.69, 2.75, and 3.63, respectively). Since GANs are notoriously hard to train and stabilize, it is likely that the GAN model could be improved further by a more careful and thorough exploration. A similar argument can be made regarding the TL model's performance: it is likely that it could be improved by a more careful investigation into the possible backbones and structures (e.g. PSPNet or Linknet). We briefly experimented with ResNet and InceptionV3 as backbones but concluded that EfficientNet was the most appropriate choice. In this project, we tried to commit roughly equal amounts of effort to all models such that the results would reflect the underlying differences and not just tuning variability.

One promising architecture we tested for the QA network used a confidence branch, which is an extension that can be made to arbitrary networks[34]. The confidence branch is trained to output an estimate of how confident the network is in its predictions. This can be achieved by letting the network "peak" at the correct answers (with a penalty) during training in order to output more correct answers. The intuition behind this is that peak-ing is only profitable for inherently uncertain predictions. Thus, the network's desire to peak can be seen as a proxy for uncertainty. We did not manage to get this branch to output useful values—the confidence always converged to either 0 or 1 for all samples, even after introducing a budget parameter.

Another DL network we experimented with for the QA model used a regression head added directly to the EfficientNet backbone, which requires much less memory and training time. This worked well in terms of training MSE but was incredibly noisy. A similar scenario occurred when we trained the final network architecture with fewer BiFPN layers and/or filters.

Generalization and mixup

The generalization improvement from mixup may be dependent on the network architecture, but there is reason to believe that networks of the U-net family behave similarly since they all follow the same fundamental principle of systematic feature extraction with convolutions. The network that was used to study mixup was designed to be as simple and general as possible while maintaining a respectable level of performance, such that the results could be generalized to more specialized network topologies as well. Therefore, it did not incorporate specialized structures like attention blocks, residual refinement blocks, squeeze and excitation blocks, depth-wise convolutions, etc., which are utilized in many state-of-the-art architectures today. The generalization improvement attributed to mixup for classification has been demonstrated to be greater for larger networks (i.e. networks with more parameters)[156] and it is plausible that this holds even for segmentation. As a result, the segmentation performance gained from mixup could be greater for more complex anatomical structures requiring larger networks, such as bone or blood vessels, than the relatively simple prostate geometry.

Since the mixup procedure renders the target mask non-binary, caution needs to be taken when designing the desired loss function. For example, the standard definition of the Dice coefficient in (2.1) may be extended to continuous cases by simply defining the intersection as the product between $\mathbf{X}$ and $\mathbf{Y}$, but such an implementation is not always maximized when $\mathbf{X} = \mathbf{Y}$ because $\sum_i x_i y_i \leq \sum_i x_i$ for $x_i, y_i \in [0, 1]$. Here we used top-k cross entropy because of the previously demonstrated success of standard cross entropy in combination with mixup for classification[156], and because it outperformed Dice in preliminary testing. Further exploration in the interest of optimizing the loss function for mixup segmentation is warranted but is outside the scope of this thesis. Currently, there appear to be no obvious reasons why the generalization improvement from mixup should be exclusive to the cross entropy loss.

The value of mixup may extend even beyond performance gains: it has been suggested that mixup training significantly improves calibration and uncertainty estimation of deep convolutional networks[139]. From our results, it is also apparent that mixup reduces the prevalence of outliers (see Figure 2.5). This could be an additional incentive to use mixup for medical image segmentation, even if the performance itself is not improved, since uncertainty in patient decisions is much more detrimental.

Additional theoretical explorations as to why and how the mixup procedure leads to better generalization performance are also warranted since this is still not well-understood[156,91,65]. Two principally distinct intuitive explanations have been proposed. First, it can be viewed as regularizing the network by simply increasing the sample size. Secondly, it can be viewed as simultaneously learning multiple samples such that the network learns to better distinguish between their corresponding labels. The original mixup paper suggested that the between-class samples simply help the network interpolate smoothly instead of using hard decision boundaries. It has also been suggested that mixing promotes more robust detection of low-level features such as lines or edges. If this is indeed the case, the strategy ought to be less effective for transfer-learned networks, where the low-level feature weights are typically frozen.

Commentary on the segmentation results

From the analysis of the relationship between clinical variables and segmentation performance, we conclude that exceptionally large prostate volume and old age may lead to poor segmentation performance, although this result was only significant for the MSD metric. In principle, a positive linear correlation between MSD and Volume also implies a good performance (i.e. low MSD) for patients with small volumes, although exceptionally small prostates are likely to cause poor performance due to other effects. Since there was only one observed significant relationship out of four performance metrics, it is likely that clinical variables are not strong predictors of segmentation performance.

The comparison between the different dataset splits (70/30 and 50/50) indicates no significant performance increase when expanding the training set by 40% from 50 to 70 images. However, in 13 out of the 16 combinations of DL-based models and metrics, the performance was better in the 70/30 split on average. It is likely that this dataset was simply too small to reveal the difference with significance. Increasing the dataset's size does not only improve the average performance, but also the models' resistance to outliers; the scores of the 50/50 split generally exhibit higher variances and more severe outliers.

An interesting observation about the results is that the variance of the performance is very low: EfficientDet's standard deviation is just 2.1% from the mean value (0.04% for the variance). This is contrary to what one would expect from such a small dataset. The worst Dice coefficient and MSD in the best performing model were 0.854/0.847 and 4.16/4.02 (70-30 split/50-50 split), respectively. The scores then increase rapidly; the same values for the 5th worst patients are 0.897/0.876 and 3.25/3.25. This could indicate that prostate segmentation models need much less data than those intended for other regions and applications. A possible direction to improve the worst-case performance further is to weigh these samples more heavily during training. We briefly experimented with various weighted loss functions but found that this led to worse average performance.

Commentary on the quality assurance results

The performance in the failure cases might seem alarming when looking at their MAE values, which range from 0.126 MAE on binary noise to 0.317 on empty contours. However, this would likely not be a major problem in practice since the overall accuracy of the model was 99.6%. For example, a large MAE does not necessarily indicate a failure of the model when both the target and predicted Dice values are low. Our QA model was not trained on any contours with target Dice values of 0.0, and as such had no way to interpolate to this regime. Anticipating failure cases and including such cases in the training set is one way to boost the model's reliability. Furthermore, a well-built segmentation model being deployed in practice ought to not output obviously poor segmentations (assuming no model/data drift), and such failures are easy to spot by simple inspection without the need for an external quality assurance model. If drift is an issue, one can use a surrogate model in addition to the QA model to detect out-of-distribution cases.

The naïve model that predicts the mean of the target Dice values for every patient achieved a MAE of 0.016, which is lower than the DL model's MAE of 0.02. However, the naïve model would not be able to identify failure cases and qualitative differences between contours, since all its predictions are identical. Similarly, the baseline CatBoost model, which only used clinical characteristics to predict the quality of automatically generated contours, also had a MAE of 0.016. The low variance along with the negative correlation of the CatBoost predictions (Figure A.3 and Table 2.3) suggests that this model has no merit over the naïve model, effectively rendering it useless in practice. This indicates that the performance of automatic segmentation models cannot be inferred from clinical characteristics alone. This is not too surprising given that the model is not able to distinguish different segmentations on the same patient.

An obvious question to raise is; if we need a deep network to safeguard the performance of the segmentation network, should we then not need a deep network to safeguard the performance of the safeguard network? The predicament is that, if the performance of the first network could be guaranteed, we wouldn't need a safeguard network in the first place, and if not, we would potentially need an infinite chain of networks. It is likely, however, that the utility of such networks diminishes the further down the chain you go because the error is necessarily reduced by a nonzero amount each step. An analogy can be drawn to gradient-boosted machines, which are chains of prediction models trained on a propagated error signal. These models are usually trained with decision trees because decision trees are extremely fast to train. On the other hand, for DL models in computer vision where training times often exceed hours or even days, it should be clear that using more than a few chained safeguard networks is practically infeasible.

While similar studies in the litterature have focused on either binary error detection or discretized ordinal regression, our model performs continuous regression. In general, this should be preferred, since it is more general and often enables better performance. This approach also enables a dynamic definition of error detection that can be changed on-the-fly, which may be valuable in a medical context. The only other study we found that regresses Dice scores directly achieved a MAE of 0.06[21] on breast cancer segmentation, which is three times higher than our MAE of 0.020.

One interesting thing to note is that, if the Dice prediction network is accurate enough, the original segmentation network could utilize the predicted Dice values to inform its own gradients, for instance by using the Dice prediction network as a discriminator in a GAN training scheme. This could improve the segmentation performance further. While this is a potential direction for future research, it was out of scope for this study.

It can be discussed whether it is wise to use an AI algorithm to assess the output of another AI algorithm in clinical practice, and when human supervision should be requested. At the very least, it seems natural to demand human supervision in early applications of AI in healthcare. One benefit of QA models like ours is that they can easily be deployed alongside human professionals in order to alleviate the workload and improve their judgment.

Dataset

The images used in this study all came from a homogeneous population, and all patients had relatively severe cancer (Gleason score 6 or above). While it is crucial to know that the methods work well for these cases, it would be helpful to also add data from healthy patients so that the performance on their images also can be ensured. If the intention of the algorithms is to exclusively handle patients within this population, a relatively small data set (albeit ideally larger than the one included here) may suffice. But as soon as results on external samples are of interest (other parameter settings of the MRI scanner, different quality of scans, and so on), it will be critical to add additional data to represent these. For the same reason, it is not surprising that the Siemens' segmentation model in the Syngo.via software had the worst performance by far (it is trained on a different training set). It should also be noted that, although the experiments here focused exclusively on T2-weighted MRI images, it is likely that the methods transfer to other modalities such as CT and PET images as well since they are all represented in the same way in the software. Moreover, as DWI images of the patients are usually acquired as part of the prostate MRI examination, translating these models to and using them with the DWI data is important.

While this project demonstrates promising results in the various tasks, it is important to note that external and rigorous validation is needed before the methods can be reliably applied in practice. With the limited dataset of 100 patients, the results herein can only be considered as a proof of concept. Ideally, a test set of unseen data should be used to assess real out-of-sample performance and reproducibility, but we opted to not use a held-out set in order to give the training and validation sets more coverage. Data sets that are too small risk being overly dominated by random effects, which can compromise the learned representations and the performance estimation. At present, there is no universally accepted solution to these tradeoffs, especially in small data scenarios.

3

Image normalization & radiomic feature variability

3.1 Background

A peculiar property of MRI images is that their intensity values are not measured on an absolute scale, meaning that the values have no consistent physical interpretation. As a consequence, one can not straightforwardly compare intensities between different MRI scans—it is for instance not possible to say that the intensities corresponding to fat tissue in image A should also correspond to fat in image B. The reason for this behavior lies deep within the physics of the machine. While the signal in a regular camera is generated by detectors picking up photons, the signal in MR images is, very roughly speaking, generated by tiny frequency variations in magnetic fields which, in turn, are generated by the relaxation of hydrogen atoms. This arbitrary nature of the intensities introduces additional uncertainty in the analysis of the images, which further necessitates the need for normalization/standardization techniques.

The goal of normalization is to standardize the intensity distributions both within and across patients. Formally, normalization should: 1) make intensities have similar distributions for the same tissue types within and across patients and 2) make intensities have a common interpretation across locations within the same tissue type, according to the seven principles of image normalization (SPIN)[128], which was proposed as an impetus to standardize research in quantitative imaging. This not only helps to alleviate the problems of MRI mentioned above but also problems resulting from variations of scanner parameters and vendors. The condition, of course, is that no information should be lost (SPIN principle three)—it is not enough to set all muscle intensities to 50, for example. There may also be variations that are worth preserving. For instance, tumors may differ physiologically, in which case the intensity difference could reflect a real medically relevant phenomenon. The remaining principles say that normalization should be replicable, minimally sensitive to artifacts, not be influenced by abnormality or heterogeneity, and preserve the rank of intensities (these conditions, as we will see later, are generally met by the normalization methods considered in this chapter).

An intuitive solution to the problem of intensity standardization might be to adjust contrast, brightness, or the intensity range, but in practice, this does not actually lead to a tissue-specific association of the intensities[109,97]. Similarly, normalizing the intensities to zero mean and unit variance is not enough since the shape of the distributions can differ between patients (consider for example a bimodal distribution; the transformation will look qualitatively different depending on which of the two peaks is dominating). To combat this, a method called histogram normalization was developed by Nyúl and Udupa in 1999 and 2000[109,110], and various alternatives have been suggested since, especially for brain scans.

The implementation and significance of image normalization vary between the different body regions. In lung imaging, where CT images are the primary source of information, normalization is typically not a significant concern since CT image distributions are much more consistent. In the brain, specific methods have been developed that exploit the smoothness of different types of tissue (it is very easy to tell apart white matter, grey matter, and spinal fluid, for instance), such as g-scale and g_b-scale normalization[98]. For the prostate, on the other hand, normalization has been somewhat disregarded, as is evident from observing the differences within the literature (see e.g. Schwier 2019[123] or Isaksson 2020[68]). At present, it is not well known what normalization methods work well or if the methods developed for brain MRI can be successfully applied to the prostate.

In this chapter, we will investigate how normalization affects the intensity distributions and the calculations of radiomic features, and discuss how this might affect downstream analyses. To do this, we will compare four different normalization techniques:

1. Method I: histogram normalization
2. Method II: g-scale normalization
3. Method III: g_b-scale normalization
4. Method IV: and a custom normalization technique based on using healthy prostate tissue as a reference.

The work presented in this chapter is largely based on Isaksson 2020[68].

3.2 Methods

3.2.1 Histogram normalization

The idea of the histogram normalization method is to match quantile landmarks of the intensity histograms with each other via a nonlinear transformation. As a result, the q^{th} quantile of every image will coincide on the same intensity value (for a specific selection of q-values). The algorithm can be divided into four consecutive steps:

1. Optional: identify the foreground pixels in the image, e.g. by considering all pixels whose intensity is higher than the mean value. The purpose of this step is to isolate pixels corresponding to tissue. The mean heuristic works well because pixels in the background (e.g. air outside the patient) have intensities very close to zero.
2. Define a set of landmark quantile points Q and a *standard scale*. The standard scale will determine the intensity range onto which all intensities will be mapped, and the quantiles in Q will determine what quantile values the mapping will use. Let i_k be the intensity value corresponding to q_k for every $q_k \in Q$ in a given image.

3. Create a benchmark histogram:

    - Let p_1 and p_2 be the intensity values of a given pair of lower and upper percentiles of the image intensities and let s_1 and s_2 be the minimum and maximum intensities on the standard scale.

    - In this process, every intensity x within the interval $[p_1, p_2]$ is linearly mapped onto $x' \in [s_1, s_2]$. The functional form of this first mapping is

    $$x' = s_1 + \frac{x - p_1}{p_2 - p_1}(s_2 - s_1). \tag{3.1}$$

    - In the resulting image, calculate the new intensity values i_k of all landmark points $q_k \in Q$

    - Define i'_k as the mean of all i_k-values across images after the transformation.

    - Define the benchmark histogram as $\mathcal{H} = [s_1, i'_1, ..., i'_n, s_2]$

4. In the normalization process, the intensity values of each image are mapped onto the standard scale by utilizing n separate linear maps—one between each interval in $\mathcal{H}$. For a single landmark point, there are two mappings: one from $x \in [p_1, i_1]$ to $x' \in [s_1, i'_1]$ and one from $x \in (i_1, p_2]$ to $x' \in (i'_1, s_2]$. The explicit mapping is

$$x' = \begin{cases} i'_k + \frac{s_1 - i'_k}{p_{1k} - i_k}(x - i_k), & \text{if } x \leq i_k \\ i'_k + \frac{s_2 - i'_k}{p_{2k} - i_k}(x - i_k), & \text{if } i_k < x \end{cases} \tag{3.2}$$

The function is illustrated in Figure 3.1.

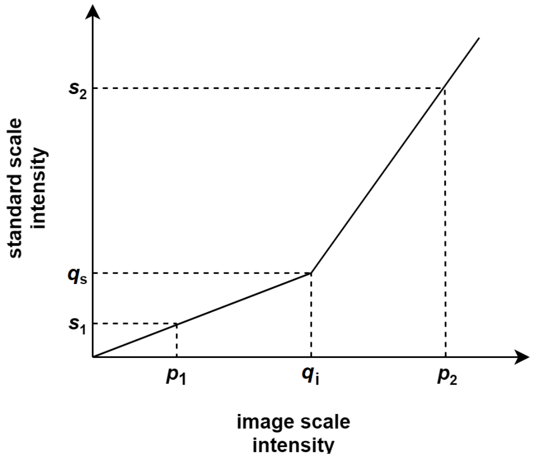


Figure 3.1: Illustration of the piece-wise linear mapping used in the different normalization methods. The landmark intensities (q_i) for the different images on the original image scale are all mapped to the same value, q_i', on the standard scale. Different linear functions map the intensities less than the landmark value and greater than the landmark value (see (3.2)).

This map ensures that the different quantile landmarks $q_k \in Q$ are mapped onto the same value (i_k') in the standard scale regardless of the initial value (i_k). Furthermore, the ranks are preserved since the function is monotonic. The result is new images in which the variation of intensities of the different tissue types are reduced across patients[43] (see Figure 3.2). Since every q_k-quantile in the different images corresponds to the same intensity after the transformation, it is generally of interest to have these correspond to something physically meaningful, like a specific tissue type.

Even though it is possible to use many q-values to match multiple landmark intensities, the current standard, and what we will do in this study, is to only use one point with the median as the landmark. As for the other parameters, p_1 and p_2 are normally set to the 2nd and 98th intensity percentiles to screen off outlier values, but the choice should take the character of the images in mind. In our experiments, where the majority of intensities are very close to zero, we chose the 0th and 99th percentiles.

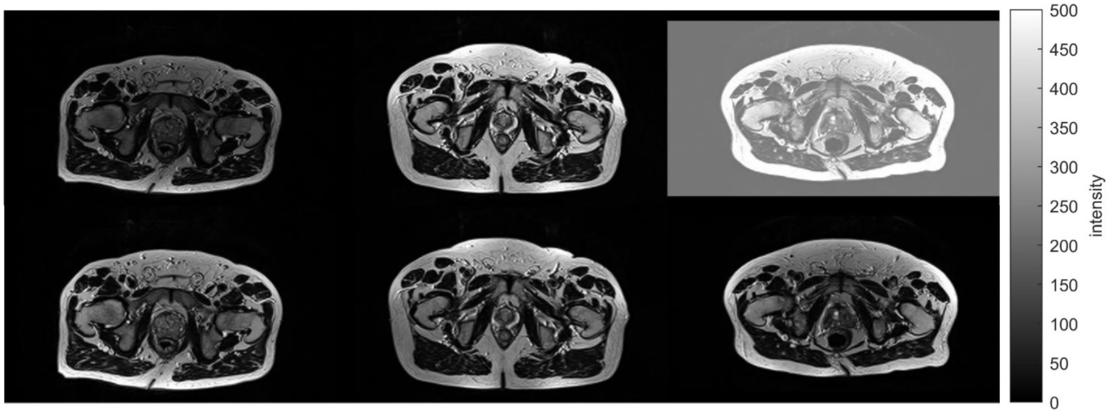


Figure 3.2: Demonstration of the change in variability between images of the pelvis region prior to normalization (top) and the same images post normalization (bottom) clearly showing the better intensity agreement in the latter. Intensities in the upper right image have been artificially altered in order to increase variability.

The benchmark histogram is simply a set of values that are used as landmark points in the transformation. In principle, one could choose an arbitrary set of increasing values without having to go through a cumbersome selection procedure. When the algorithm was proposed, the image standards used solely integers, in which case one had to be careful not to let the transformation truncate values such that information is lost. However, this problem is largely solved with modern floating point precision.

3.2.2 Generalized scale & generalized ball-scale normalization

Generalized scale (g-scale)[98,97] and generalized ball-scale (g_b-scale)[98,97] normalization are two normalization methods developed for brain MRI normalization. They both use the same piece-wise linear transformation as histogram normalization but they differ in the way the landmark points are chosen. The basic idea is to choose the points such that they more frequently correspond to the modes of different tissues, e.g. the mean of gray matter or white matter distributions. The methods are based on concepts in scale-space theory and fuzzy connectedness, which sprang out of the areas of computer vision and signal theory due to the fact that important information is present at different scales in real-world systems. By incorporating such methods, the models can become semi-locally adaptive, which has certain advantages over both local and global scale formulations individually.

The g-scale method partitions an image into different regions based on the degree of "hanging-togetherness", which is to say regions that are locally homogeneous. The g-scale at any voxel p in an image I is defined as the largest fuzzily connected subset of I containing p, such that all voxels in the g-scale satisfy a predetermined homogeneity criterion. The criterion can be thought of as a threshold hyperparameter that controls the inclusiveness of pixels—a lower threshold means that pixels are more likely to be included in the subset, which leads to fewer regions globally. The result after applying this calculation to the whole image is a new array defining the fuzzily connected regions in the image. The actual transformation is done by (3.2) after letting q be the mean intensity within the largest g-scale region.

The g_b-scale differs from the g-scale method in the way the image is partitioned into regions. Instead of using the original image, the g_b-scale method uses the so-called ball-scale (b-scale) scene when determining the connectivity. This b-scale scene is an image where each voxel value is determined by its b-scale value, which is defined as the radius of the largest ball, centered at the voxel, that contains only similar voxels (for some predefined similarity condition). The g_b-scale at any voxel p in an image I is then defined as the largest connected subset of I containing p such that the b-scale of voxels within the g_b-scale is greater than a predetermined threshold value. Thus, the difference between the g_b- and g-scale methods is that the g_b-scale uses a homogeneity criterion on the b-scale scene of a given image, whereas the g-scale uses a homogeneity criterion based on fuzzily connected areas calculated directly on the original image. The transformation is again done by (3.2), but the q value is set to the mean intensity within the largest g_b-scale region. One of the advantages of the g_b-scale method over g-scale was shown to be increased resistance to noise artifacts in the images.

3.2.3 Prostate-specific normalization

The prostate-specific normalization method relies on matching the intensities within the prostate region itself rather than finding landmarks within the foreground image (as in method I) or the largest connected sub-region (as in methods II and III). The method uses the median intensity within healthy prostate tissue as the landmark point q. However, this requires segmentation maps of the prostate, the lesions, and preferably also the urethra. The rationale for this approach is to ensure that the landmark points of different patients correspond to healthy prostate tissue for all patients. Thus, the healthy prostate intensity distributions should co-align post normalization, and, as such, intensity deviations from the healthy prostate tissue, which is typical for cancerous lesions, ought to be more consistently identified.

3.2.4 Dataset

The experiments in this chapter were conducted on axial T2-w pre-radiotherapy MRI scans of 49 patients with low- to intermediate-risk organ-confined prostate cancer in a phase II prospective, single-arm, single-center clinical trial. A summary of the patient characteristics is presented in Table 3.1. All images were acquired using a 1.5 T scanner (AvantoFit,

Table 3.1: Summary of prostate cancer characteristics for the study cohort.

Characteristic		Number of patients
PSA		6.47 (3.07)*
T-stage	cT1	7 (14%)
	cT2	42 (86%)
Gleason Score	3+3	26 (53%)
	3+4	17 (35%)
	4+3	6 (12%)
EPE score	1	3 (6%)
	2	12 (24%)
	3	15 (31%)
	4	19 (39%)
PI-RADS score	2	1 (2%)
	3	4 (8%)
	4	27 (55%)
	5	17 (35%)
Risk class	High	8 (16%)
	Low	41 (84%)

*Mean (standard deviation)

Siemens Healthineers) with a slice thickness of 3 mm, pixel spacing 0.59×0.59 mm, echo time 118 ms, and repetition time 3780 ms. The prostate, urethra, and dominant intraprostatic lesion (DIL) were identified and manually contoured on each image slice by one of three radiologists with 4 to 8 years of experience.

3.2.5 Evaluation criteria

The four different normalization methods were applied to the 49 3D images, resulting in five different image sets including the raw images. To evaluate the methods, we analyzed the resulting images based on their intensity distributions and the corresponding radiomic features.

Intensity-based evaluation

To quantify the similarity within tissue types across images we used two measures: the standard deviation of the normalized mean intensity (NMI), a quantity that has been used in image normalization comparisons previously[109], and the coefficient of variation (cv) among all healthy and cancerous voxel intensities. The NMI for a patient p and tissue type t (here either healthy or cancerous prostate tissue) is defined on the MRI volume as

$$\mathrm{NMI}_{p,t} = \frac{\mu_{p,t}}{s_{2,p} - s_{1,p}} \tag{3.3}$$

where μ is the mean intensity, and s_1 and s_2 are the 2nd and 98th percentile intensity values. The coefficient of variation is defined over the collection of all intensity values for a specific tissue type t among all patients by $cv_t = \sigma_t / \mu_t$ where σ and μ are the standard deviation and the mean, respectively.

We also compared the difference between distributions of healthy and cancerous tissue. It is reasonable to suspect that downstream analyses like radiomics calculations, segmentation models, or computer vision models would benefit from a greater separation between healthy and cancerous tissue since they all operate directly on the intensities. This separation was quantified in three ways: the normalized difference of means (NDM) between healthy and cancerous intensities, the Jeffreys' divergence (J-divergence; not to be confused with the Jensen-Shannon or JS-divergence), and the Wilcoxon signed rank test of the differences between the pre- and post-normalization distances.

If we let $\mathcal{I}_h^i$ be the multiset of all healthy intensity values and $\mathcal{I}_c^i$ be the multiset of all cancerous intensity values for a patient i, then the mean NDM can be calculated as

$$\langle \mathrm{NDM} \rangle = \frac{1}{N} \sum_{i}^{N} \frac{|\langle \mathcal{I}_h^i \rangle - \langle \mathcal{I}_c^i \rangle|}{\langle \mathcal{I}_h^i \uplus \mathcal{I}_c^i \rangle} \tag{3.4}$$

The J-divergence between the healthy and cancerous tissue distributions were computed over all MRI volumes considered together (similar to Shah 2011[125]). It is defined for two probability distributions p and q by the sum of their relative entropy (also termed Kullback-Lieber divergence) to each other:

$$\mathrm{J}(p, q) = \int\limits_{-\infty}^{\infty} p(x) \log \frac{p(x)}{q(x)} dx + \int\limits_{-\infty}^{\infty} q(x) \log \frac{q(x)}{p(x)} dx. \tag{3.5}$$

In contrast to the relative entropy, this measure is symmetric, which makes it suitable as a distance metric.

Radiomic-based evaluation

The extraction of radiomic features was performed on the whole prostate region (excluding the urethra) using the IBEX[157] software with parameters optimized for the data set[*]. All available features were calculated for every image volume, resulting in 1702 features for each patient. Any feature with identical values to one already included was removed from the feature set, leaving 1058 features for the analysis.

[*] The following parameters were modified: Neighbour Intensity Difference 2.5: RangeMax=8000, Nbins=100. Intensity Histogram Gauss Fit: RangeMin=1, RangeMax=60000, RangeFix=1. Intensity Histogram: RangeFix=0, Nbins=100. Intensity Direct: ThresholdHigh=8000. Gray Level Run Length Matrix 2.5: GrayLimits=[], NumLevels=100. Gray Level Cooccurence Matrix 2.5: GrayLimits=[], NumLevels=100. Gray Level Cooccurrence Matrix 3: GrayLimits=[], NumLevels=100. Empty brackets (”[]”) indicate the input to be left empty.

To investigate how feature behaviour depends on normalization, the concordance correlation coefficient (CCC) between the pre-norm and post-norm values for each feature was analyzed. The CCC between two sets x and y measures their agreement and is often seen as a measure of reproducibility. It is defined as

$$\mathrm{CCC} = \frac{2s_{xy}}{\sigma_x^2 + \sigma_y^2 + (\mu_x - \mu_y)^2} \tag{3.6}$$

for standard deviations σ, means μ and covariance s_{xy}. The features were assumed to be unchanged if their CCC was above an absolute threshold of 0.8, indicating a very strong correlation[4].

3.3 Results

3.3.1 Qualitative assessment

Intensity histograms of the different normalization methods are shown in Figure 3.3. Each row represents a different normalization technique, with unnormalized images at the top, and columns corresponding to histograms for the whole image, healthy prostate tissue, and cancer tissue, respectively. Normalization methods I and III are almost identical due to very similar landmark values. Overall, normalization methods I and III appear to co-align histogram peaks very well, whereas normalization methods II and IV instead tend to have a larger variability at higher intensity values (as seen in the histograms of the full images). Normalization method II seems to worsen the alignment altogether.

Apart from normalization method II, the healthy tissue looks similarly distributed between the methods, with slightly more co-aligned peaks after normalization. Cancerous

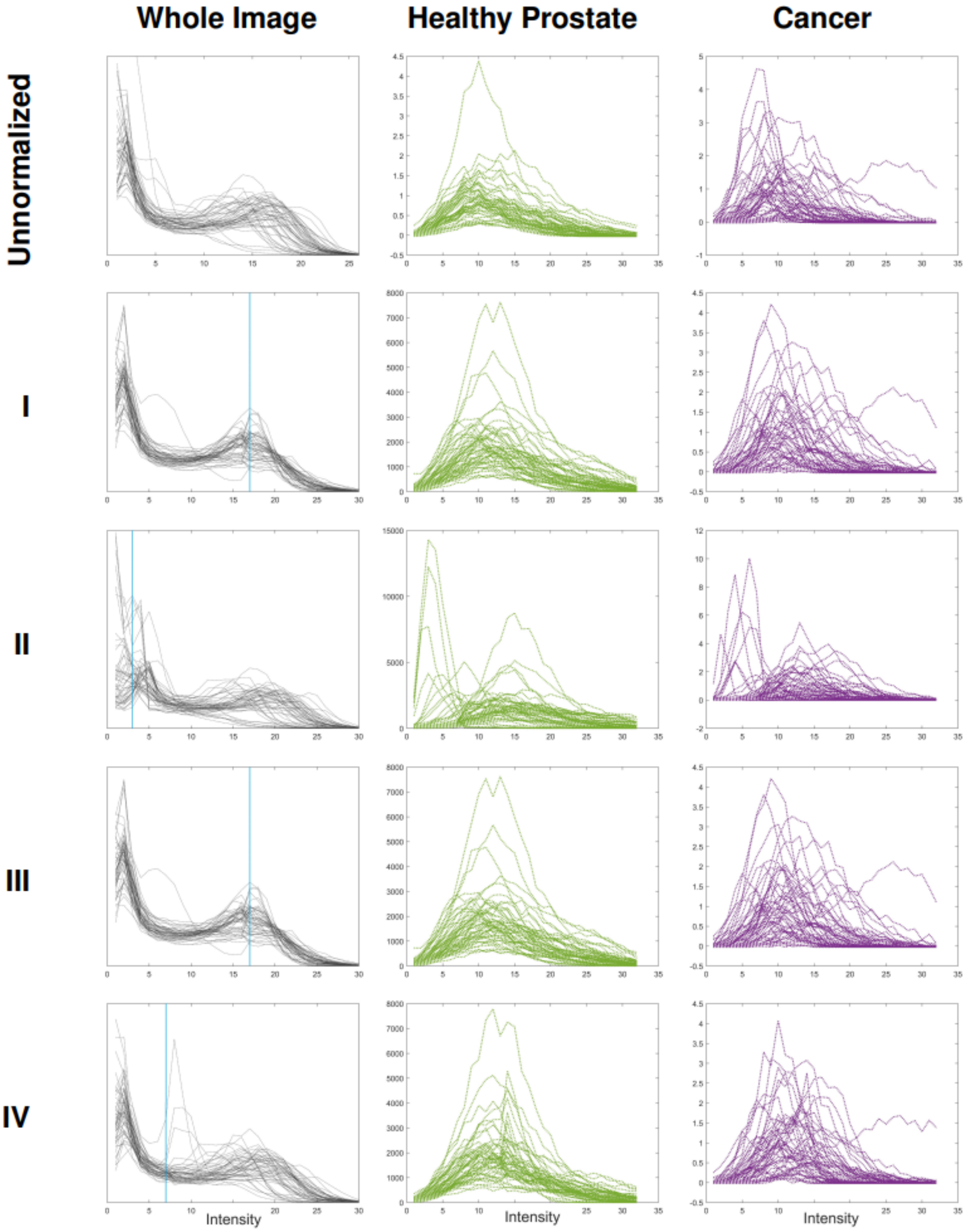


Figure 3.3: Intensity histograms of the full images, healthy prostate tissue, and cancerous tissue from different normalization methods. The top row shows unnormalized images, and the following rows show normalization methods I to IV, respectively. Vertical blue lines in the left column indicate the histogram landmark point (q). Intensities have been grouped into 32 bins to smooth out the histograms. The vertical axes in the last two columns have been scaled to favor visual comparison between columns.

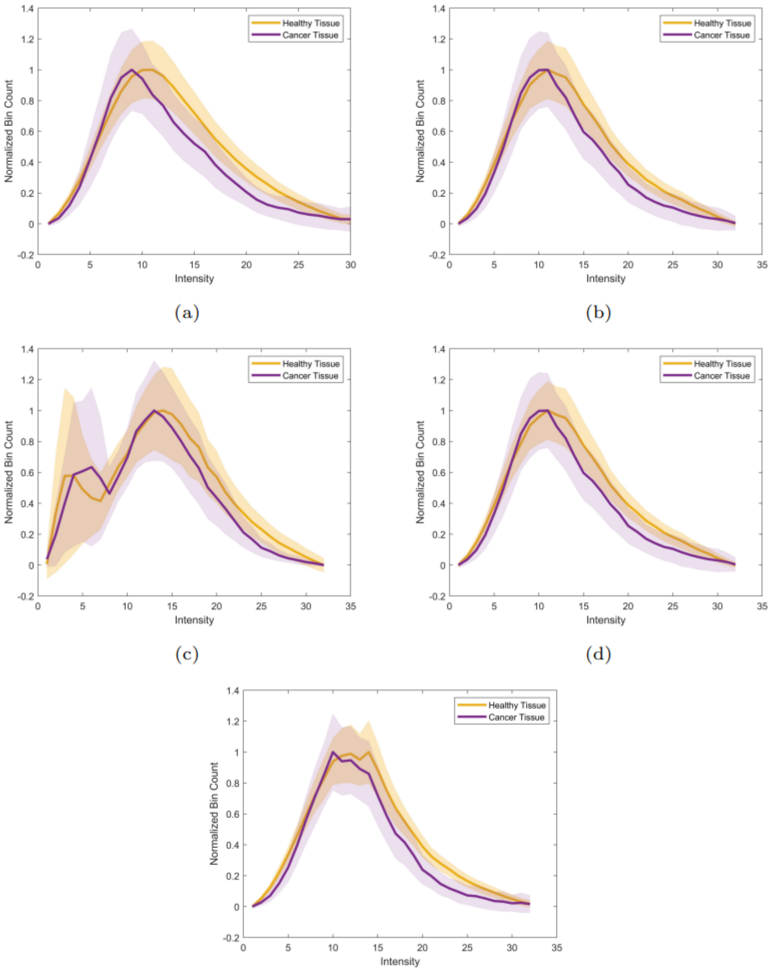
tissue intensity distributions are also very similar across different methods (excluding normalization method II), but with much higher variation within methods as compared to healthy tissue. Apart from normalization method II, it is hard to notice any difference at all for cancerous tissue across methods.

In order to more readily visualize the difference in intensity distributions between healthy and cancerous prostate tissue, Figure 3.4 shows the mean histogram of the collection of histograms in the second and third columns of Figures 3.3. In this ensemble view, there is only a minor skew between the healthy and cancerous intensities, and the difference is somewhat larger prior to normalization.

3.3.2 Intensity-based evaluation

The results for both the standard deviation of the NMI and the coefficient of variation from inter-tissue analysis (Table 3.2) follow the same pattern: prostate-specific normalization (method IV) has the most consistent (the least varying) intensities for both healthy and cancerous tissue, followed by normalization methods I and III which gave identical results. Method II performed worse than no normalization at all.

In the comparison of the distance between healthy and cancerous tissue intensity distributions (Table 3.3), prostate-specific normalization (method IV) produced the largest separation between healthy and cancerous tissue intensity populations, both in terms of mean NDM and J-divergence. Normalization methods I and III, on the other hand, more consistently produced an increased separation post normalization, as evident by the lower *p*-value (36 out of the 49 normalized images had an increased difference after normalization in method I and III, as opposed to 31 images in method IV). *g*-scale normalization (method



(a) (b) (c) (d) (e)

Figure 3.4: Mean histograms of patients' healthy and cancerous prostate tissue constructed from the collection of histograms shown in the second and third column of Figure 3.3. Transparent bands indicate the 95% confidence interval of the mean. (a) Unnormalized images (b) method I: Histogram normalized images (c) method II: g-scale normalized images (d) method III: g_b-scale normalization (e) method IV: prostate normalization.

Table 3.2: Intensity variation within healthy and cancerous tissue, measured by the coefficient of variation (*cv*) and standard deviation of the normalized mean intensity (NMI). A low variation indicates good agreement of intensity values for the specific tissue. Parentheses represent rank (lower is better).

	cv		SD(NMI)	
	Healthy	Cancer	Healthy	Cancer
Unnormalized	0.528 (4)	0.522 (4)	0.068 (4)	0.092 (4)
Method I	0.518 (2)	0.489 (2)	0.057 (2)	0.086 (2)
Method II	0.557 (5)	0.493 (5)	0.094 (5)	0.101 (5)
Method III	0.518 (2)	0.489 (2)	0.057 (2)	0.086 (2)
Method IV	0.510 (1)	0.464 (1)	0.032 (1)	0.078 (1)

Table 3.3: Metrics of separation between healthy and cancerous tissue intensity distributions, measured by the mean normalized difference of means (NDM), and the J-divergence. Large values indicate good separation of cancerous and healthy tissue intensities. The *p*-value of the Wilcoxon signed rank test indicates the significance of the NDM change between unnormalized and normalized images.

	Mean NDM	Wilcoxon signed rank test (*p*-value)	J-Divergence ($\times 10^{10}$)
Unnormalized	0.1998		705
Method I	0.2004	0.0027	1125
Method II	0.1835	0.0001	1165
Method III	0.2004	0.0027	1125
Method IV	0.2061	0.0518	2206

II) increased the similarity of tissues in terms of mean NDM, which in turn means that its low *p*-value is largely irrelevant since the interest lies in decreasing the similarity. After method IV, method II produced the greatest J-divergence, followed by methods I and III.

3.3.3 Radiomic-based evaluation

Figure 3.5 illustrates the CCC between features pre- and post-normalization. In general, features are less impacted by the histogram and g_b-scale normalizations. The number of

Table 3.4: Numbers of unchanged features (CCC$\geq$0.8) in different feature categories for the different normalization methods. Percentages indicate the fraction of unchanged features within each category. GOH: gradient order histogram, GLCM25/GLCM3: gray-level run-length matrix in 2.5/3 dimensions. Left-out categories (GLRLM25, ID, IH, IGGF, NID25, and NID3) had no unchanged features.

	GOH	Shape	GLCM25	GLCM3	Total
Method I	39 (89%)	17 (94%)	24 (8%)	57 (10%)	137 (13%)
Method II	23 (52%)	17 (94%)	21 (7%)	44 (8%)	105 (10%)
Method III	39 (89%)	17 (94%)	24 (8%)	56 (10%)	136 (13%)
Method IV	38 (86%)	17 (94%)	21 (7%)	53 (9%)	129 (12%)

unchanged features (above 0.8 CCC) in the different methods are presented in Table 3.4. Eighty-nine (89) features had a CCC very close to zero, which occurs if either variable has a very small variance or if the linear correlation between them is close to zero. These features were primarily in the IntensityDirect (44 features) and IntensityHistogram (17 features) categories. The features in the Shape category are mostly unaffected since they measure shape parameters (e.g. the radius of the ROI).

3.4 Discussion

Our results support the claim that image normalization is as important for prostate MRI as it is for brain MRI, as demonstrated by the significant changes in intensity distributions as well as the vast majority of radiomic features. In contrast to most other normalization-oriented studies, we have focused on a single homogeneous data set acquired within a single institution, which is more akin to the working situation in clinical practice. The results demonstrate that image normalization is no less important in these cases.

The method that most often resulted in smaller coefficients of variation and standard deviation of values within tissues (taken to be desirable in terms of indicating higher intra-

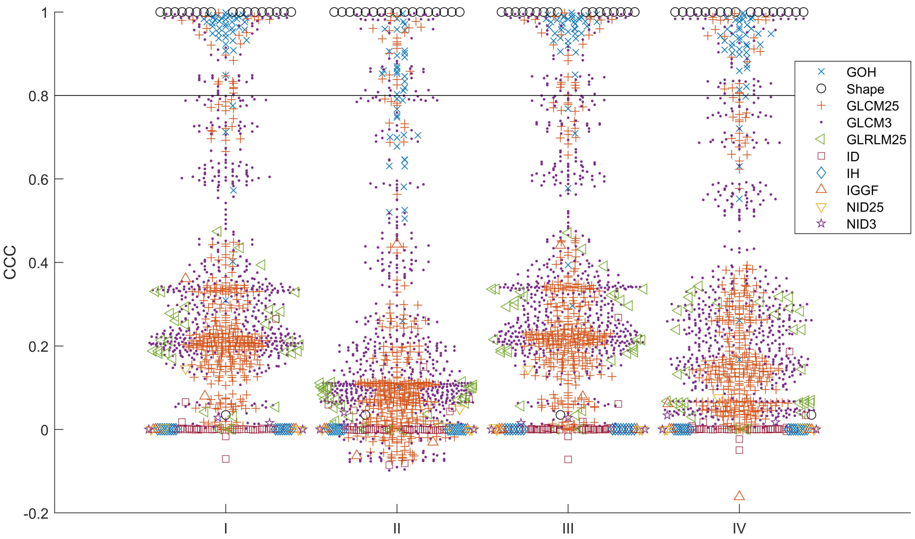


Figure 3.5: Concordance correlation coefficients between radiomic features from unnormalized images and images normalized by methods I, II, III, and IV. A single marker represents the concordance correlation of a single feature (each feature has 49 different values—one for each patient). Marker symbol and color represent different groups of features according to the legend, with the following abbreviations (in order): gradient orient histogram, gray level cooccurrence matrix (2.5 and 3 dimensions), gray level run length matrix (2.5 dimensions), intensity direct, intensity histogram, intensity histogram Gauss fit, neighbor intensity difference (2.5 and 3 dimensions). This figure visualizes the huge changes in feature values caused by normalization as well as the similarity between methods I and III.

tissue similarity) was that of prostate normalization (method IV), which normalizes images according to pixel intensities within healthy prostate tissue. This method was followed by a tie between histogram normalization (method I) and g_b-scale normalization (method III). While method IV also provided the greatest overall separation between distributions of healthy and unhealthy prostate tissues, methods I and III resulted in more individual subjects showing a statistically significant increase in the difference between healthy and unhealthy prostate tissue intensity distributions. A significant reduction in the separation of the distributions was found for method II, which would make the distinction between healthy and cancerous prostate tissue harder.

Methods I and III had arguably the best performance by the measure of feature robustness (assuming fewer drastic changes are to be preferred), although the differences between the top three methods are minor (137, 136, and 129 unchanged features, respectively). It should be noted, however, that comparing how features change is not intrinsically conclusive; it may be more appropriate to analyze the performance of the radiomic prediction models in which they are to be included. But based on the fact that a purely random image transformation would have very few features that are concordant, this analysis can still be useful.

Normalization method II (g-scale normalization) had the worst performance of the methods and was even worse than unnormalized images both in terms of tissue consistency and healthy/cancerous tissue separation. This is most likely a consequence of the low intensity of the landmark point within the largest g-scale region (after inspection, we found that the region often corresponded to locally homogeneous regions of dark muscle mass). It should be noted that this method is highly parameter-dependent and that other parameters may be able to produce results in line with the other methods. Our results for this method make it clear that caution should be exercised when performing normalization, and that blindly applying normalization algorithms without concern for their consequences and applications may severely alter, and even invalidate one's results or conclusions.

The large changes in the radiomic feature values due to normalization make it clear that normalization alone is not sufficient to prevent the reproducibility and variability problems within radiomics. To properly compare studies, one would most likely have to implement normalization with the exact same method and landmark points in addition to having a similar data set.

One of the problems with normalizing images from cancer patients is that the pathology may influence the results. Lots of cancerous tissue may skew the intensity distribution, especially if it is not isolated from the healthy tissue (as was done in method IV here). Moreover, prostate cancer is often multifocal, meaning that there can be cancerous tissue even outside of the dominant lesion, and there is often non-cancerous disease or inflammation in surrounding areas. Therefore, what we consider healthy prostate tissue might include some abnormalities, and, as a consequence, the similarity seen between healthy and cancerous tissue could be deceptively small. In the current clinical workflow, there is no solution to this issue since it would require results from prostatectomy specimens, which are not usually collected.

A possible improvement to the normalization algorithms is to use multiple landmark points, as was done in a follow-up study of the histogram normalization[110]. However, the authors concluded that the performance is not significantly improved with more landmarks, which has likely contributed to the popularity of the simpler one-point approach.

Since the landmark points by construction match the intensity values of the quantiles, it is likely that more landmarks would increase the overall similarity of the images. However, it is not universally true that higher similarity is to be preferred. As noted earlier, it may be more appropriate to analyze the performance of the radiomic prediction models in which the images are included.

The foreground identification step in the histogram method (step one of the algorithm as described above, which we labeled optional) was not thoroughly discussed in the original publications. The original purpose of this step was to find the intensity of the first non-background mode of the histograms, which by assumption corresponds to pixels of interest. In effect, this increases the intensity values of the landmark points because it excludes low-intensity background pixels. The procedure performs well on brain MRIs where relevant tissues are dominated by hyperintense structures, but it may be less appropriate for the prostate, where the tissue of interest is dominated by values fairly close to the mean value of the full image (see the last row of Figure 3.3). Therefore, it may be more suitable to not use foreground identification when analyzing pictures of the prostate and similar structures.

Image noise is a critical factor that may influence the outcome of normalization. Spatially uniform Gaussian noise does not impact the quantile values, which leaves the histogram and prostate-specific method largely unaffected. For the Rician distribution of noise in MRI images[49], the effect depends on the parameters, but it is likely that normal noise levels do not severely alter the outcome. For the g-scale and g_b-scale algorithms, the connectivity of the pixels can be impacted, but their authors concluded that the effects were manageable[98]. The images analyzed in this study are likely representative of typical noise levels that would be encountered in standard clinical settings. As for the effect of noise on the radiomic features themselves, we refer to studies addressing this issue separately, such as Lafata 2017[86] or Pfaehler 2019[113].

While the size of the patient cohort in this study is not comprehensive, we believe that it is adequate to provide a good understanding of how the normalization methods behave for typical clinical images. Indeed, cohorts of other prostate cancer radiomic studies typically include similar numbers (e.g. 33[1], 54[148], 64[58], etc.), and have even gone as low as 23[106]. On the other hand, increasing the normative cohort size of prostate cancer radiomic studies should be a legitimate concern for future research in the field.

4

Computer-aided prostate cancer assessment

4.1 Background

Prediction models for tabular data are a staple in machine learning research since considerable amounts of real-world data are collected in the form of spreadsheets, from financial movements in the stock market to the time and space variations of the weather. In medical research, traditional algorithms such as logistic regression or support vector machines are often preferred when building prediction models, in part because they naturally integrate with statistical inference such as the likelihood ratio test. When performance is a more pressing concern, as is the case when the goal is the actual deployment of a model, modern algorithms tend to be superior. One of the most popular types of models is gradient-boosted decision trees (GBDTs), which use the gradient-boosting technique to iteratively build an ensemble of decision trees (analogous to a gradient descent algorithm). A great deal of research has also been devoted to developing DL models that can compete with GBDTs; some noteworthy efforts being TabNet[7], NODE[116], DNF-Net[2], SNN[82], GrowNet[8], DCN V2[149], and AutoInt[133]. However, when evaluated on a wide range of different tasks, many of these models have been shown to offer little to no improvement over simpler baselines such as 1D convolutional neural networks (CNNs) or multilayer perceptrons (MLPs)[47,129,77,132,83]. The question remains as to whether modern DL models have the potential to compete with GBDTs in radiomics, where the datasets are characterized by few samples, many variables, and a mix of categorical, ordinal, and continuous data.

In radiomics, a recurring problem is that it is difficult to predict a priori which features will be useful for specific tasks. Moreover, many are known to be non-robust or irreproducible[160,141,9,154,39]. Therefore, a radiomic model or pipeline must identify the features to use and those to ignore, which is a challenging task given the number of features involved. Many modern prediction models have inherent attention mechanisms that allow them to focus attention on useful features, for instance by weighting the inputs differently, but these mechanisms can be hard to train when the input dimension is large. As a result, it is customary to reduce the features further, e.g. by using a clustering method. A prediction pipeline's ability to focus on important features is directly reflected in its performance since non-relevant features usually make predictions worse. At present, it is not known how different models and feature selection strategies compare when it comes to finding and utilizing the information from important radiomic features and simultaneously ignoring the noise from useless features.

In clinical prostate oncology, it is recommended practice to characterize the tumor via both MR-image-related parameters (such as prostate volume, number of dominant lesions, PI-RADS score, EPE value, and ADC value) and biopsy-related parameters (such as initial PSA, ISUP grade group, tumor stage, lymph node status, and risk class) in order to select an appropriate treatment and follow-up schedule. Surgery by radical prostatectomy (removal of the prostate) and radiotherapy are the two most common treatments for localized prostate cancer and have comparable oncological outcomes. In the latter case, the pathological assessment of the surgical specimen is unavailable, making accurate evaluation harder for the clinicians, leading to an increased risk for over- or undertreatment. Thus, the information from the imaging and biopsy can constitute an invaluable asset to the pre-treatment evaluation of the patient. In this scenario, accurate non-invasive prediction of pathological determinants of prostate cancer at the initial diagnosis or before surgery could improve decision-making and patient prognosis, but so far this has not been studied. In this chapter, we focus on predicting multiple endpoints from prostate cancer pathology which all hold critical clinical value.

In most cases, radiomics studies have focused on deriving features from the dominant cancerous lesion, but some studies have suggested that whole-prostate radiomics also can be applied with some success[36,45,151,50]. This has the potential to circumvent the need for the tedious and highly observer-dependent process of lesion segmentation, leading to reduced costs and time requirements. Moreover, this incorporates information from non-dominant and heterogeneous lesions and other relevant conditions such as infections and benign lesions. At present, however, this approach needs to be further evaluated, particularly with larger datasets, before it can be considered viable.

When developing radiomic models, an often overlooked aspect is the involvement of the different features. Various features' influence can act as both an interpretation tool and a quality or security check. If researchers are not vigilant, results that appear to be a consequence of radiomic features may in fact be a consequence of small implementation changes or simply random variations, which can lead to dubious conclusions. Conversely, if we observe a model relying on the same variables as clinicians for a well-known condition, it incites trust and provides a sense of reliability that may be critical for its clinical integration. In this context, clinicians might disregard the prediction of an algorithm when they notice that it uses information from irrelevant sources, or update their own beliefs when they notice that the algorithm took into account something they missed. Therefore, an analysis of the different features' impact should be carried out whenever possible.

In this chapter, we address the issue of building ML models with radiomics for the prediction of pathological prostate cancer endpoints. It will be divided into two parts aiming to approach the problem from two distinct viewpoints: one technical part, honing in on the specifics of the implementation and the performance of the models, and one clinical part, focusing on the clinical utility and the concrete behavior and tendencies of the models. The specifics of the two parts can be summarized as follows:

1. **Building & comparing clinical prediction models**: in this section, we focus on constructing the best-performing models. To do this, we compare the performance

of four common ML models for tabular data, including both GBDT and variants of different successful DL models. We evaluate the models in predicting nine different pathology endpoints of clinical value in prostate cancer care. To combat the problem of overfitting and overoptimistic estimates, we employ rigorous training, validation, and test procedures in all experiments.

2. **Evaluating predictions and model behavior**: in this section, we further explore the best-performing model. We will analyze the role of radiomics in relation to the clinical and radiological information, what variables the model relies on, and how the model behaves for specific sub-groups within the cohort such as low- vs. high-risk patients. This type of analysis will be crucial for the clinical integration of ML models because it provides a perspective on how the model behaves and makes decisions. This can give users and clinics a sense of when using the model is appropriate or inappropriate.

This chapter presents material that is currently under peer review in multiple undisclosed scientific journals.

4.2 Dataset

Acquisition

The patient data used in this chapter were retrospectively collected from an unprecedentedly large cohort of 949 prostate cancer patients who had undergone multiparametric prostate MRI and prostatectomy in the European Institute of Oncology (IEO) between 2015 and 2018. For each patient, we used the T2-weighted MRI sequences and all relevant clinical variables.

The MRI images were acquired using 1.5 T MR scanners with slice thickness 3.0-3.6 mm, slice gap 0.3 mm, pixel spacing 0.59×0.59 mm, echo time 118 ms, and repetition time 3780 ms.

Each image was corrected with the N4 bias-field correction algorithm (implemented in sITK 2.1.1 using default parameters), and the image intensities were subsequently normalized with an outlier-aware range normalization that linearly maps the 0th and 99th percentile values of every image to a predefined range (in this case between 0 and 424). Every unique value in the 100th percentile was appended after 424 using a 1:1 linear map (the first unique value was mapped to 424+1, the second to 424+2, etc.)

The clinical variables included age, initial PSA, and comorbidity as well as MRI-related data such as prostate volume, number of dominant lesions, PI-RADSv2 score, extraprostatic extension score, and ADC (apparent diffusion coefficient). The following tumor-related variables were also obtained: initial ISUP grade group, clinical tumor stage (T), clinical lymph node status (N), and NCCN2019 risk class. For prediction, we included six outcome variables related to the pathological assessment of the patients: post-operative ISUP grade group, pathological T, pathological N, surgical margin, biochemical progression, and clinical progression. An overview of the clinical characteristics within the cohort is presented in Table B.1.

Image segmentation

The prostate in each image was segmented with the best-performing DL model (the adapted EfficientDet3D) from Section 2.4. To maximize its inference performance, the model was retained on the full dataset for roughly 1600 epochs (about twice as long as the convergence point). The training was performed twice with different random seeds, and the final segmentation masks were established by taking the average of both outputs.

In order to ascertain an acceptable standard for the automatically generated contours, a subset of the segmentations was selected and sent for correction by an expert radiologist. Segmentations were included in this subset if at least one of the following two criteria was met: 1) the segmentation had an estimated Dice coefficient of 0.8 or lower, and 2) the volume of the segmentation was in the upper or lower two percentiles. The Dice coefficient was estimated by the QA model developed in Section 2.5, but calculated as the average of all five models from the different validation folds. In total, 98 segmentations were selected.

Radiomic feature extraction and pruning

Radiomic features were extracted from the whole prostate using PyRadiomics[143] 3.0 in Python 3.7. All available features were extracted from all available filter classes (Laplacian of Gaussian, wavelet, square, square root, logarithm, exponential, gradient, local binary pattern 2D, and local binary pattern 3D). For the Laplacian of Gaussian filter, we calculated features for three different values of sigma corresponding to 1, 2, and 5 times the in-plane spacing (0.59375 mm). No image resampling was performed since all the images had nearly identical resolution and spacing. In total, 1967 features were extracted.

Features were removed if their variance was lower than 1^{-6} or if they had an absolute Spearman correlation above 0.98 with any other feature. In the latter case, the feature with the lowest cumulative correlation with other features was kept. After these steps, 737 features remained.

Data preprocessing

The clinical variables were ordinally encoded as follows:

- Clinical and pathological T: stages were encoded by their integer value such that both T3a and T3b were encoded as 3, and T2a and T2b as 2, etc.
- Clinical and pathological N: stages were encoded by their integer value (N1→1, N0→0).
- Risk class: "low" and "very low" was encoded as 0, "intermediate favorable" and "favorable" as 1, "intermediate unfavorable" and "unfavorable" as 2, and "high" or "very high" as 3.
- Clinical progression: absence of progression was encoded as 0, and any type of progression (pelvic, extrapelvic, or both) was encoded as 1.

To adapt the data for the DL models, all input variables were normalized with a quantile transformer (with the number of quantiles set to the number of training samples), and missing values were imputed with a k-nearest neighbors imputer with $k = 8$ and distance-based weights (see Appendix B.2.1 for a deeper analysis on the imputation and its parameters). Quantile transformers, which have been used with great success in previous studies[47,46], circumvent some of the limitations of other common methods such as z-norm and min-max scaling by being robust to both outliers and scale variabilities[124]. Neither imputation nor quantile transformation is needed for GBDT models since they inherently handle missing values* and the tree-growing algorithm uses cutoff values that are scale indepen-

*In the GBDT library that was used in this study (CatBoost), they are considered smaller than the smallest real-valued entry, effectively making them a separate category.

dent.

The full list of all target variables is given in Section 4.3.1. Prior to training the models, we binarized the non-binary target variables (post-operation ISUP grade, pathological T, and the delta variables), allowing us to readily compare the performance of all different targets with the same scale and metrics. We also observed that this achieved better classification performance than binarizing the predictions after training the models with regression. This can be done without much loss of clinical utility since many of the decisions within the clinical workflow surrounding these parameters are largely made based on threshold values.

4.3 Building & comparing clinical prediction models

In this section, we compare four common high-performance ML models and learning techniques: gradient-boosted decision trees (GBDTs), a multilayer perceptron (MLP), a one-dimensional convolutional neural network (1DCNN), and a transformer model with a feature tokenizer (FT-Transformer)[47] specialized for tabular data. In addition, we examine the benefits of multitask learning, which involves training a model to predict several targets simultaneously, potentially leading to improvements in both speed and performance. These methods, albeit common in areas outside of healthcare research, have not previously been evaluated in a radiomics context.

4.3.1 Experiments

To compare the different models, we train them to predict six different pathological endpoints of clinical interest:

1. Post-operation ISUP grade group
2. Pathological tumor stage (T)
3. Pathological lymph node status (N)
4. Surgical margin
5. Biochemical progression
6. Clinical progression

as well as three additional ("delta") target variables based on whether or not there was a positive change between the clinical and pathological assessment of the ISUP, T, and N endpoints:

7. ΔISUP
8. ΔT
9. ΔN

Details of the models and their training routines are given below.

The models were compared in terms of their overall test performance in a nested five-fold cross-validation routine (see Section 4.3.3 for details). The performance was measured in terms of three different classification metrics: Matthews correlation coefficient (MCC), AUC, and accuracy.

4.3.2 Models

CatBoost

CatBoost[117] is a free open-source library for GBDT models and an alternative to the similar libraries XGBoost[19] and LightGBM[78]. GBDT models have been incredibly successful in tabular data analysis applications, in large part due to their high discriminative power yet simple structure and high speed. Unlike traditional methods, they can model arbitrary-dimensional decision bounds and unlike DL models, they grow their architecture iteratively, meaning that they are less sensitive to an initial choice of model structure/architecture. This allows users to perform hyperparameter searches without concern for the underlying network topology, which is typically not possible for advanced DL models. Instead, training GBDT models requires specifying only the number of decision trees used in the ensemble, their maximum depth (or the number of leaves), and the learning rate along with other parameters for regularization (e.g. L1/L2 regularization, subsampling, class weights, etc.).

MLP

The MLP model is one of the earliest and most straightforward DL architectures. It consists of layers of interconnected nodes, where each node in one layer is connected to every node in its preceding layer (it is "fully connected"). Despite their age and relative simplicity, they have been shown to perform well on tabular data if trained properly[77,46,83]. They are often faster and use fewer hyperparameters than their alternatives, which makes them attractive candidates for search-based optimization pipelines. Apart from regularization parameters, MLPs can be parameterized by just the number of layers and the number of

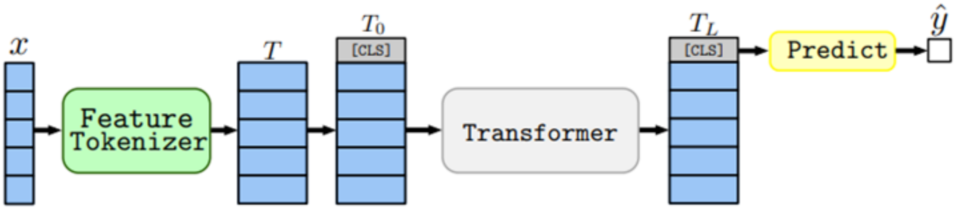


Figure 4.1: The architecture of the FT-Transformer. An input x is passed through a feature tokenizer and converted into embeddings T. Multiple transformer layers (T_0, ..., T_L) are connected in sequence after which the CLS token is used for prediction. Image source: Gorishniy 2021[47].

neurons within each layer. Our implementation used two layers (we found that more layers did not improve the performance but increased the training time) and a swish activation function followed by batch norm and dropout.

FT-Transformer

The FT-Transformer[47] was recently proposed as a transformer architecture specifically tailored for tabular data. The use of transformers in this scenario is not commonplace, but the research direction is attractive given the success of transformers in natural language processing[146] and more recently computer vision[35]. The model consists of a feature tokenizer module and several consecutive transformer layers (Figure 4.1), where the former converts the input variables (both categorical and numeric) into embeddings, and the latter performs the recurrent self-attention that is characteristic of the transformer. The final layer performs prediction with the CLS ("classification") token, which is designed to contain information about the sentiment of the whole input sequence (as opposed to information about individual elements in the sequence).

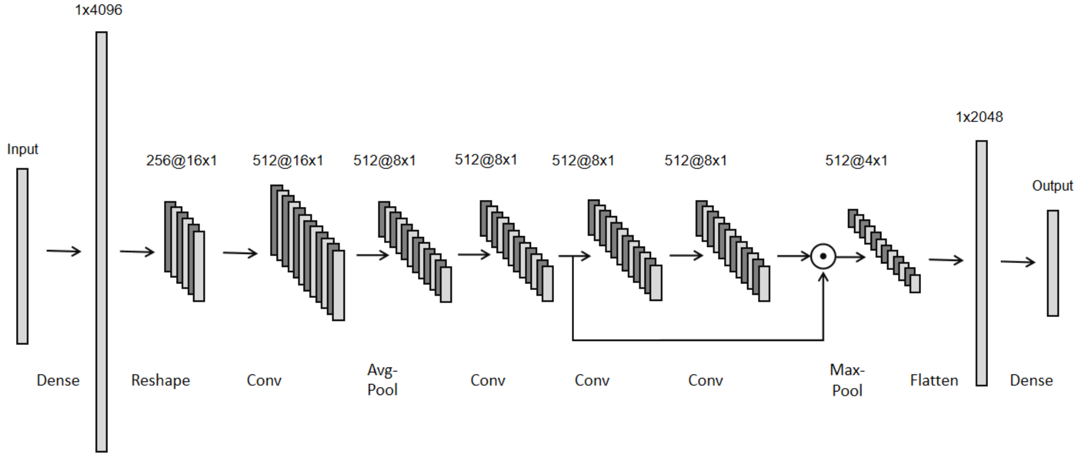


Figure 4.2: The architecture of the 1DCNN model. An initial dense layer maps the input feature vector into a larger vector more suited for convolutional operations. Standard convolutional and pooling operations are connected, and a normal dense layer performs the final prediction. The numbers represent the dimension at each step ($n_{channels}@n_{depth}$).

1DCNN

Convolutional archetypes are commonly used for image analysis because they can handle local correlations efficiently and are not impaired by positional and morphological variances. Tabular data, however, do not display these types of characteristics, which makes the value of CNNs less apparent in these scenarios. One way to circumvent this is to map the data into a higher-dimensional superspace in which the locality of convolutions is not an impediment. The implementation we used is taken from the second-place submission (the 1D-CNN itself had the best single-model performance) in a previous Kaggle competition for tabular data in 2020[10] and uses a learnable dense/linear layer to perform this mapping. After the initial dense layer, several 1D-convolution and pooling layers are connected in sequence along with a skip connection and a flatten operation (see Figure 4.2).

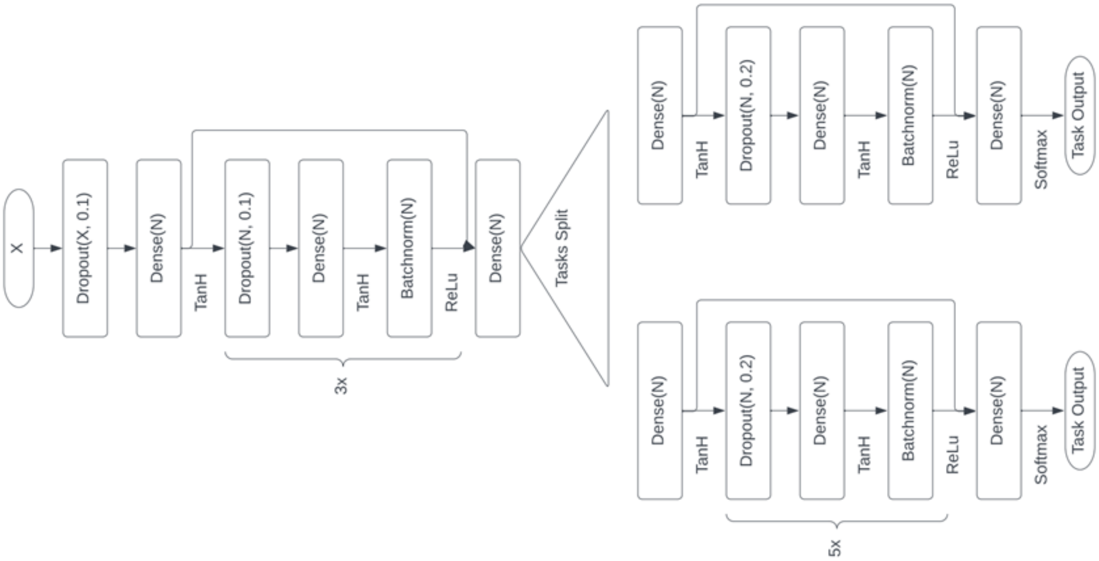


Figure 4.3: The architecture of the multitask-tailored MLP model. A base (left) structure is shared between all different tasks, and multiple prediction heads (right) are trained to optimize each individual task separately.

Multitask-tailored MLP

In addition to multitask variants of the four models above, an additional MLP model was created and optimized specifically for the multitask scenario. This is motivated by the fact that some design and architectural choices have no analogous counterpart in the single-task setting. In particular, the multitask model was constructed by two distinct partitions: a base that is shared among all different tasks, and N classification heads that have distinct parameters for each output task. The base handles the input and latent representations of the model while the heads are fine-tuned for their respective tasks. The architecture (displayed in Figure 4.3) builds upon the MLP and uses skip connections and varying activation functions.

4.3.3 Model training & feature selection

CatBoost

The CatBoost model was trained with a nested 5-by-5 stratified cross-validation (CV) and a standard binary cross-entropy (log-loss) loss function. The inner 5-fold was used to search for parameters and the outer 5-fold was used to estimate the performance. This eliminates the selection bias from selecting the lowest error model. To search for parameters, we used the Tree-structured Parzen Estimator (TPE) in the Optuna (v2.10) python package[3] and ran it for 600 trials with default parameters. Among the trials, we selected the model with the highest Matthews correlation coefficient (MCC) on the validation sets and proceeded to retrain it on the full training+validation set. The performance of the retrained model was then evaluated on the previously held-out test sets.

The hyperparameter search space is shown in Table B.2 in Appendix B.2.2. Note that we directly optimize the number of radiomic features to include ("n_features") based on their estimated predictive power over the target variable (see Appendix B.3 for details).

Deep learning models

The DL models were trained with a similar cross-validation procedure, but instead of a full 5-fold validation in the inner loop, we used a single train-test split (80%-20%) due to computational constraints. Moreover, the DL hyperparameter search was carried out with 64 trials instead of 600. All random seeds were fixed such that every model was trained and evaluated on the same data (including the CatBoost model).

The hyperparameter search spaces and complete training details for the different DL models are shown in Appendix B.2.2. As for the CatBoost model, we directly optimized the number of radiomic features to use within the parameter search (see Appendix B.3 for details).

We trained the DL models to directly optimize a differentiable version of the MCC, which can be achieved by defining continuous versions of the true and false positives/negatives. In other words, if we let y be the real target label and $h(x)$ be the network's prediction:

$$TP = \sum_i \mathbf{y}_i h(\mathbf{x}_i) \tag{4.1}$$

$$TN = \sum_i (1 - \mathbf{y}_i)(1 - h(\mathbf{x}_i)) \tag{4.2}$$

$$FP = \sum_i (1 - \mathbf{y}_i) h(\mathbf{x}_i) \tag{4.3}$$

$$FN = \sum_i \mathbf{y}_i (1 - h(\mathbf{x}_i)). \tag{4.4}$$

In the above formulae, it is assumed that the last operation in h is a logistic function such that the outputs are in the $(0, 1)$ range.

Multitask learning

For each of the four models mentioned above, we built alternative versions that were trained with a multitask/multi-objective loss function. This can be done with very minor modifications to the architectures (e.g. simply adjusting the number of outputs of the final layer) and has the potential to improve both speed and accuracy[140,29,144]. The multitask versions of the models were trained in the same way as the regular ones, but with three important modifications:

1. Since the loss of the models needs to be a single scalar value, we calculated the total loss by averaging the individual classification errors over all different targets (the binarization of the target values allows us to make this aggregation without needing to tune the weights between different types of targets).

2. A consequence of simultaneously predicting all targets is that missing target data needs to be handled differently. In the single-task case, we did not implement any target-specific imputation considerations since we could simply select and train on all the patients with target data available. In the multi-task case, the patients have varying target values missing, which means that some loss values are not defined. Hence, we divided the full data set into training and test data as the first step, and whenever a batch of samples with missing target data was encountered, we calculated the total loss for each patient as the mean loss over all its non-missing values[*]. A consequence of this is that regular (e.g. non-stratified) cross-validation has to be used.

3. A similar issue to point 2 is faced when we calculate the predictive power of the features prior to feature selection. The predictive power was thus also calculated by averaging the loss over all available targets.

[*] We can do this without worrying about undefined gradients since no sample has all its target values missing. If such a patient existed, it would not have been included in the dataset in the first place since we cannot possibly hope to receive any signal from it.

Table 4.1: Class distributions of the pathological target variables in terms of the number of positive and negative cases and percentages. In total, 949 patients were considered.

Class	n_{cases} (+/-)	% (+/-)	Missing
Post-op ISUP Group	877/68	93/7	4
Pathological T	582/367	61/39	
Pathological N	495/76	87/13	378
Surgical margin	716/232	76/24	1
Biochemical progression	637/140	82/18	172
Clinical progression	726/50	94/6	173
ΔISUP	603/343	64/36	3
ΔT	2030/715	24/76	4
ΔN	493/75	87/13	381

4.3.4 Results

Dataset

Table 4.1 shows the class distribution of the target variables (the number of positive, negative, and missing values) after discretization. A detailed overview of the clinical properties of the patient cohort is given in Appendix B.1.

Radiomic model performance

An overview of the performance of the models is displayed in Figure 4.4 and a summary of their relative scores in terms of their rank is displayed in Figure 4.5. For all endpoints, the CatBoost model achieved the highest MCC whereas the FTT and 1DCNN models appear to be nearly equivalent (2.33 and 2.78 mean rank, respectively). The MLP only achieved similar MCC to the other DL models in one of the nine cases (biochemical progression), resulting in the worst overall mean rank of 3.89. In terms of AUC, the results are similar, with the exception that the 1DCNN achieved a better mean rank than the FTT (2.33 vs.

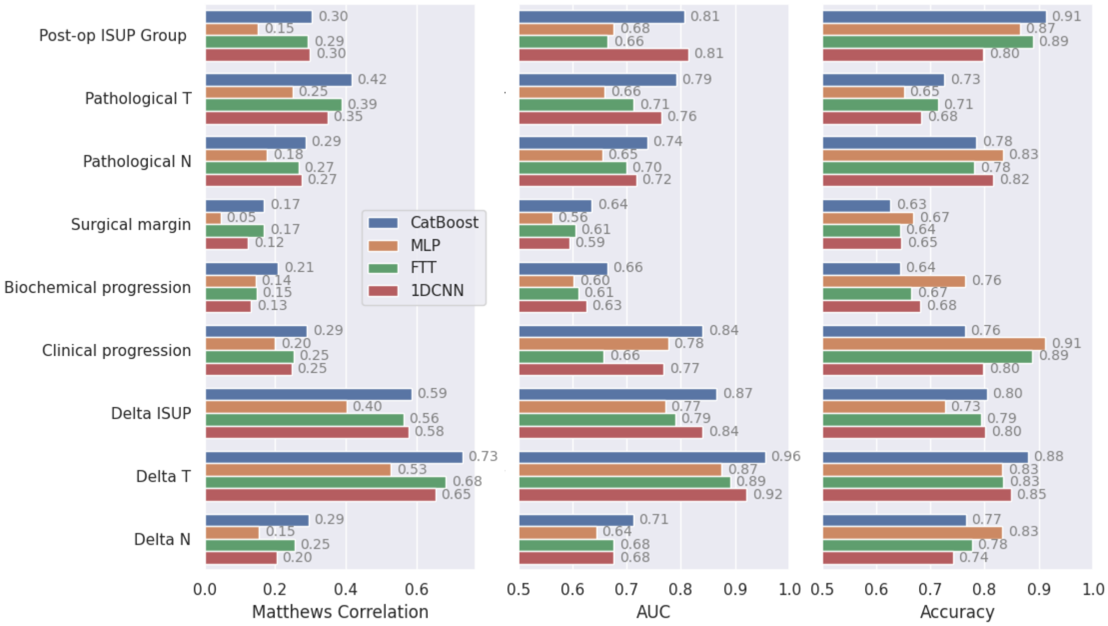


Figure 4.4: Performance of the different models in terms of Matthews correlation, AUC, and accuracy for the nine different prediction targets. Higher scores are better. MLP: multilayer perceptron, FTT: feature-tokenizer transformer, 1DCNN: 1D convolutional neural network.

3.0). The results for the accuracy are different: all four models performed at a comparable level, with MLP achieving the best mean rank. However, the mean rank difference between the MLP and the worst performing model, which is a tie between the FTT and the 1DCNN (mean rank of 2.67), is relatively small (0.56 mean ranks).

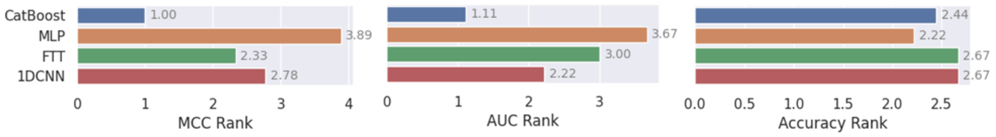


Figure 4.5: Total mean rank of the different models, aggregated as the mean over all nine different prediction targets. A lower rank is better.

Multitask performance

Figure 4.6 shows the performance of the multitask models and Figure 4.7 shows their respective ranks. The 1DCNN appears to be the overall best model in terms of both MCC and AUC (though tied for first place with the CatBoost model in the AUC case). CatBoost, MLP, and FTT all had comparable performance in terms of MCC, but the MT-MLP clearly performed worse. The MT-MLP has no AUC score since it was trained with regression, but between the remaining two models, MLP and FTT, the FTT performed better. The accuracy ranks are very different: the 1DCNN was the worst performing model and MLP was the best. The MT-MLP still performs poorly with its second-to-last place, but CatBoost and FTT perform very similarly.

Multitask improvement

In Figure 4.8, we show the mean percentage performance gained from training multitask models compared to several single-task models. The overall difference is very large in terms of MCC (16-31%) but fairly small for AUC (0-9%) and accuracy (2-6%). The added benefit of multitask training differs between models and metrics which makes it hard to declare a single best approach. Note that the performance difference can be positive for some endpoints even though the mean improvement is negative, and vice versa. The most drastic change was, by far, the MCC for the CatBoost model, which performed 31% worse in the multitask case on average. The second largest change was the MCC for the FTT model (-18%). In contrast, the MCC of both the MLP and 1DCNN benefited greatly (16% and 17%) from the multitask training. In summary, the mean performance was better in one case (and a tie in one case) for the CatBoost model, two cases for the MLP, two cases for the

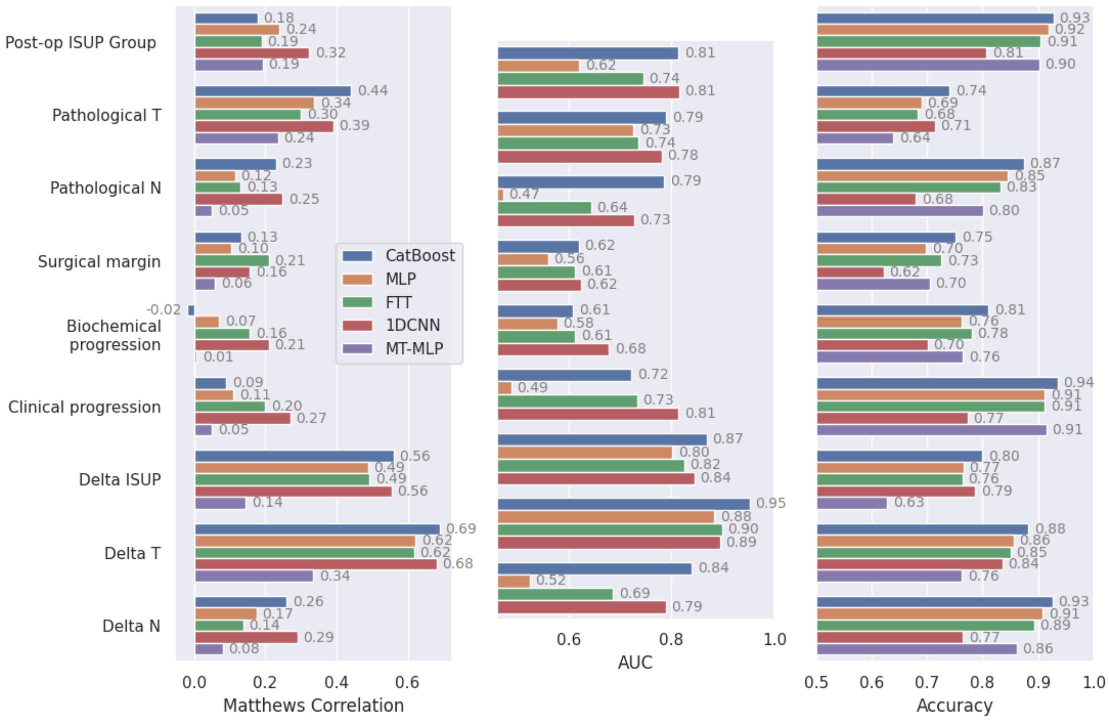


Figure 4.6: Performance of the multitask versions of the different models in terms of Matthews correlation, AUC, and accuracy for the nine different prediction targets. Higher scores are better. The MT-MLP model has no AUC since it does not output produce probability estimates.

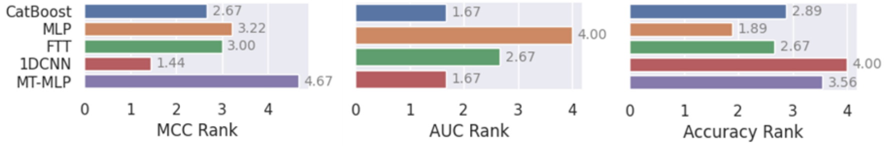


Figure 4.7: Total mean rank of the multitask versions of the different models, aggregated as the mean over all nine different prediction targets. A lower rank is better. The MT-MLP model has no AUC since it does not output produce probability estimates

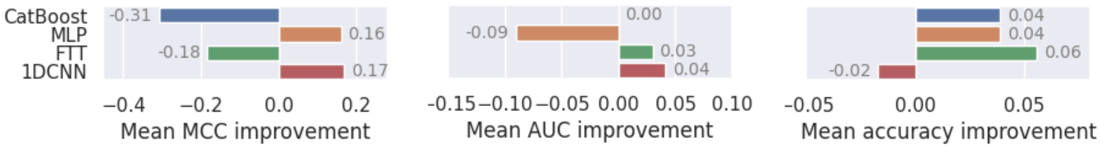


Figure 4.8: Mean improvement gained from multitask training of the different models, aggregated over all nine different prediction targets.

Table 4.2: The single best model, including both single-task (ST) and multitask (MT), in every prediction target and performance metric.

Target	MCC	AUC	Accuracy
Post-op ISUP Group	1DCNN (MT)	CatBoost (ST+MT), 1DCNN (ST+MT)	CatBoost (MT)
Pathological T	CatBoost (MT)	CatBoost (ST+MT)	CatBoost (MT)
Pathological N	CatBoost	CatBoost (MT)	CatBoost (MT)
Surgical margin	CatBoost	CatBoost	CatBoost (MT)
Biochemical progression	CatBoost, 1DCNN (MT)	1DCNN (MT)	CatBoost (MT)
Clinical progression	CatBoost	CatBoost	CatBoost (MT)
ΔISUP	CatBoost	CatBoost (ST+MT)	CatBoost (ST+MT), 1DCNN (MT)
ΔT	CatBoost	CatBoost	CatBoost (ST+MT)
ΔN	CatBoost, 1DCNN (MT)	CatBoost (MT)	CatBoost (MT)

FTT, and two cases for the 1DCNN.

By comparing the best multitask models with the best single-task models, we can discern the overall best-performing model across all endpoints (Table 4.2). Only two architectures are represented: CatBoost and, to a lesser extent, the 1DCNN. In the accuracy metric, multitask models are clearly overrepresented. In MCC and AUC, the single-task CatBoost model is the most common, and multitask 1DCNN the second most common.

4.4 Evaluating predictions & model behavior

This section further explores the CatBoost single-task model, which was the best of the models in the previous section. In particular, we analyze how the performance varies when changing the input variables, how the variables are used, and to what extent the different types of features (clinical, radiological, and radiomics) influence the predictions. This can allow us to identify the information needed to build good models and thus provide important insights into the model's behavior.

4.4.1 Methods

Roughly speaking, there are two angles from which the interesting behavior of the models can be analyzed: the performance angle (e.g. how good are the predictions) and the feature angle (e.g. what information is the model using when making its decisions). We elicited a list of concrete questions from both clinicians and engineers to directly illuminate the aspects of the model's behavior that they find important. The aim was to elucidate the ML models' usefulness and how they behave from both these stakeholders' perspectives. Below, we list the questions along with our approach to addressing them:

1. How influential are the radiological variables?

    - Since the radiological variables are crucial for the doctors' clinical assessment, it is interesting to know if this is also the case for ML models. If not, it may undermine the credibility of the ML models or raise new questions about how the model extracts complementary information. This can be analyzed by training the model with and without the radiological variables and then comparing the performances.

2. Are the radiomic features helpful?

    - Similarly to the point above, we can train models with and without radiomic features in order to gauge whether the features are impactful.

3. Do the radiomic features actually influence the decisions of the model?

    - Even if the performance improves when radiomic features are included, it is conceivable that the improvement could be a result of random variations rather than added information (depending on what model is used and how it is trained). To properly gauge whether the radiomic features add value, we must also investigate whether they influence the decisions. This can be done by analyzing their importance, with can be measured for example by how much the prediction changes when the feature's value changes, or by calculating their SHAP (SHapley Additive exPlanations) values[96][*]. The SHAP values have an added benefit in that groups of features can be analyzed simultaneously—if feature A influences the prediction on a patient positively, and feature B influences it negatively, their net contribution can be zero, thus having no cumulative influence.

4. On what variables does the model base its decisions?

[*] SHAP values can be viewed as a local measure of feature importance (i.e. it can be calculated for each patient individually). At a high level, it measures how the prediction changes when the feature is *not* included. Crucially, the SHAP values differ from feature importance in that they measure directionality; features can influence a decision both positively and negatively. The values can be converted to a global measure by taking the mean absolute value over all patients.

- The feature importance analysis described above can also be carried out for clinical and radiological features. By inspecting these importances for different target variables, we can get a good idea of the model's tendency to use the information clinicians are already familiar with (such as the ISUP grade and PI-RADS score). For example, a model predicting survival ought to use the age variable at least to a moderate degree, since old patients are likely to die from age-related diseases (assuming the cohort includes predominantly senior citizens).

5. Is there a performance difference between the low-risk and high-risk patients?

   - In clinics, there is a large discrepancy between the assessment certainty of a high-risk and a low-risk patient. Low-risk patients have smaller and less developed lesions, which makes it harder to assess the images visually and makes the biopsy less likely to hit the pathological tissue. As such, AI can have a greater impact when applied to low-risk patients. It is therefore interesting to analyze if the AI's performance on this subgroup of patients is better, worse, or unaffected. To do this, we divided the patients by risk class (low/favorable/unfavorable/high) and compared the performance between the different groups.

6. Does radiomics play a specific role for low-risk patients? What about high-risk patients?

   - Even if one observes that the radiomic features are helpful in improving the performance (question 2) and that they actively influence the decisions (question 3), it may be the case that the features' contributions are exclusive to a specific sub-group (e.g. high-risk patients). In other words, it is possible to detect a significant influence from the radiomic features even though they contribute nothing to patients within the low-risk group. If so, the features would behave akin to the radiological variables. To analyze the radiomic features' contribution within different groups of patients, we calculated their cumulative SHAP values and compared the magnitude to the cumulative SHAP values of the other variables. The values were normalized to get a percentage contribution.

7. Is the ML model better at predicting the pathological ISUP, T, and N variables than a naïve model? (A naïve model uses the pre-surgery value as the prediction.)

    - The pathological variables are not explicitly predicted in the current medical practice, and it is therefore not possible to compare the AI model to the clinical workflow. However, using the naïve model is a reasonable assumption for the best pre-surgery prediction. By comparing how much better/worse the AI model is compared to the naive model, we can get a rough estimate of how useful the AI might be. This can only be done for ISUP, T, and N because the other variables have no pre-surgery equivalent.

With the above goals in mind, we expanded the CatBoost training procedure outlined in Section 4.3.3 to also incorporate model variants trained with 1) only clinical variables, 2) clinical variables and radiological variables, 3) clinical variables and radiomic features, and 4) all variables (clinical, radiological, and radiomic). Figure 4.9 presents an overview of the study design. For the relevant model variants, we collected the feature importances (measured in terms of prediction value change) and each feature's SHAP values.

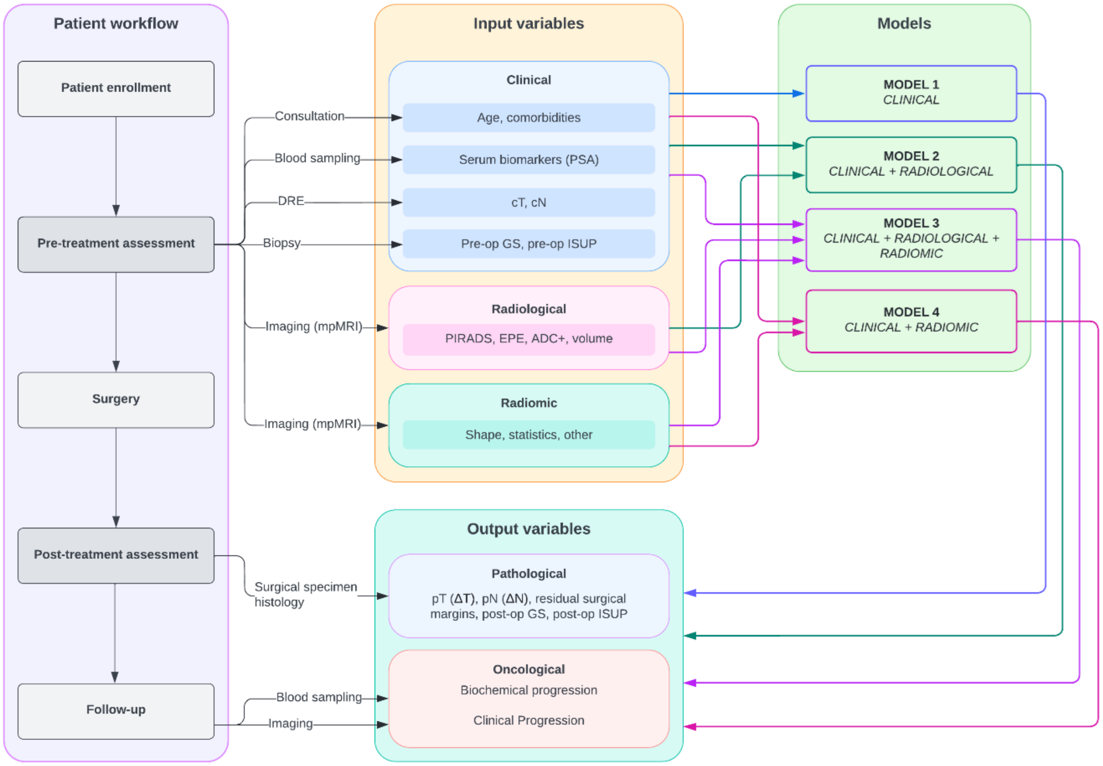


Figure 4.9: Overview of the model evaluation study. After patients are enrolled, clinical variables like the clinical tumor stage (cT) and the pre-operation ISUP grade are collected in the consultation, blood sampling, digital rectal exam (DRE), and biopsy stages. Radiological and radiomic variables are judged/extracted from the mpMRI scan of the patients (here the T2-w images). Four different CatBoost models are built with different combinations of the collected variables. Finally, pathological and oncological variables are predicted with histology and follow-up blood tests and imaging as confirmation.

4.4.2 Results

To answer questions 1 and 2 (How influential are the radiological variables? Are the radiomic features helpful?), Figure 4.10 shows the performance of models 1-4 and Figure 4.11 shows their relative mean ranks. We can gain insights into question 1 by comparing the clinical model (blue) with the clinical + radiological model (orange), and question 2 by comparing the clinical model (blue) with the clinical + radiomic model (green) or the clinical + radiological model (orange) with the all-model (red). A salient result is that the radiological variables improve the performance by a substantial margin (blue vs. orange). The radiomic variables appear to improve the performance by a small margin (blue vs. green and orange vs. red), but not in all cases (decrease for surgical margin, and unchanged for ΔT in terms of the MCC). However, there is a considerable difference between the blue vs. green comparison and the orange vs. red comparison. Comparing model 2 (orange) and model 3 (green), it appears that the radiomic variables do not fully encompass the useful information within the radiological variables. The overall trends are similar for all scores (Figure 4.11) but slightly different in accuracy, where the clinical-only model achieves the best score for two targets (Pathological N and ΔN).

In Figure 4.12, the cumulative feature importance of different groups of variables (clinical, radiological, and radiomics) in models 2, 3, and 4 are presented, which provides insights into question 3 (Do the radiomic features actually influence the decisions of the model?) and partly question 4 (On what variables does the model base its decisions?). Model 4 (red bars in Figure 4.10, right panel in Figure 4.12) uses radiomic features to a moderate degree (save for the ΔT target), which gives assurance that the benefit from radiomics observed above is not a coincidence. The minuscule feature importance (2%) of the radiomic fea-

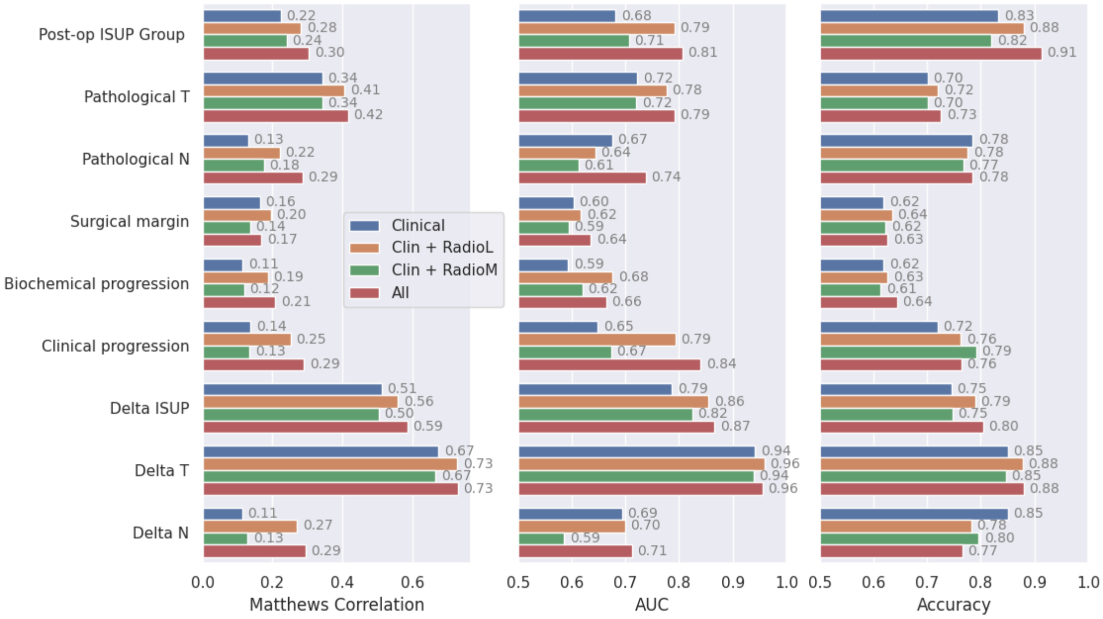


Figure 4.10: Performance of the CatBoost model when trained with 1) only clinical variables, 2) clinical + radiological variables, 3) clinical + radiomic features, and 4) all variables.

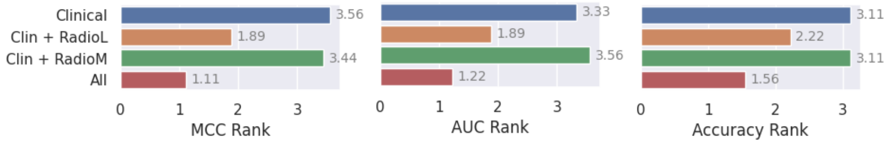


Figure 4.11: Relative mean ranks of the models in Figure 4.10.

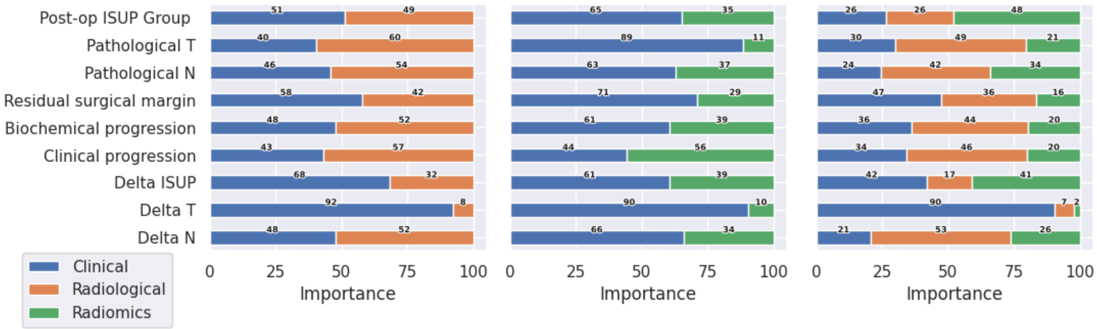


Figure 4.12: Cumulative importance of different groups of features when training model 2 (left), model 3 (middle), and model 4 (right).

tures in predicting ΔT with model 4 is in agreement with the performance, which was unchanged by adding radiomic features (orange vs. red bars in Figure 4.10). By comparing model 2 (clinical + radiological) and model 3 (clinical + radiomics), we see that the clinical variables have a larger overall contribution in model 3, consistent with the fact that the radiological variables improved the performance more (thus having larger importance).

By inspecting the importance of individual features, we get answers to question 4 (On what variables does the model base its decisions?). In this context, the clinical and radiological variables are primarily the ones of interest, because we can readily compare them with how important they are for doctors in the current medical practice. Figure 4.13 shows model 4's importance of the five most important clinical and radiological variables and their cumulative importance. EPE (within top-5 for all targets, most important feature for five targets) and PI-RADS (within top-5 for seven targets, most important twice) stand out as recurringly important variables, consistent with the clinical expectation. The clinical T is a powerful predictor for ΔT, which explains the low feature importance of the radiological and radiomics variables apparent in Figure 4.12.

In answering question 5 (Is there a performance difference between the low-risk and

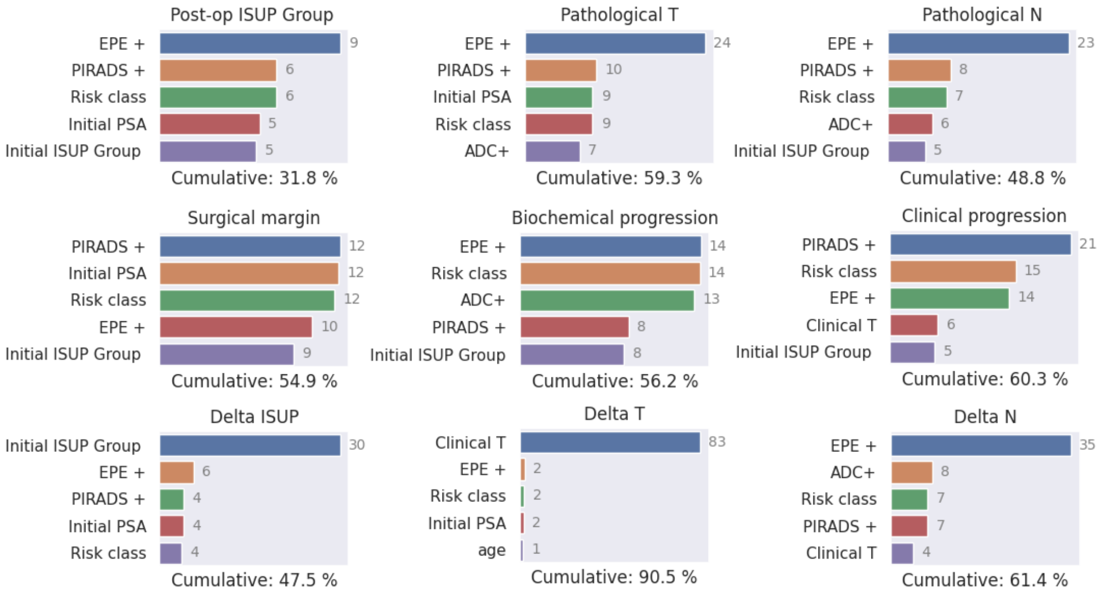


Figure 4.13: Top-five feature importance among clinical and radiological variables when training model 4 (all variables), along with their cumulative importance.

high-risk patients?), the division of patients into different risk groups introduces a target-distribution imbalance (see Figure B.1), which renders the above scores (MCC, AUC, and accuracy) inappropriate. To compensate, we measured the mean absolute error (MAE) instead, which can be calculated and compared individually for each patient. Figure 4.14 presents the distribution of MAE values for patients in the different risk groups. In all targets except pathological T, the median absolute error is lower in the low-risk group, sometimes by a large margin. The distribution is significantly wider within the high-risk group than in the low-risk group (for post-op ISUP, pathological N, surgical margin, and clinical progression) or similar (for the remaining targets).

For question 6 (Does radiomics play a specific role for low-risk patients? What about high-risk patients?), the radiomic features' normalized SHAP contribution within low- and high-risk patients are presented in Figure 4.15. These results make it clear that the radiomic

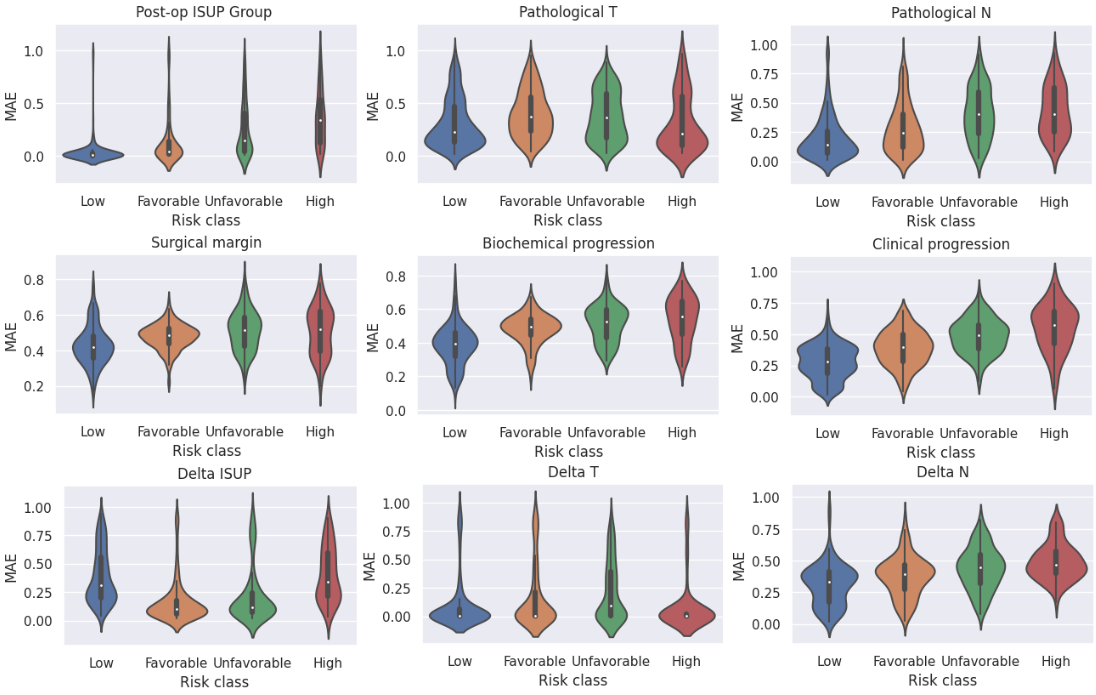


Figure 4.14: Distribution of MAE values for patients within different risk groups. The width of the violins is proportional to the number of patients at different y-values, and a white dot at the center indicates the median.

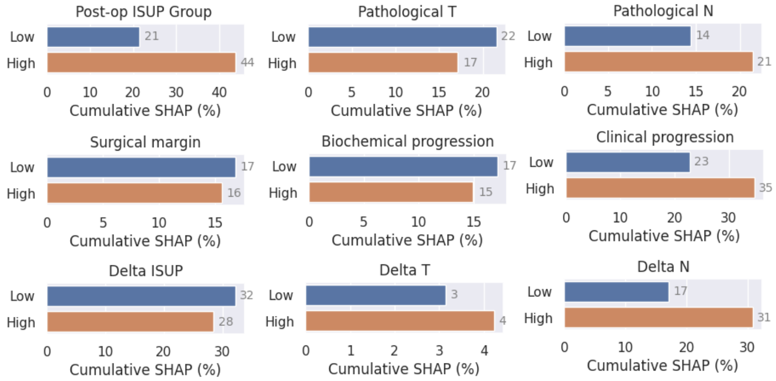


Figure 4.15: Percentage contribution of radiomic features in terms of their absolute cumulative SHAP value. The bars indicate the total cumulative contribution of *all* radiomic features, meaning that feature interactions are taken into account (if feature A influences the decision negatively, and feature B positively, their net contribution may cancel out, totaling a contribution of zero).

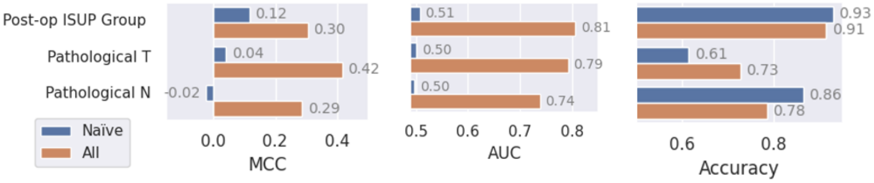


Figure 4.16: Performance comparison between a naive model (one that uses the clinical values as predictions for the pathological values) and the best ML model (trained with all variables, i.e. model 4). Since the naive model does not output probability estimates, the AUC is somewhat one-sided.

features are in fact used in both the low-risk and the high-risk groups, although the contribution generally seems somewhat larger within the high-risk group.

The results from comparing the naive model and the radiomic model (question 7) are shown in Figure 4.16. The best ML model (model 4 above, which was trained with all variables) clearly outperforms the naive model significantly in both MCC and AUC. However, the naive model has better accuracy for two of the targets.

4.5 Discussion

Overview

Pathological prediction models could constitute a great asset to the clinical workflow since they could reduce under- and overtreatment by providing clinicians with pathological information prior to surgery. Such models can estimate the degree of malignancy even when the standard assessment is not possible, for example in patients with comorbidities. Here, we showed that the pathological characteristics of prostate cancer can indeed be predicted successfully using ML models, particularly with GBDTs. The results demonstrate that whole-prostate radiomics can offer a small improvement over models using only clinical and radiological variables.

In the performance comparison between different models, the CatBoost model performed consistently well and was the model that most often achieved the best score. Of the deep learning models, the multitask 1DCNN was generally the best, followed by the FT-Transformer. There appears to be a tendency for the multitask models to perform better in terms of accuracy than in terms of the other metrics, which may indicate that there is a slight tradeoff between MCC/AUC and accuracy. This is also supported by observing that the MLP generally performed much better in terms of accuracy than the other two scores. However, it is likely that the MLP is not inherently better at achieving higher accuracy, and that CatBoost would still be the best model if we decided to optimize for accuracy instead. It is conceivable that the somewhat subpar performance of the DL models could be due to insufficient training data since DL models are known to be data-hungry. A potential solution for this that was not studied in this article is data augmentation, which is standard practice for most non-tabular DL models.

When further analyzing the CatBoost model, it is clear that the best performance is consistently achieved by the model trained on all variables (clinical, radiological, and radiomics). The experiments also demonstrated that the radiomic features often contributed significantly (between 16% and 41% if we disregard the highest and lowest values) to the predictions and that the model tends to rely on the same variables used in clinics such as PI-RADS and EPE (Figure 4.13). Since the pathological variables are currently not explicitly predicted in clinical practice, it is not possible to compare the ML models against the current workflow. However, when comparing ML against naïve models to detect the differences between the clinical and pathological statuses (Figure 4.16), the improvement is profound. The model's confidence in its predictions was slightly higher for low-risk patients than for high-risk, which is contrary to the experience of clinicians, possibly indicating an area where ML models could have an elevated impact. On the other hand, this may be a consequence of the fact that the low-risk group generally contained fewer positive cases (see Figure B.1). In general, the methods highlighted here represent an excellent tool to analyze the behavior and gain insights into ML models intended for clinical care.

Data

This study was performed on an extensive dataset of 949 patients, the largest cohort to date (the previous one being 489 patients) according to a recent review covering 57 different prostate cancer radiomics studies[37]. These results demonstrate that T2-only radiomics might be a promising approach moving forward, although a head-to-head comparison with models that include the other modalities (DWI and ADC) is necessary. Since the PI-RADS evaluation performed by the clinicians is partially based on the ADC images, the information in these is not entirely lost by using only T2 radiomic features.

Even though the experiments here were carried out by binarizing the target variables, both the DL and the GBDT models can be modified to perform regression or multi-class classification by simply changing the loss function (and the output layer for the DL models). The motivation behind making the binarization was to simplify the comparison between different target variables and to simplify the training procedure. It is likely that the overall results would be similar for the other tasks as well since the training procedure is identical.

The automatic segmentation procedure that was applied in this study appears to produce viable contours that can be used for analysis and feature extraction, further validating the use of automatic segmentation in clinics. When using such contours, it is essential to employ quality checks such as the one outlined in Section 4.2 in order to detect critical failures. In our case, this enabled us to detect and revise a few cases where the algorithm erroneously delineated most of the image as prostate tissue. It should be noted that many of the contours flagged for correction did not need any modifications.

Classification performance

When contrasting our results to other studies comparing the performance of clinical and radiomic models, the results here appear much more conservative. In our results, radiomics offered an improvement of just 0.02 units of MCC and 0.022 units of AUC on average amongst all targets. By contrast, it is not uncommon to see AUC improvements of 0.2-0.3 in the medical literature. It is not possible to conclusively identify the source of this discrepancy without performing extensive experiments, but a few potential factors can be identified. The T2-only approach, the whole-prostate radiomics extraction, and the automatic segmentation may all influence the radiomic performance negatively compared to their alternatives. However, we believe the primary cause to be the larger dataset and more rigorous validation procedure. On a smaller dataset, it is much easier to overfit and overestimate the performance. Furthermore, nested cross-validation (and sometimes even single cross-validation) is not performed in most other studies. This has become a problem to such an extent that some researchers have argued that there is a widespread overfitting epidemic in the healthcare domain, supported by several independent meta-analyses and discussions[135,25,153,72,112,54].

Comparing the results on different target variables reveals a sizeable difference, suggesting that some targets are more easily predicted than others. While it may be interesting to look at these differences, comparing target variables head-on might not be fair for targets with different distributions of missing data and positive/negative cases. The two variables having much better performance (ΔISUP and ΔT) appear to be more easily predicted because their initial values (initial ISUP group and clinical T) are very strong predictors (see Figure 4.12).

Feature selection is one of the most important aspects of model development because it dictates what information the model will have at its disposal. Despite this, it is not commonly discussed or researched within studies of medical prediction models. Since the procedure's outcome is heavily data and problem-dependent, it can be wise to explore the options when presented with a new problem or data set (though this requires additional effort). In most studies, features are selected prior to training (e.g. by a clustering procedure or statistical testing) or internally to the model (e.g. with LASSO), which is problematic in several ways. First, it does not properly account for feature-parameter interactions. Second, it may introduce leakage into the validation pipeline if not incorporated correctly (once for each training set). It is also susceptible to additional variance and lower performance. For these reasons, we instead incorporated the selection procedure into the parameter optimization (see Appendix B.3), which should generally be preferred if its implementation is possible.

A major concern in studies comparing the performance of different models is the variance and reproducibility of the results. In the model validation pipeline, there is a delicate balance between exploration and validation in the sense that finding higher-performing models via deeper exploration in the model space competes with performing more thorough validation procedures (such as repeated cross-validation). If the model space is not sufficiently explored, there will be large variations in the selected models (since there will be a sparser sampling of the error landscape). Conversely, if the models are not thoroughly validated (e.g. with repeated experiments with different random seeds), there may be large variations due to instabilities in the training data or model weights. Multiple sources of variations make it hard to analyze the performance with regular statistical tools without excessive amounts of compute. In the medical field, these instabilities are largely overlooked, even though the small datasets may exacerbate the problem. Due to limited computational resources, this study focused on finding high-performance models instead of repeated experiments, which is currently the standard practice. To ensure the reliability of this approach, we performed informal repeated tests (only on the first target variable due to the required computational investment) and concluded that this variability would not invalidate the results. It is worth noting that another way of improving the reliability of predictions is to create model ensembles, but this also requires an increased investment of computational resources.

Optimization and training

In this study, we chose to optimize the MCC because it is known to be a more reliable and informative measure of performance than other metrics like accuracy, F1-score, and AUC[13,22,23,24]. For example, the MCC only achieves a high value when the classifier produces good results in all four quadrants of the binary confusion matrix. In medical research, the AUC has been the standard reporting metric for prediction models, which in many ways can be seen as a cause for concern (see e.g. Lobo 2008[95], Halligan 2015[52], or Byrne 2016[15]). For instance, the definition of AUC allows classifiers to increase their AUC without modifying a single prediction since it considers all decision thresholds and not just the actual operating threshold. A similar mechanism also allows them to simultaneously increase their AUC and decrease their accuracy. This is not to say that AUC does not have its uses, but we believe these are legitimate reasons to not optimize for AUC directly. Accuracy is another natural candidate for optimization, but this metric can be misleading for unbalanced data. In preliminary experiments, we observed that optimizing either AUC or accuracy often leads to majority classifiers that classify all patients into the majority class, which are essentially useless in practice (they do not utilize any information available in the variables). When deploying prediction models in real clinical environments, it will be crucial to discuss and clarify if true/false positives and negatives should be weighted equally, and then optimize the appropriate metric.

When selecting features for the multitask models, we chose to select the feature with the highest mean predictive power over all different endpoints. An exciting alternative to this is to favor features with high predictive power for tasks the model struggles with, which would effectively act as an indirect loss-weighing for different targets. In loss-weighting, each task's loss function is weighted differently to optimize an aggregated loss function that may be more ideal. This of course raises the question of how exactly to weigh the different tasks, which is a difficult optimization problem in and of itself. Furthermore, the desired balance is heavily influenced by external factors such as preference and the individual samples being evaluated. We chose not to explore these considerations due to the foreseen complexity, but it is of interest for future research.

Interpretation

The analysis methods for the model's behavior and tendencies outlined here can be used as an effective interpretation tool that can reveal critical flaws that are not apparent from simply inspecting the performance. Such analyses will be required before ML models can be deployed in practice as they allow both users and administrators to learn when not to trust the model and what its strength and weaknesses are. While the feature importance measures provided by GBDT packages are exclusive to GBDT models, the other methods are model agnostic. Feature retraining, SHAP value analyses, and sub-group analyses such as those presented in Figures 4.12, 4.14, and 4.15 can all be applied to DL models and traditional models as well (the SHAP value may be interpreted as a measure of feature importance). The calibration analysis in Figure 4.14 can be used with all models that provide probability estimates. Many other useful methods exist such as ICE (individual conditional expectation) plots and PDPs (partial dependence plots), which can be used to analyze the response when individual features of interest are changing. For DL models specifically, a plethora of different methods has been developed to give users explanations such as saliency maps and feature visualizations.

5 Conclusion

In this thesis, we have explored various applications of deep learning and radiomics for the assessment of clinically relevant prostate cancer. We showed that automatic segmentation models can achieve excellent performance with very few training samples with clever use of extensive data augmentation. The mixup augmentation technique appeared particularly useful when training segmentation models on the small scale typical for medical applications. Furthermore, we introduced a novel method to automatically predict the quality of contours with deep learning that can be used as a safeguard when deploying segmentation models. In Chapter 3, we discussed the issue of image normalization and demonstrated that the choice of normalization can significantly impact the calculation of the radiomic features. Finally, the practical utility of the automatic segmentation methods was demonstrated in predicting pathological endpoints of interest. On an unprecedentedly large dataset of 949 patients, we showed that the pathological status of prostate cancer patients can be reliably predicted by ML models, particularly using GBDTs. We also presented a framework for analyzing prediction models' behavior that can be used as an interpretation tool before deploying models in the real world. In summary, the methods outlined in this thesis demonstrate how DL and ML can successfully be combined with radiomics and integrated into the pathway of the patient, from image acquisition to treatment decision.

It is clear that AI has a vast potential to improve and enhance the clinical care process in myriad ways. But before AI can become a reality in clinics, we need to carefully address the available options and their strengths and weaknesses. This includes different types of models, how they behave in different situations, and how we should interpret and handle their predictions and decisions. It will be important to approach the challenges with an open mind and communicate what the models can and cannot do. Likewise, users, developers, and administrators need to engage in conversation about how to solve the problems collectively. Following these principles will ensure that the ensuing integration of AI will be beneficial for everyone involved, from patients and doctors to developers and administrators.

A

Supplementary information for Chapter 2

A.1 Association between clinical variables and segmentation results

The analysis in this appendix covers two aspects of the relationship between the clinical variables and the segmentation performance. First, pure associations (correlation for continuous variables and the Kruskal-Wallis test for categorical variables) can reveal confounding factors that may influence the segmentation performance. Second, the CatBoost prediction model that was trained to predict the segmentation performance from clinical variables can be analyzed in terms of its accuracy and feature importance. The motivation for carrying out these analyses is to find factors that could serve to indicate whether a given prostate is suitable for automatic segmentation.

In the first analysis, no significant associations ($p \geq 0.08$) between any of the categorical clinical variables (ISUP grade, EPE score, and PI-RADS) and segmentation performance were found (see Figure A.1). Out of the continuous clinical variables (prostate volume, age, and iPSA), prostate volume and age were found to be significantly associated with performance, albeit only in terms of MSD (see Figure A.2). In both cases, the relationship was weak: Spearman rank correlation ρ=0.33 for volume (p=0.021), and ρ=0.29 for age (p=0.040). The lack of any strong relationship consistent for both Dice and MSD suggests that the segmentation performance is not strongly influenced by single clinical parameters.

The CatBoost prediction model did not perform better than the naïve model on average (the MAE was 0.016 in both models; see Table 2.3), suggesting that clinical variables alone are not indicative of segmentation performance. A plot of the predicted Dice values is shown in Figure A.3. There was a slight negative correlation of -0.155 but the relationship was not significant ($p = 0.54$). The most important features were iPSA (68%) and volume (21%), but since the overall performance was low, this is unlikely to indicate a significant underlying relationship.

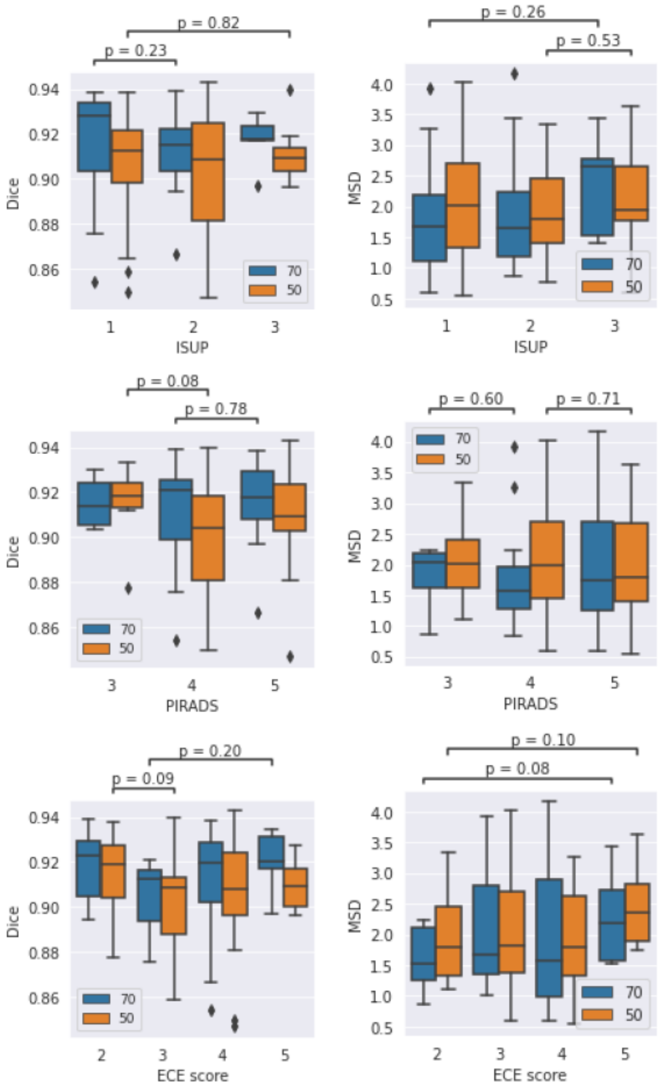


Figure A.1: Association between clinical categorical variables and segmentation performance (Kruskal-Wallis test). Only the lowest p-values for each variable and dataset split (blue: 70/30, orange: 50/50) are shown. No significant associations were found.

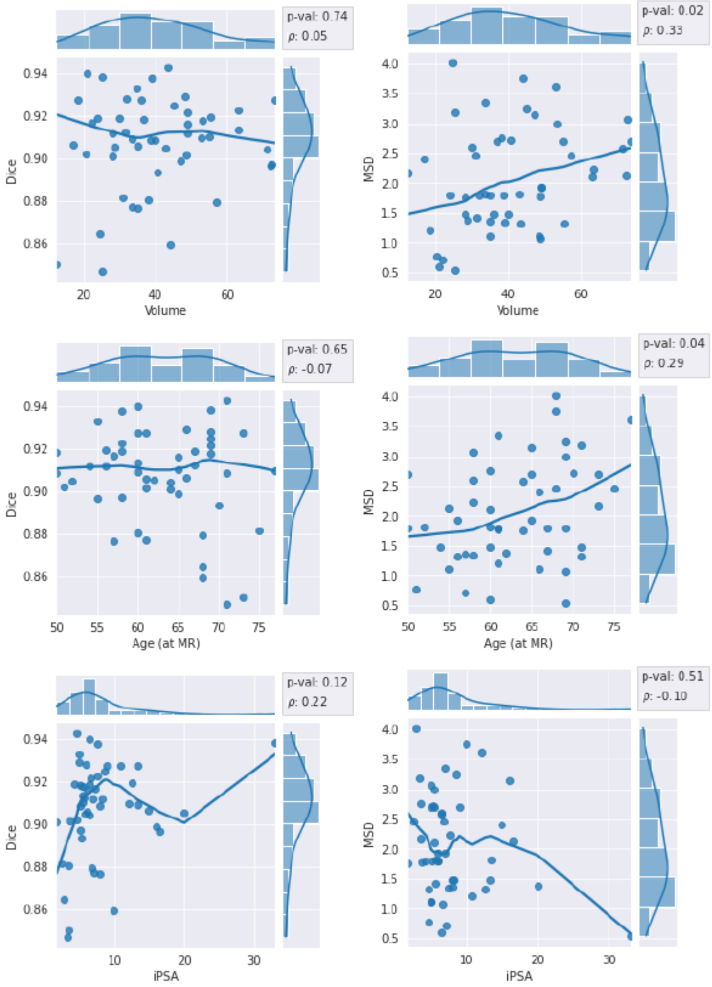


Figure A.2: Association between continuous clinical variables and segmentation performance in terms of Spearman rank correlation. Left column: Dice coefficient, right column: mean surface distance. The trend line is a non-parametric lowess regression smoother (for visual aid only).

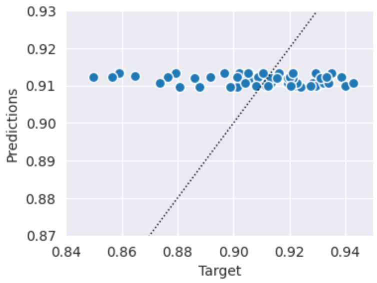


Figure A.3: Predicted vs. target Dice values of the baseline CatBoost model, which only uses clinical variables to predict segmentation quality. The dotted line indicates perfect $x = y$ predictions. The predictions of this model tend to only vary minimally (very close to the naïve model), suggesting that the clinical variables are not indicative of segmentation performance.

B

Supplementary information for Chapter 4

B.1 Additional patient details

Table B.1 displays the clinical characteristics of the dataset and the number of patients with different conditions. Figure B.1 shows the distribution of positive and negative patients within different risk groups, illustrating the large class imbalances within some target variables.

Table B.1: Clinical characteristics of the dataset. The column headers indicate the stage/group/number (e.g. ISUP grade group 1, 2, 3, etc.), and the numbers in each cell represent how many patients there are within that particular category. For binary variables (e.g. biochemical progression) a zero indicates absence and a one indicates the presence of the condition. For risk class, the numbers 0-3 indicate low, favorable, unfavorable, and high risk, respectively. # DIL: number of dominant lesions, T: tumor stage, N: lymph node status.

	0	1	2	3	4	5	Missing
Comorbidity	541	406					2
# DIL		310	536	93	5	1	4
PI-RADS +		2	48	163	380	355	1
EPE +	3	80	232	242	227	133	32
Initial ISUP Group		461	328	157	1		2
Clinical T		559	324	62			4
Clinical N	943	3					3
Risk class	389	277	201	81			1
Post-op ISUP Group		207	437	233	37	31	4
Pathological T			582	367			
Pathological N	495	76					78
Surgical margin	716	232					1
Biochemical progression	637	140					172
Clinical progression	726	50					173

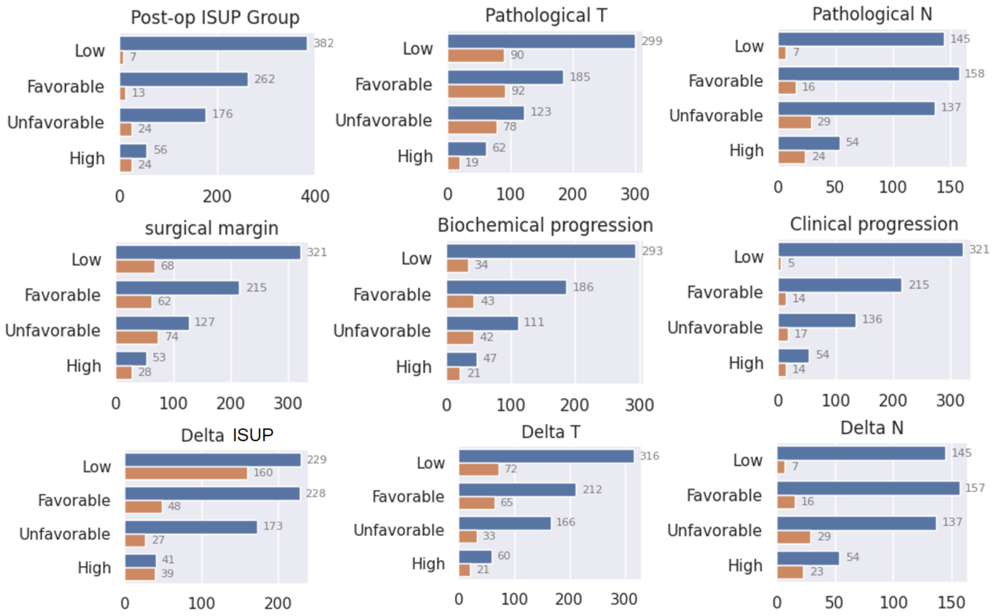


Figure B.1: Number of positive and negative patients (after binarizing the target variables) within different risk groups. In some cases, there are extreme class imbalances.

B.2 Additional training details

B.2.1 Imputation

Imputation of missing variables is needed to train models that do not inherently handle missing values such as DL models. Naturally, the impact of imputation depends on the distribution of missing values within the data, which makes it hard to assess its importance and downstream effects. Some guidelines have been suggested such as not imputing a variable if more than 40% of the data are missing (in such a case, excluding the variable may be more appropriate)[73], or when the statistical assumptions of the particular imputation method do not hold (e.g. if the data are not randomly missing, but the result of a systematic error)[134]. Clinical studies typically do not discuss this issue and tend to deploy simple

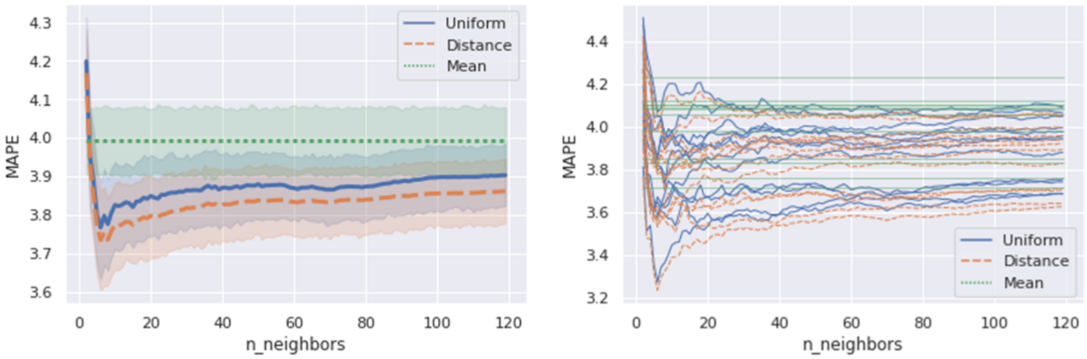


Figure B.2: Mean absolute percentage error (MAPE) of different imputation strategies with varying neighbors: a kNN imputer with uniform weights, a kNN imputer with distance-based weights, and a simple mean imputer. The best performance was generated by a distance-weighted kNN imputer with eight neighbors.

and easy imputation strategies such as filling missing data with the average or median value. However, it seems unwise from an information perspective to not use the available information when making such inferences.

Here, we constructed a simple test where we compared the performance of two promising imputation methods with varying parameters: one k-nearest neighbor (kNN) imputer[111] and one simple mean imputer. The kNN imputer uses a predetermined number of neighbors ("n_neighbors") and weighs them either uniformly or according to their distance from the sample being imputed. We removed random entries from the data following the observed missing data distribution (see Table 4.1) and performed imputation twelve repeated times. The different imputation methods were evaluated based on the mean absolute percentage error (see Figure B.2). Following these results, we selected the distance-weighted kNN-imputer with eight neighbors.

B.2.2 Learning parameters & hyperparameter search spaces

This section covers the learning parameters and the search spaces that were used in the Optuna hyperparameter search for the different models. For the DL models, we complemented the automatic search by manually tuning the learning rate, batch size, and number of epochs. The learning rate was tuned with a special heuristic that is covered in section B.2.3.

The number of epochs was selected by observing the point at which the validation loss ceased to decrease using each model's default parameters. We used 40% fewer epochs (roughly coinciding with the start of the loss plateau) in the Optuna search to accelerate the search process. The batch size was set to the largest batch that could fit into memory without splitting the training set unevenly (in this case 384). All DL models used the Adam optimizer with default parameters.

The search space for the CatBoost model is shown in Table B.2 and the DL models' search spaces are shown in in Table B.2. Note that the number of radiomic features to include is optimized in all cases (see Appendix B.3 for details). The MLP optimizes the number of hidden units and the dropout rate in each of the two layers. The FTT optimizes the token dimension for each feature, the dimension (i.e. the number of hidden units) and the dropout rate in the feed-forward part (ffn) of the transformer blocks, the dropout rate in the attention mechanism, and the number of transformer blocks. The 1DCNN optimizes the dimension of the initial dense up-scaling layer, the number of channels the output of this layer is split into ("Reshape" in Figure 4.2), the number of channels in the first three convolution layers, the number of channels in the last convolution layer, and the dropout rate (we used the same dropout rate throughout the whole network to narrow the search

Table B.2: Hyperparameter search space for the CatBoost model. Parentheses and brackets indicate continuous and integer search ranges, respectively.

Parameter	Value
bootstrap_type	'MVS', 'Bernoulli'
n_estimators	[2, 600]
max_depth	[2, 8]
l2_leaf_reg	$(1^{-4}, 30)$ (log-uniform)
random_strength	(0, 6)
min_data_in_leaf	[1, 100]
subsample	(0.1, 1)
colsample_bylevel	(0.1, 1)
auto_class_weights	'None', 'Balanced'
n_features	[0, 32]

Table B.3: Hyperparameter search space for the MLP, FTT, and 1DCNN deep learning models. Parentheses and brackets indicate continuous and integer search ranges, respectively. The third number (if any) within the brackets indicates the step size discretization.

MLP		FTT		1DCNN	
param	**value**	**param**	**value**	**param**	**value**
n_units 1	[32, 2000]	d_token	[8, 512, 8]	upscale size	[256, 4096, 32]
n_units 2	[32, 2000]	d_ffn	[8, 256, 8]	n_channels 1	[8, 512, 8]
dropout 1	(0, 0.5)	dropout_fnn	(0, 0.5)	n_channels 2	[8, 512, 8]
dropout 2	(0, 0.5)	dropout_att.	(0, 0.5)	n_channels 3	[8, 512, 8]
n_features	[0, 32]	n_blocks	[1, 5]	dropout	(0, 0.5)
		n_features	[0, 32]	n_features	[0, 32]

space).

The only changes made to the FTT and 1DCNN multitask models were to change the number of output units in the final layer to one. In the multitask MLP model, we added an additional layer and changed the search range for the number of hidden units in each layer to [32, 1024] (we observed that this deeper network was beneficial in the multitask scenario).

As for the multitask-tailored MLP, the core architecture was fixed throughout the training pipeline. The dropout rate and the number of hidden units in the layers were tuned with a grid search covering dropout rates between 0 and 0.5, and hidden units from 100 to 1000. In contrast to the other models, this one was trained with a focal loss with gamma=4 and instead of binarizing all targets, it was trained to regress the non-binary outcomes directly. Furthermore, the model relies on implicit feature selection and L2 regularization instead of the feature selection procedure described above. These choices were based upon prior work with similar multitask models.

B.2.3 Learning rate

The learning rate was selected using FastAI's LR-finder heuristic—a strategy in which one plots the training loss against the learning rate after iteratively increasing the learning rate for a set number of steps while training. The result is typically a curve that starts flat (the network fails to learn anything at very low learning rates) and then decreases quickly and finally succumbs to pure noise or a giant spike (the network fails to consistently reduce the loss at extreme learning rates). See Figure B.3 for examples from our models. The heuristic is to select the learning rate with the steepest decline, which is motivated by the assumption that a steep decline in the loss corresponds to a learning rate regime in which the network learns well. In especially noisy cases and in cases where the largest gradient was not evident, we sampled multiple learning rates and chose the best one. This procedure was done on the full data set with the models' default parameters.

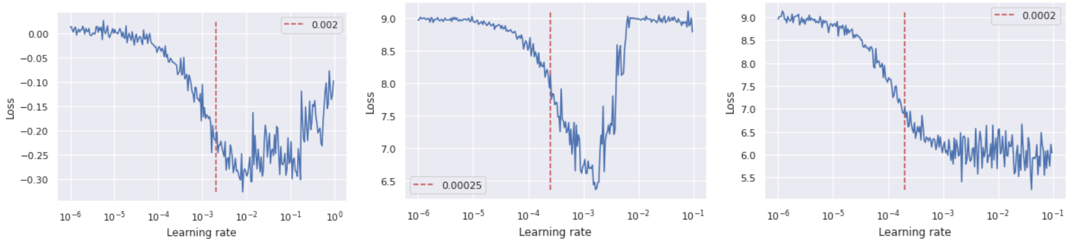


Figure B.3: Examples of curves from the LR-finder procedure and selected learning rates (vertical dashed line) for the MLP (left), FTT (middle), and 1DCNN (right) models. The shape and behavior of the curves vary from target to target—these examples were cherry-picked for the purpose of illustration.

B.3 Feature selection

This appendix gives a brief account of the feature selection issue. Despite its importance, it is not commonly addressed in ML studies for healthcare. First, we introduce the core problems and some critical aspects of common selection strategies. In section B.3.2 we describe the method that was used in this study in more detail. Finally, we present tentative experiments we performed to compare selection strategies.

B.3.1 Background

Most modern prediction models incorporate some sort of implicit feature attention/selection mechanism (e.g. L1 penalty in traditional models like LASSO, the weights & biases in DL models, or the splitting criterion in decision trees), but these mechanisms alone appear not to achieve optimal selection since prior feature selection can consistently improve the performance. It is sometimes argued that a good prediction model should be able to learn what information to rely on and what to ignore, which is partially supported by theory, but it appears that this does not sufficiently address problems like overfitting and the curse of dimensionality in practice. For example, virtually all radiomic studies incorporate some prior selection procedure, and not a single high-performing Kaggle submission uses unprocessed data.

An additional obstacle is that one needs to decide not only how to select the features, but also how many to select. Sometimes this number can be chosen based on statistical arguments like the "one in ten rule", which says that one predictor variable can be included for every ten events, but such arguments fail to account for the fact that e.g. deep learning models, which have been dominating many fields lately, are extraordinarily overparameterized and often use tens of thousands of features. Furthermore, the rule is just a heuristic—there is no guarantee that it should work well, so more rigorous or data-driven approaches should be used whenever possible. For this reason, we chose to include the number of selected features in the parameter search, which is a relatively easy way to support the choice based on empirical evidence. The downside is that it increases the dimensionality of the search space and incurs a slight additional computational burden (depending on what methods are used). The latter problem can sometimes be alleviated by pre-calculating the quantities prior to the validation loop such that they are not re-calculated every time a new set of parameters is evaluated.

Many feature selection methods rely on correlation as a criterion to search for redundancy, for instance by clustering features based on their correlation patterns. However, correlation is limited to linear (in the case of Pearson correlation) or monotonic (in the case of Spearman correlation) relationships. Moreover, correlation is symmetric (the correlation between A and B is equal to the correlation between B and A), which is a property that is often not desirable. Mutual information is sometimes used to circumvent the former problem, but when asymmetry is desired, KL-divergence is more appropriate. However,

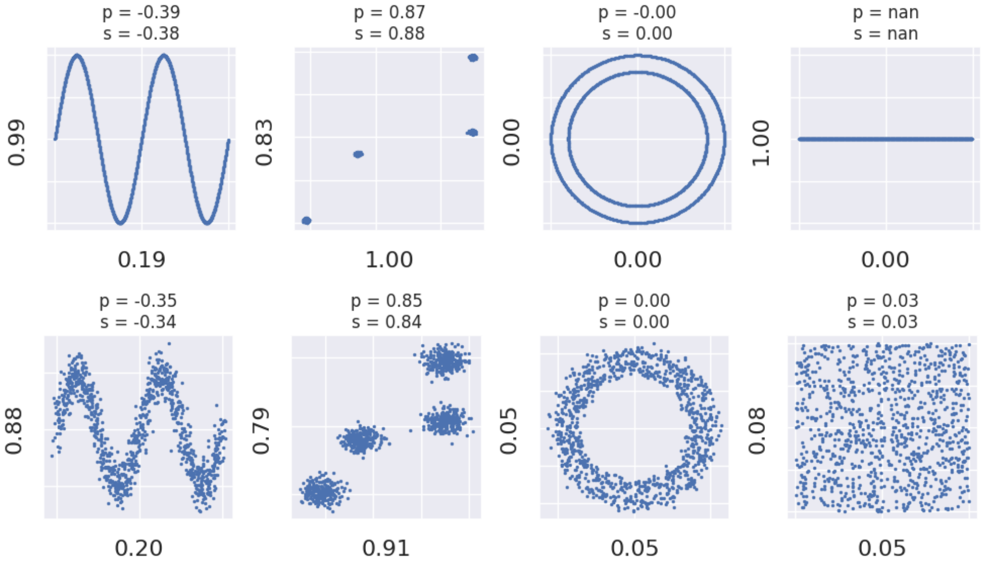


Figure B.4: Demonstration of the behaviors of correlation and predictive power in different scenarios. The Pearson and Spearman correlation in each context is indicated by the p and the s above each subfigure. The y-label indicates the predictive power the x-variable has over y, and the x-label indicates the predictive power the y-variable has over x.

while a high correlation or KL-divergence implies a high predictive power, the converse is not true—it is possible to have a very strong predictive power and yet zero correlation. To illustrate this, Figure B.4 shows the behavior of correlation and our implementation of predictive power in different scenarios. The symmetry difference is very pronounced in some cases, which is a crucial property in many real-world contexts. For instance, zip code is a very strong predictor of household income, but household income does not tell us much about where in the world the household is located.

B.3.2 Predictive power selection method

The core principle in this method was to select features based on their "predictive power" over the target variable—if feature A is better than feature B at predicting the target in a univariate test, feature A would be chosen over feature B. Thus, this selection can account for both nonlinear relationships and directional variation (unlike e.g. correlation). Then, for a given *n* (which is one of the parameters being optimized), we include the top-*n* best features in terms of their estimated predictive power.

The predictive power was estimated by the training loss of a shallow univariate LightGBM regressor trained to predict the targets within the training set with the given feature as the only input. The parameters of the regressor was n_estimators=192, learning_rate=1, max_depth=3, and min_child_samples=10 (all other parameters were set to their default value). This shallow structure prevents excessive overfitting and is extremely fast to train. We observed that this selection performed better than other common alternatives such as correlation clustering or recursive feature elimination.

After selecting the set of *n* radiomic features, the total input to the model was the chosen features plus the clinical variables. This way, the models can learn to not use the radiomic features when these are not related to the outcome. All clinical variables were included and fixed throughout the selection procedure. This ensures that clinically relevant information is always available and allows us to directly gauge the variables' contribution, which may be used as a sanity check and quasi-interpretation.

B.3.3 Experiments

To estimate how different feature selection methods compare against the predictive power selection strategy, we performed a set of tentative experiments. Since this was not the primary focus of the study, we did not devote sufficient resources to conclusively determine the best method. The primary goal of these explorations was to identify if any of the methods were clear outliers. A full-scale study examining these methods in detail is warranted, but would ideally require multiple datasets and many repeated experiments.

The following methods were tested in addition to the predictive power method mentioned above:

1. A correlation clustering method that clusters the features into *n* clusters and selects one representative feature from each cluster. The representative feature was selected by maximum mutual information with the target variable within the training set (for a deeper overview of mutual information compares with correlation, see Ince 2017[64]). The clustering was performed with a *k*-means clustering on the absolute Spearman correlations.

2. A PCA (principal component analysis) method that transforms the features set into its principal components and uses the *n* leading components as features.

3. A recursive feature elimination strategy where the model is first trained on all features simultaneously, and then re-trained on just the features with more than 1% feature importance (measured internally by CatBoost). The retraining was repeated until all features had more than 1% importance or until three models had been trained.

4. A CatBoost selection procedure where the *n* most important features in the CatBoost model were selected. This procedure was only tested on DL models since this is where it integrates most naturally (in the CatBoost case, it is equivalent to using a select-*n*-best procedure with one iteration).

Method 1 was compared against predictive power selection on all available endpoints, while methods 2 and 3 were compared in terms of four repeated measurements on the

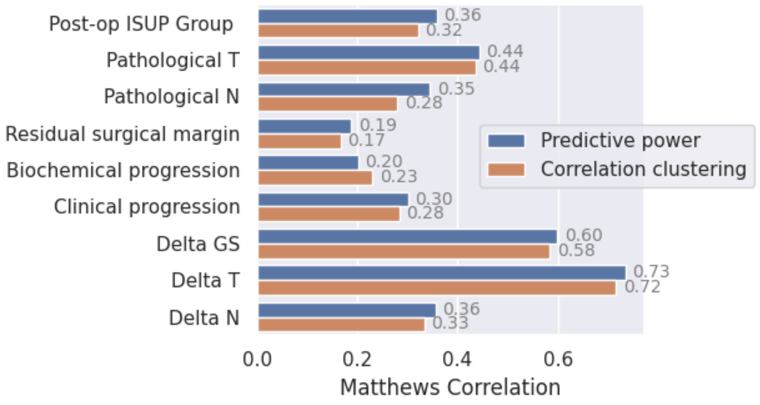


Figure B.5: Performance of feature selection by predictive power and by correlation clustering in terms of Mathews correlation coefficient on the different targets.

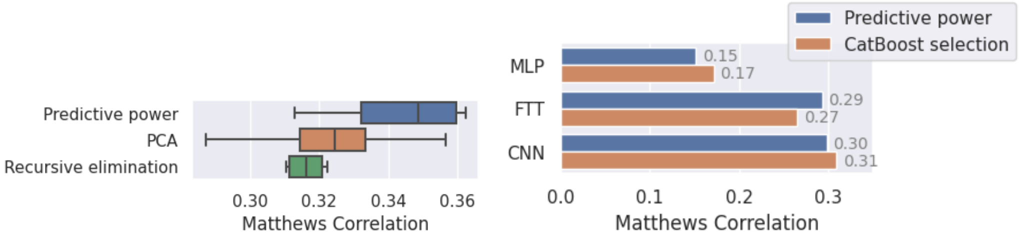


Figure B.6: Left: performance of the predictive power, PCA, and recursive elimination feature selection methods on four repeated experiments on the ISUP target variable. Right: performance of DL models trained with predictive power and CatBoost feature selection on the ISUP target variable.

ISUP target variable. Method 4 was compared to predictive power selection on the three DL models exclusively: MLP, FTT, and 1DCNN.

Method 1 (correlation clustering) was worse than predictive power selection on eight out of the nine different target variables (Figure B.5), the exception being biochemical progression. Predictive power selection also outperformed PCA and recursive feature elimination on average (Figure B.6) over repeated measurements on the ISUP target. For the DL models, CatBoost feature selection outperformed predictive power selection when training the ML and the CNN, but not the FTT. With these results in mind, predictive power selection was deemed an appropriate choice for the analyses performed in the remainder of this thesis.

References

[1] Abdollahi, H., Mofid, B., Shiri, I., Razzaghdoust, A., Saadipoor, A., Mahdavi, A., Galandooz, H. M., & Mahdavi, S. R. (2019). Machine learning-based radiomic models to predict intensity-modulated radiation therapy response, Gleason score and stage in prostate cancer. *La radiologia medica*, 124(6), 555–567.

[2] Abutbul, A., Elidan, G., Katzir, L., & El-Yaniv, R. (2020). Dnf-net: A neural architecture for tabular data. *arXiv preprint arXiv:2006.06465*.

[3] Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 2623–2631).

[4] Akoglu, H. (2018). User's guide to correlation coefficients. *Turkish journal of emergency medicine*, 18(3), 91–93.

[5] Altman, M., Kavanaugh, J., Wooten, H., Green, O., DeWees, T., Gay, H., Thorstad, W., Li, H., & Mutic, S. (2015). A framework for automated contour quality assurance in radiation therapy including adaptive techniques. *Physics in Medicine & Biology*, 60(13), 5199.

[6] Ardakani, A. A., Bureau, N. J., Ciaccio, E. J., & Acharya, U. R. (2021). Interpretation of radiomics features: a pictorial review. *Computer Methods and Programs in Biomedicine*, (pp. 106609).

[7] Arik, S. Ö. & Pfister, T. (2021). Tabnet: Attentive interpretable tabular learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35 (pp. 6679–6687).

[8] Badirli, S., Liu, X., Xing, Z., Bhowmik, A., Doan, K., & Keerthi, S. S. (2020). Gradient boosting neural networks: Grownet. *arXiv preprint arXiv:2002.07971*.

[9] Baeßler, B., Weiss, K., & Dos Santos, D. P. (2019). Robustness and reproducibility of radiomics in magnetic resonance imaging: a phantom study. *Investigative radiology*, 54(4), 221–228.

[10] baosenguo (2020). Kaggle-MoA 2nd place solution. https://github.com/baosenguo/Kaggle-MoA-2nd-Place-Solution. Online: accessed 2022-08-21.

[11] Becker, A. S., Chaitanya, K., Schawkat, K., Muehlematter, U. J., Hötker, A. M., Konukoglu, E., & Donati, O. F. (2019). Variability of manual segmentation of the prostate in axial t2-weighted mri: A multi-reader study. *European journal of radiology*, 121, 108716.

[12] Berg, A., Oskarsson, M., & O'Connor, M. (2021). Deep ordinal regression with label diversity. In *2020 25th International Conference on Pattern Recognition (ICPR)* (pp. 2740–2747).: IEEE.

[13] Boughorbel, S., Jarray, F., & El-Anbari, M. (2017). Optimal classifier for imbalanced data using matthews correlation coefficient metric. *PloS one*, 12(6), e0177678.

[14] Bragman, F. J., Tanno, R., Eaton-Rosen, Z., Li, W., Hawkes, D. J., Ourselin, S., Alexander, D. C., McClelland, J. R., & Cardoso, M. J. (2018). Quality control in radiotherapy-treatment planning using multi-task learning and uncertainty estimation. *MIDL 2018, Amsterdam, 4–6th July*.

[15] Byrne, S. (2016). A note on the use of empirical auc for evaluating probabilistic forecasts. *Electronic Journal of Statistics*, 10(1), 380–393.

[16] Cao, W., Mirjalili, V., & Raschka, S. (2020). Rank consistent ordinal regression for neural networks with application to age estimation. *Pattern Recognition Letters*, 140, 325–331.

[17] Cardenas, C. E., Yang, J., Anderson, B. M., Court, L. E., & Brock, K. B. (2019). Advances in auto-segmentation. In *Seminars in radiation oncology*, volume 29 (pp. 185–197).

[18] Chen, H.-C., Tan, J., Dolly, S., Kavanaugh, J., Anastasio, M. A., Low, D. A., Harold Li, H., Altman, M., Gay, H., Thorstad, W. L., et al. (2015). Automated contouring error detection based on supervised geometric attribute distribution models for radiation therapy: a general strategy. *Medical physics*, 42(2), 1048–1059.

[19] Chen, T. & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785–794).

[20] Chen, X., Men, K., Chen, B., Tang, Y., Zhang, T., Wang, S., Li, Y., & Dai, J. (2020a). Cnn-based quality assurance for automatic segmentation of breast cancer in radiotherapy. *Frontiers in Oncology*, 10, 524.

[21] Chen, X., Men, K., Chen, B., Tang, Y., Zhang, T., Wang, S., Li, Y., & Dai, J. (2020b). Cnn-based quality assurance for automatic segmentation of breast cancer in radiotherapy. *Frontiers in Oncology*, 10, 524.

[22] Chicco, D. & Jurman, G. (2020). The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC genomics*, 21(1), 1–13.

[23] Chicco, D., Tötsch, N., & Jurman, G. (2021a). The matthews correlation coefficient (mcc) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation. *BioData mining*, 14(1), 1–22.

[24] Chicco, D., Warrens, M. J., & Jurman, G. (2021b). The matthews correlation coefficient (mcc) is more informative than cohen's kappa and brier score in binary classification assessment. *IEEE Access*, 9, 78368–78381.

[25] Christodoulou, E., Ma, J., Collins, G. S., Steyerberg, E. W., Verbakel, J. Y., & Van Calster, B. (2019). A systematic review shows no performance benefit of machine learning over logistic regression for clinical prediction models. *Journal of clinical epidemiology*, 110, 12–22.

[26] Collins, D. L., Holmes, C. J., Peters, T. M., & Evans, A. C. (1995). Automatic 3-d model-based neuroanatomical segmentation. *Human brain mapping*, 3(3), 190–208.

[27] Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., & Schiele, B. (2016). The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3213–3223).

[28] Cox, B. W., Kapur, A., Sharma, A., Lee, L., Bloom, B., Sharma, R., Goode, G., & Potters, L. (2015). Prospective contouring rounds: A novel, high-impact tool for optimizing quality assurance. *Practical radiation oncology*, 5(5), e431–e436.

[29] Crawshaw, M. (2020). Multi-task learning with deep neural networks: A survey. *arXiv preprint arXiv:2009.09796*.

[30] D'Amour, A., Heller, K., Moldovan, D., Adlam, B., Alipanahi, B., Beutel, A., Chen, C., Deaton, J., Eisenstein, J., Hoffman, M. D., et al. (2020). Underspecification presents challenges for credibility in modern machine learning. *arXiv preprint arXiv:2011.03395*.

[31] Dawant, B. M., Hartmann, S. L., Thirion, J.-P., Maes, F., Vandermeulen, D., & Demaerel, P. (1999). Automatic 3-d segmentation of internal structures of the head in mr images using a combination of similarity and free-form transformations. i. methodology and validation on normal subjects. *IEEE transactions on medical imaging*, 18(10), 909–916.

[32] De Craene, M., Du Bois d'Aische, A., Macq, B., & Warfield, S. K. (2004). Multi-subject registration for unbiased statistical atlas construction. In *MICCAI (1)* (pp. 655–662).

[33] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition* (pp. 248–255).

[34] DeVries, T. & Taylor, G. W. (2018). Learning confidence for out-of-distribution detection in neural networks. *arXiv preprint arXiv:1802.04865*.

[35] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

[36] Fernandes, C. D., Dinh, C. V., Walraven, I., Heijmink, S. W., Smolic, M., van Griethuysen, J. J., Simões, R., Losnegård, A., van der Poel, H. G., Pos, F. J., et al. (2018). Biochemical recurrence prediction after radiotherapy for prostate cancer with t2w magnetic resonance imaging radiomic features. *Physics and imaging in radiation oncology*, 7, 9–15.

[37] Ferro, M., de Cobelli, O., Musi, G., Del Giudice, F., Carrieri, G., Busetto, G. M., Falagario, U. G., Sciarra, A., Maggi, M., Crocetto, F., et al. (2022). Radiomics in prostate cancer: an up-to-date review. *Therapeutic Advances in Urology*, 14, 17562872221109020.

[38] Florez, E., Fatemi, A., Claudio, P. P., & Howard, C. M. (2018). Emergence of radiomics: novel methodology identifying imaging biomarkers of disease in diagnosis, response, and progression. *SM journal of clinical and medical imaging*, 4(1).

[39] Fornacon-Wood, I., Mistry, H., Ackermann, C. J., Blackhall, F., McPartlin, A., Faivre-Finn, C., Price, G. J., & O'Connor, J. P. (2020). Reliability and prognostic value of radiomic features are highly dependent on choice of feature extraction platform. *European radiology*, 30(11), 6241–6250.

[40] Gal, Y. (2016). *Uncertainty in deep learning*. PhD thesis, University of Cambridge.

[41] Galloway, M. M. (1975). Texture analysis using gray level run lengths. *Computer graphics and image processing*, 4(2), 172–179.

[42] Gao, B.-B., Liu, X.-X., Zhou, H.-Y., Wu, J., & Geng, X. (2020). Learning expectation of label distribution for facial age and attractiveness estimation. *arXiv preprint arXiv:2007.01771*.

[43] Ge, Y., Udupa, J. K., Nyul, L. G., Wei, L., & Grossman, R. I. (2000). Numerical tissue characterization in MS via standardization of the MR image intensity scale. *Journal of Magnetic Resonance Imaging*, 12(5), 715–721.

[44] Ghai, S. & Haider, M. A. (2015). Multiparametric-mri in diagnosis of prostate cancer. *Indian journal of urology: IJU: journal of the Urological Society of India*, 31(3), 194.

[45] Gong, L., Xu, M., Fang, M., Zou, J., Yang, S., Yu, X., Xu, D., Zhou, L., Li, H., He, B., et al. (2020). Noninvasive prediction of high-grade prostate cancer via biparametric mri radiomics. *Journal of Magnetic Resonance Imaging*, 52(4), 1102–1109.

[46] Gorishniy, Y., Rubachev, I., & Babenko, A. (2022). On embeddings for numerical features in tabular deep learning. *arXiv preprint arXiv:2203.05556*.

[47] Gorishniy, Y., Rubachev, I., Khrulkov, V., & Babenko, A. (2021). Revisiting deep learning models for tabular data. *Advances in Neural Information Processing Systems*, 34, 18932–18943.

[48] grand challenge.org (2014). MICCAI grand challenge: Prostate MR image segmentation 2012 (PROMISE12) online leaderboard. https://promise12.grand-challenge.org/evaluation/leaderboard/. Online: accessed 2020-08-04.

[49] Gudbjartsson, H. & Patz, S. (1995). The Rician distribution of noisy MRI data. *Magnetic resonance in medicine*, 34(6), 910–914.

[50] Gugliandolo, S. G., Pepa, M., Isaksson, L. J., Marvaso, G., Raimondi, S., Botta, F., Gandini, S., Ciardo, D., Volpe, S., Riva, G., et al. (2021). Mri-based radiomics signature for localized prostate cancer: a new clinical tool for cancer aggressiveness prediction? sub-study of prospective phase ii trial on ultra-hypofractionated radiotherapy (airc ig-13218). *European Radiology*, 31(2), 716–728.

[51] Gurevich, P. & Stuke, H. (2019). Pairing an arbitrary regressor with an artificial neural network estimating aleatoric uncertainty. *Neurocomputing*, 350, 291–306.

[52] Halligan, S., Altman, D. G., & Mallett, S. (2015). Disadvantages of using the area under the receiver operating characteristic curve to assess imaging tests: a discussion and proposal for an alternative approach. *European radiology*, 25(4), 932–939.

[53] Haralick, R. M., Shanmugam, K., & Dinstein, I. H. (1973). Textural features for image classification. *IEEE Transactions on systems, man, and cybernetics*, SMC-3(6), 610–621.

[54] Harrell Jr, F. E. (2022). *Biostatistics for Biomedical Research*. Vanderbilt Institute for Clinical and Translational Research.

[55] Hayashi, F. (2011). *Econometrics*. Princeton University Press.

[56] He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision* (pp. 1026–1034).

[57] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).

[58] Hectors, S. J., Cherny, M., Yadav, K. K., Beksaç, A. T., Thulasidass, H., Lewis, S., Davicioni, E., Wang, P., Tewari, A. K., & Taouli, B. (2019). Radiomics features measured with multiparametric magnetic resonance imaging predict prostate cancer aggressiveness. *The Journal of urology*, (pp. 10–1097).

[59] Hobbs, S. K., Shi, G., Homer, R., Harsh, G., Atlas, S. W., & Bednarski, M. D. (2003). Magnetic resonance image–guided proteomics of human glioblastoma multiforme. *Journal of Magnetic Resonance Imaging: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 18(5), 530–536.

[60] Hui, C. B., Nourzadeh, H., Watkins, W. T., Trifiletti, D. M., Alonso, C. E., Dutta, S. W., & Siebers, J. V. (2018). Quality assurance tool for organ at risk delineation in radiation therapy using a parametric statistical approach. *Medical physics*, 45(5), 2089–2096.

[61] Huynh, E., Hosny, A., Guthier, C., Bitterman, D. S., Petit, S. F., Haas-Kogan, D. A., Kann, B., Aerts, H. J., & Mak, R. H. (2020a). Artificial intelligence in radiation oncology. *Nature Reviews Clinical Oncology*, 17(12), 771–781.

[62] Huynh, E., Hosny, A., Guthier, C., Bitterman, D. S., Petit, S. F., Haas-Kogan, D. A., Kann, B., Aerts, H. J., & Mak, R. H. (2020b). Artificial intelligence in radiation oncology. *Nature Reviews Clinical Oncology*, 17(12), 771–781.

[63] Hwang, J. G. & Ding, A. A. (1997). Prediction intervals for artificial neural networks. *Journal of the American Statistical Association*, 92(438), 748–757.

[64] Ince, R. A., Giordano, B. L., Kayser, C., Rousselet, G. A., Gross, J., & Schyns, P. G. (2017). A statistical framework for neuroimaging data analysis based on mutual information estimated via a gaussian copula. *Human brain mapping*, 38(3), 1541–1573.

[65] Inoue, H. (2018). Data augmentation by pairing samples for images classification. *arXiv preprint arXiv:1801.02929*.

[66] International Agency for Research on Cancer (2016). *World cancer report: World Health Organization*. IARC Press.

[67] Isaksson, L. J., Pepa, M., Summers, P., Zaffaroni, M., Vincini, M. G., Corrao, G., Mazzola, G. C., Rotondi, M., Presti, G. L., Raimondi, S., et al. (2022a). Comparison of automated segmentation techniques for magnetic resonance images of the prostate. *PREPRINT (Version 1) available at Research Square [https://doi.org/10.21203/rs.3.rs-1850296/v1]*.

[68] Isaksson, L. J., Raimondi, S., Botta, F., Pepa, M., Gugliandolo, S. G., De Angelis, S. P., Marvaso, G., Petralia, G., De Cobelli, O., Gandini, S., et al. (2020). Effects of mri image normalization techniques in prostate cancer radiomics. *Physica Medica*, 71, 7–13.

[69] Isaksson, L. J., Summers, P., Bhalerao, A., Gandini, S., Raimondi, S., Pepa, M., Zaffaroni, M., Corrao, G., Mazzola, G. C., Rotondi, M., et al. (2022b). Quality assurance for automatically generated contours with additional deep learning. *Insights into Imaging*, 13(1), 1–10.

[70] Isaksson, L. J., Summers, P., Raimondi, S., Gandini, S., Bhalerao, A., Marvaso, G., Petralia, G., Pepa, M., & Jereczek-Fossa, B. A. (2022c). Mixup (sample pairing) can improve the performance of deep segmentation networks. *Journal of Artificial Intelligence and Soft Computing Research*, 31, 29–39.

[71] Isola, P., Zhu, J.-Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1125–1134).

[72] Jacobucci, R., Littlefield, A. K., Millner, A. J., Kleiman, E. M., & Steinley, D. (2021). Evidence of inflated prediction performance: A commentary on machine learning and suicide research. *Clinical Psychological Science*, 9(1), 129–134.

[73] Jakobsen, J. C., Gluud, C., Wetterslev, J., & Winkel, P. (2017). When and how should multiple imputation be used for handling missing data in randomised clinical trials–a practical guide with flowcharts. *BMC medical research methodology*, 17(1), 1–10.

[74] Jie, C., Rongbo, L., & Ping, T. (2014). The value of diffusion-weighted imaging in the detection of prostate cancer: a meta-analysis. *European radiology*, 24(8), 1929–1941.

[75] Jin, J., Dundar, A., & Culurciello, E. (2014). Flattened convolutional neural networks for feedforward acceleration. *arXiv preprint arXiv:1412.5474*.

[76] Jolicoeur-Martineau, A. (2018). The relativistic discriminator: a key element missing from standard gan. *arXiv preprint arXiv:1807.00734*.

[77] Kadra, A., Lindauer, M., Hutter, F., & Grabocka, J. (2021). Regularization is all you need: Simple neural nets can excel on tabular data. *arXiv preprint arXiv:2106.11189*.

[78] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30.

[79] Kendall, A. & Gal, Y. (2017). What uncertainties do we need in bayesian deep learning for computer vision? *arXiv preprint arXiv:1703.04977*.

[80] Khan, Z., Yahya, N., Alsaih, K., Al-Hiyali, M. I., & Meriaudeau, F. (2021). Recent automatic segmentation algorithms of mri prostate regions: A review. *IEEE Access*.

[81] Kingma, D. P. & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[82] Klambauer, G., Unterthiner, T., Mayr, A., & Hochreiter, S. (2017). Self-normalizing neural networks. *Advances in neural information processing systems*, 30, 972–982.

[83] Kossen, J., Band, N., Lyle, C., Gomez, A. N., Rainforth, T., & Gal, Y. (2021). Self-attention between datapoints: Going beyond individual input-output pairs in deep learning. *Advances in Neural Information Processing Systems*, 34, 28742–28756.

[84] Kumar, V., Gu, Y., Basu, S., Berglund, A., Eschrich, S. A., Schabath, M. B., Forster, K., Aerts, H. J., Dekker, A., Fenstermacher, D., et al. (2012). Radiomics: the process and the challenges. *Magnetic resonance imaging*, 30(9), 1234–1248.

[85] Kuo, M. D., Gollub, J., Sirlin, C. B., Ooi, C., & Chen, X. (2007). Radiogenomic analysis to identify imaging phenotypes associated with drug response gene expression programs in hepatocellular carcinoma. *Journal of Vascular and Interventional Radiology*, 18(7), 821–830.

[86] Lafata, K., Cai, J., Wang, C., Hong, J., Kelsey, C., & Yin, F. (2017). Sensitivity of radiomic features to acquisition noise and respiratory motion. *International Journal of Radiation Oncology*Biology*Physics*, 99(2, Supplement), S93–S94.

[87] Lambin, P., Rios-Velazquez, E., Leijenaar, R., Carvalho, S., Van Stiphout, R. G., Granton, P., Zegers, C. M., Gillies, R., Boellard, R., Dekker, A., et al. (2012). Radiomics: extracting more information from medical images using advanced feature analysis. *European journal of cancer*, 48(4), 441–446.

[88] Lawrence, E. M., Gallagher, F. A., Barrett, T., Warren, A. Y., Priest, A. N., Goldman, D. A., Sala, E., & Gnanapragasam, V. J. (2014). Preoperative 3-t diffusion-weighted mri for the qualitative and quantitative assessment of extracapsular extension in patients with intermediate-or high-risk prostate cancer. *American Journal of Roentgenology*, 203(3), W280–W286.

[89] Lee, K., Lee, H., Lee, K., & Shin, J. (2017). Training confidence-calibrated classifiers for detecting out-of-distribution samples. *arXiv preprint arXiv:1711.09325*.

[90] Li, H., Xiong, P., An, J., & Wang, L. (2018). Pyramid attention network for semantic segmentation. *arXiv preprint arXiv:1805.10180*.

[91] Liang, D., Yang, F., Zhang, T., & Yang, P. (2018). Understanding mixup training methods. *IEEE Access*, 6, 58774–58783.

[92] Liang, S., Li, Y., & Srikant, R. (2017). Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*.

[93] Litjens, G., Toth, R., van de Ven, W., Hoeks, C., Kerkstra, S., van Ginneken, B., Vincent, G., Guillard, G., Birbeck, N., Zhang, J., et al. (2014). Evaluation of prostate segmentation algorithms for mri: the promise12 challenge. *Medical image analysis*, 18(2), 359–373.

[94] Liu, X., Zou, Y., Kuang, H., & Ma, X. (2020). Face image age estimation based on data augmentation and lightweight convolutional neural network. *Symmetry*, 12(1), 146.

[95] Lobo, J. M., Jiménez-Valverde, A., & Real, R. (2008). Auc: a misleading measure of the performance of predictive distribution models. *Global ecology and Biogeography*, 17(2), 145–151.

[96] Lundberg, S. M. & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 4765–4774.

[97] Madabhushi, A. & Udupa, J. K. (2006). New methods of MR image intensity standardization via generalized scale. *Medical physics*, 33(9), 3426–3434.

[98] Madabhushi, A., Udupa, J. K., & Souza, A. (2006). Generalized scale: Theory, algorithms, and application to image inhomogeneity correction. *Computer vision and image understanding*, 101(2), 100–121.

[99] Marcus, D. M., Rossi, P. J., Nour, S. G., & Jani, A. B. (2014). The impact of multiparametric pelvic magnetic resonance imaging on risk stratification in patients with localized prostate cancer. *Urology*, 84(1), 132–137.

[100] McIntosh, C., Svistoun, I., & Purdie, T. G. (2013). Groupwise conditional random forests for automatic shape classification and contour quality assessment in radiotherapy planning. *IEEE transactions on medical imaging*, 32(6), 1043–1057.

[101] Mehrtash, A., Wells, W. M., Tempany, C. M., Abolmaesumi, P., & Kapur, T. (2020). Confidence calibration and predictive uncertainty estimation for deep medical image segmentation. *IEEE transactions on medical imaging*, 39(12), 3868–3878.

[102] Men, K., Geng, H., Biswas, T., Liao, Z., & Xiao, Y. (2020). Automated quality assurance of oar contouring for lung cancer based on segmentation with deep active learning. *Frontiers in Oncology*, 10, 986.

[103] Miller, M. I., Christensen, G. E., Amit, Y., & Grenander, U. (1993). Mathematical textbook of deformable neuroanatomies. *Proceedings of the National Academy of Sciences*, 90(24), 11944–11948.

[104] Milletari, F., Navab, N., & Ahmadi, S.-A. (2016). V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)* (pp. 565–571).

[105] NIH National Cancer Institute (2022). Cancer stat facts: Prostate cancer. https://seer.cancer.gov/statfacts/html/prost.html. Online: accessed 2022-08-14.

[106] Nketiah, G., Elschot, M., Kim, E., Teruel, J. R., Scheenen, T. W., Bathen, T. F., & Selnæs, K. M. (2017). T2-weighted MRI-derived textural features reflect prostate cancer aggressiveness: preliminary results. *European radiology*, 27(7), 3050–3059.

[107] Norouzi, A., Rahim, M. S. M., Altameem, A., Saba, T., Rad, A. E., Rehman, A., & Uddin, M. (2014). Medical image segmentation methods, algorithms, and applications. *IETE Technical Review*, 31(3), 199–213.

[108] Nowak, J., Malzahn, U., Baur, A. D., Reichelt, U., Franiel, T., Hamm, B., & Durmus, T. (2016). The value of adc, t2 signal intensity, and a combination of both parameters to assess gleason score and primary gleason grades in patients with known prostate cancer. *Acta Radiologica*, 57(1), 107–114.

[109] Nyúl, L. G. & Udupa, J. K. (1999). On standardizing the MR image intensity scale. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 42(6), 1072–1081.

[110] Nyúl, L. G., Udupa, J. K., & Zhang, X. (2000). New variants of a method of MRI scale standardization. *IEEE transactions on medical imaging*, 19(2), 143–150.

[111] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.

[112] Petersen, E., Potdevin, Y., Mohammadi, E., Zidowitz, S., Breyer, S., Nowotka, D., Henn, S., Pechmann, L., Leucker, M., Rostalski, P., et al. (2022). Responsible and regulatory conform machine learning for medicine: A survey of challenges and solutions. *IEEE Access*.

[113] Pfaehler, E., Beukinga, R. J., de Jong, J. R., Slart, R. H., Slump, C. H., Dierckx, R. A., & Boellaard, R. (2019). Repeatability of 18F-FDG PET radiomic features: A phantom study to explore sensitivity to image reconstruction settings, noise, and delineation method. *Medical physics*, 46(2), 665–678.

[114] Pham, D. L., Xu, C., & Prince, J. L. (2000). Current methods in medical image segmentation. *Annual review of biomedical engineering*, 2(1), 315–337.

[115] Pietikäinen, M. & Ojala, T. (1996). Texture analysis in industrial applications. In *Image Technology* (pp. 337–359). Springer.

[116] Popov, S., Morozov, S., & Babenko, A. (2019). Neural oblivious decision ensembles for deep learning on tabular data. *arXiv preprint arXiv:1909.06312*.

[117] Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). Catboost: unbiased boosting with categorical features. *Advances in neural information processing systems*, 31, 6638–6649.

[118] Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention* (pp. 234–241).

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

[120] Rosenblatt, F. (1962). *Perceptions and the theory of brain mechanisms*. Spartan books.

[121] Rutman, A. M. & Kuo, M. D. (2009). Radiogenomics: creating a link between molecular diagnostics and diagnostic imaging. *European journal of radiology*, 70(2), 232–241.

[122] Sahiner, B., Pezeshk, A., Hadjiiski, L. M., Wang, X., Drukker, K., Cha, K. H., Summers, R. M., & Giger, M. L. (2019). Deep learning in medical imaging and radiation therapy. *Medical physics*, 46(1), e1–e36.

[123] Schwier, M., van Griethuysen, J., Vangel, M. G., Pieper, S., Peled, S., Tempany, C., Aerts, H. J., Kikinis, R., Fennessy, F. M., & Fedorov, A. (2019). Repeatability of multiparametric prostate mri radiomics features. *Scientific reports*, 9(1), 1–16.

[124] scikit learn (2020). Preprocessing data. `https://scikit-learn.org/stable/modules/preprocessing.html`. Online: accessed 2022-08-24.

[125] Shah, M., Xiao, Y., Subbanna, N., Francis, S., Arnold, D. L., Collins, D. L., & Arbel, T. (2011). Evaluating intensity normalization on MRIs of human brain with multiple sclerosis. *Medical image analysis*, 15(2), 267–282.

[126] Shahedi, M., Cool, D. W., Romagnoli, C., Bauman, G. S., Bastian-Jordan, M., Gibson, E., Rodrigues, G., Ahmad, B., Lock, M., Fenster, A., et al. (2014). Spatially varying accuracy and reproducibility of prostate segmentation in magnetic resonance images using manual and semiautomated methods. *Medical physics*, 41(11), 113503.

[127] Sharma, N. & Aggarwal, L. M. (2010). Automated medical image segmentation techniques. *Journal of medical physics/Association of Medical Physicists of India*, 35(1), 3.

[128] Shinohara, R. T., Sweeney, E. M., Goldsmith, J., Shiee, N., Mateen, F. J., Calabresi, P. A., Jarso, S., Pham, D. L., Reich, D. S., Crainiceanu, C. M., et al. (2014). Statistical normalization techniques for magnetic resonance imaging. *NeuroImage: Clinical*, 6, 9–19.

[129] Shwartz-Ziv, R. & Armon, A. (2022). Tabular data: Deep learning is not all you need. *Information Fusion*, 81, 84–90.

[130] Siemens Healtineers (2022). syngo.via. https://www.siemens-healthineers.com/medical-imaging-it/advanced-visualization-solutions/syngovia. Online: accessed 2022-08-27.

[131] Smith, W. L., Lewis, C., Bauman, G., Rodrigues, G., D'Souza, D., Ash, R., Ho, D., Venkatesan, V., Downey, D., & Fenster, A. (2007). Prostate volume contouring: a 3d analysis of segmentation using 3dtrus, ct, and mr. *International Journal of Radiation Oncology* Biology* Physics*, 67(4), 1238–1247.

[132] Somepalli, G., Goldblum, M., Schwarzschild, A., Bruss, C. B., & Goldstein, T. (2021). Saint: Improved neural networks for tabular data via row attention and contrastive pre-training. *arXiv preprint arXiv:2106.01342*.

[133] Song, W., Shi, C., Xiao, Z., Duan, Z., Xu, Y., Zhang, M., & Tang, J. (2019). Autoint: Automatic feature interaction learning via self-attentive neural networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* (pp. 1161–1170).

[134] Sterne, J. A., White, I. R., Carlin, J. B., Spratt, M., Royston, P., Kenward, M. G., Wood, A. M., & Carpenter, J. R. (2009). Multiple imputation for missing data in epidemiological and clinical research: potential and pitfalls. *Bmj*, 338.

[135] Steyerberg, E. W. & Harrell, F. E. (2016). Prediction models need appropriate internal, internal–external, and external validation. *Journal of clinical epidemiology*, 69, 245–247.

[136] Tajbakhsh, N., Jeyaseelan, L., Li, Q., Chiang, J. N., Wu, Z., & Ding, X. (2020). Embracing imperfect datasets: A review of deep learning solutions for medical image segmentation. *Medical Image Analysis*, 63, 101693.

[137] Tan, M. & Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning* (pp. 6105–6114).

[138] Tan, M., Pang, R., & Le, Q. V. (2020). Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 10781–10790).

[139] Thulasidasan, S., Chennupati, G., Bilmes, J. A., Bhattacharya, T., & Michalak, S. (2019). On mixup training: Improved calibration and predictive uncertainty for deep neural networks. *Advances in Neural Information Processing Systems*, 32, 13843–13855.

[140] Thung, K.-H. & Wee, C.-Y. (2018). A brief review on multi-task learning. *Multimedia Tools and Applications*, 77(22), 29705–29725.

[141] Traverso, A., Wee, L., Dekker, A., & Gillies, R. (2018). Repeatability and reproducibility of radiomic features: a systematic review. *International Journal of Radiation Oncology* Biology* Physics*, 102(4), 1143–1158.

[142] Udupa, J. K. & Samarasekera, S. (1996). Fuzzy connectedness and object definition: Theory, algorithms, and applications in image segmentation. *Graphical models and image processing*, 58(3), 246–261.

[143] Van Griethuysen, J. J., Fedorov, A., Parmar, C., Hosny, A., Aucoin, N., Narayan, V., Beets-Tan, R. G., Fillion-Robin, J.-C., Pieper, S., & Aerts, H. J. (2017). Computational radiomics system to decode the radiographic phenotype. *Cancer research*, 77(21), e104–e107.

[144] Vandenhende, S., Georgoulis, S., Proesmans, M., Dai, D., & Van Gool, L. (2020). Revisiting multi-task learning in the deep learning era. *arXiv preprint arXiv:2004.13379*, 2(3).

[145] Vandewinckele, L., Claessens, M., Dinkla, A., Brouwer, C., Crijns, W., Verellen, D., & van Elmpt, W. (2020). Overview of artificial intelligence-based applications in radiotherapy: recommendations for implementation and quality assurance. *Radiotherapy and Oncology*, 153, 55–66.

[146] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30, 5999–6010.

[147] Vinod, S. K., Min, M., Jameson, M. G., & Holloway, L. C. (2016). A review of interventions to reduce inter-observer variability in volume delineation in radiation oncology. *Journal of medical imaging and radiation oncology*, 60(3), 393–406.

[148] Wang, J., Wu, C.-J., Bao, M.-L., Zhang, J., Wang, X.-N., & Zhang, Y.-D. (2017). Machine learning-based analysis of MR radiomics can help to improve the diagnostic performance of PI-RADS v2 in clinically relevant prostate cancer. *European radiology*, 27(10), 4082–4090.

[149] Wang, R., Shivanna, R., Cheng, D., Jain, S., Lin, D., Hong, L., & Chi, E. (2021). Dcn v2: Improved deep & cross network and practical lessons for web-scale learning to rank systems. In *Proceedings of the Web Conference 2021* (pp. 1785–1797).

[150] Wong, J., Fong, A., McVicar, N., Smith, S., Giambattista, J., Wells, D., Kolbeck, C., Giambattista, J., Gondara, L., & Alexander, A. (2020). Comparing deep learning-based auto-segmentation of organs at risk and clinical target volumes to expert inter-observer variability in radiotherapy planning. *Radiotherapy and Oncology*, 144, 152–158.

[151] Woźnicki, P., Westhoff, N., Huber, T., Riffel, P., Froelich, M. F., Gresser, E., von Hardenberg, J., Mühlberg, A., Michel, M. S., Schoenberg, S. O., et al. (2020). Multiparametric mri for prostate cancer characterization: Combined use of radiomics model with pi-rads and clinical parameters. *Cancers*, 12(7), 1767.

[152] Wu, Z., Shen, C., & Hengel, A. v. d. (2016). Bridging category-level and instance-level semantic image segmentation. *arXiv preprint arXiv:1605.06885*.

[153] Wynants, L., Van Calster, B., Collins, G. S., Riley, R. D., Heinze, G., Schuit, E., Bonten, M. M., Dahly, D. L., Damen, J. A., Debray, T. P., et al. (2020). Prediction models for diagnosis and prognosis of covid-19: systematic review and critical appraisal. *bmj*, 369.

[154] Xue, C., Yuan, J., Zhou, Y., Wong, O. L., Cheung, K. Y., & Yu, S. K. (2022). Acquisition repeatability of mri radiomics features in the head and neck: a dual-3d-sequence multi-scan study. *Visual computing for industry, biomedicine, and art*, 5(1), 1–13.

[155] Yakubovskiy, P. (2019). Segmentation models. https://github.com/qubvel/segmentation_models. Online: accessed 2021-03-14.

[156] Zhang, H., Cisse, M., Dauphin, Y. N., & Lopez-Paz, D. (2017). mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*.

[157] Zhang, L., Fried, D. V., Fave, X. J., Hunter, L. A., Yang, J., & Court, L. E. (2015). IBEX: an open infrastructure software platform to facilitate collaborative work in radiomics. *Medical physics*, 42(3), 1341–1353.

[158] Zhang, M., Lucas, J., Ba, J., & Hinton, G. E. (2019a). Lookahead optimizer: k steps forward, 1 step back. In *Advances in Neural Information Processing Systems* (pp. 9597–9608).

[159] Zhang, Y., Plautz, T. E., Hao, Y., Kinchen, C., & Li, X. A. (2019b). Texture-based, automatic contour validation for online adaptive replanning: a feasibility study on abdominal organs. *Medical Physics*, 46(9), 4010–4020.

[160] Zhao, B., Tan, Y., Tsai, W.-Y., Qi, J., Xie, C., Lu, L., & Schwartz, L. H. (2016). Reproducibility of radiomics for deciphering tumor phenotype with imaging. *Scientific reports*, 6(1), 1–7.

[161] Zhu, Y., Li, H., Guo, W., Drukker, K., Lan, L., Giger, M. L., & Ji, Y. (2015). Deciphering genomic underpinnings of quantitative mri-based radiomic phenotypes of invasive breast carcinoma. *Scientific reports*, 5(1), 1–10.

[162] Zwanenburg, A., Vallières, M., Abdalah, M. A., Aerts, H. J., Andrearczyk, V., Apte, A., Ashrafinia, S., Bakas, S., Beukinga, R. J., Boellaard, R., et al. (2020). The image biomarker standardization initiative: standardized quantitative radiomics for high-throughput image-based phenotyping. *Radiology*, 295(2), 328–338.

UNIVERSITAS STUDIORUM MEDIOLANENSIS