FACULTY OF ENGINEERING
AND ARCHITECTURE

A pathology-based prediction strategy for SBRT response in prostate cancer

Tim Willems
Student number: 01707294

Supervisors: Prof. Kathleen Marchal, Prof. dr. Roel Van Holen
Counsellor: Maarten Larmuseau

Master's dissertation submitted in order to obtain the academic degree of
Master of Science in Computer Science Engineering

Academic year 2021-2022

GHENT
UNIVERSITY

FACULTY OF ENGINEERING
AND ARCHITECTURE

A pathology-based prediction strategy for SBRT response in prostate cancer

Tim Willems
Student number: 01707294

Supervisors: Prof. Kathleen Marchal, Prof. dr. Roel Van Holen
Counsellor: Maarten Larmuseau

Master's dissertation submitted in order to obtain the academic degree of
Master of Science in Computer Science Engineering

Academic year 2021-2022

GHENT
UNIVERSITY

Acknowledgements

Over the course of a year, I worked on a topic that combined two of my interests: the medical field and artificial intelligence. I was able to put my skills to test and brought into practice what I have learned over the past few years. Conducting my own research learned me multiple valuable lessons and in this section I want to take a moment to thank everyone who contributed to the final result.

First of all, I want to thank my promotors Prof. Kathleen Marchal and Prof. Roel Van Holen for giving me the opportunity to conduct research on an interdisciplinary subject.

I also want to thank my supervisors Maarten Larmuseau and Marija Pizurica. They were always ready whenever I had questions or needed a second opinion. Our weekly sessions always resulted in valuable feedback which without a doubt has significantly improved this work. Special thanks to Marija who despite the different time zone, always made time, be it early in the morning or late in the evening. Thank you!

Lastly, I want to thank my friends and family, especially my father. He proofread this work multiple times and probably knows it by heart now. I cannot thank him enough for the support and feedback.

To all of you, a sincere thank you!

Tim

Permission of use on loan

Tim Willems
May 2022

Remark on the master's dissertation and the oral presentation

This master's dissertation is part of an exam. Any comments formulated by the assessment committee during the oral presentation of the master's dissertation are not included in this text.

A pathology-based prediction strategy for SBRT response in prostate cancer

by
Tim Willems

Master's dissertation submitted in order to obtain the academic degree of
Master of Science in Computer Science Engineering

Academic year 2021 - 2022

Supervisors: Prof. Kathleen Marchal, Prof. dr. Roel Van Holen
Counsellors: Maarten Larmuseau, Ir. Marija Pizurica

Faculty of Engineering and Architecture
Ghent University

Abstract

Therapy response prediction in cancer enables clinicians to personalise treatment planning for each patient. Because of the heterogeneous nature of cancer, patients often receive ineffective treatments, exposing them to unneeded side effects. Gene analysis have shown promising results towards therapy response prediction but is slow and expensive. On the other hand, microscopic tissue samples are widely available because they have high diagnostic value.

We evaluated if Whole Slide Images (WSI) contain features to predict the outcome of Stereotactic Body Radiation Therapy (SBRT) on prostate cancer. A tile-level based deep learning model was trained on 72 patients from the SBRT dataset. The dataset contains multiple end-points on patients with metachronous oligorecurrent prostate cancer. We concluded that microscopic tissue samples have limited prediction value for the outcome of SBRT in prostate cancer.

Keywords

deep learning, segmentation, therapy response prediction, whole slide imaging

A pathology-based prediction strategy for SBRT response in prostate cancer

Tim Willems
Supervisors: Kathleen Marchal, Roel Van Holen, Maarten Larmuseau, Marija Pizurica

***Abstract*—Therapy response prediction in cancer enables clinicians to personalise treatment planning for each patient. Because of the heterogeneous nature of cancer, patients often receive ineffective treatments, exposing them to unneeded side-effects. Gene analysis have shown promising results towards therapy response prediction but is slow and expensive. On the other hand, microscopic tissue samples are widely available because they have high diagnostic value.**

We evaluated if Whole Slide Images (WSI) contain features to predict the outcome of Stereotactic Body Radiation Therapy (SBRT) on prostate cancer. A tile-level based deep learning model was trained on 72 patients from the SBRT dataset. The dataset contains multiple end-points on patients with metachronous oligorecurrent prostate cancer. We concluded that microscopic tissue samples have limited prediction value for the outcome of SBRT in prostate cancer.

***Index Terms*—deep learning, segmentation, therapy response prediction, whole slide imaging**

I. Introduction

Oncology has gained in attention over the past decade and researchers are attempting to personalise the treatment planning for cancer patients [1]. Patients receiving the same treatment for the same cancer subtype, do not necessarily react in the same way [2]. Given the genetic nature of cancer, models based on gene expression data have great potential to predict the correct therapy for cancer for each unique patient [3, 4]. However, the cost and time of genetic analysis serve as a bottleneck in oncology workflows [2].

Therefore, attention is shifting towards other more easily accessible datatypes. For example, MRI-scans are increasingly used to assess the morphological features for targeted therapies, building upon developments in genomics and molecular biology features [5, 6, 7, 8]. Shao et al. combined radiological and pathological information of a tumor to predict the outcome of chemoradiotherapy in rectal cancer.

To the best of our knowledge, no other studies exist that assess the predictive value of WSIs in therapy response prediction. Because the predictive features are unknown, we opted to use deep learning models known for their automatic feature extraction capabilities [9]. More specifically, we adopted convolutional neural networks (CNNs) which are commonly used for image analysis. To reduce the amount of noisy data, a tumor segmentation model was developed to extract the Region of Interest (ROI).

The main contributions of this work are:

1) Development of a robust deep learning model to automatically segment the tumor regions on WSIs.
2) Development of an advanced post-processing method to improve the segmentation results.
3) Evaluation about the predictive features of WSIs with respect to SBRT outcome.

II. Data preparation

Two datasets containing H&E stained WSIs are used in this work: the PANDA dataset is a public dataset originating from Kaggle [10]. It contains binary masks for each tumor region and is used to achieve segmentation. The SBRT dataset is the result of two independently organised trials by Johns Hopkins and Ghent University. The trails contained two arms: an observation arm and an arm receiving SBRT. Prostate biopsies were performed for diagnostic evaluation, allowing us to link the WSIs with the outcome of SBRT.

A. PANDA dataset

The public dataset contains 10616 H&E stained WSIs provided by Radboud university medical centre and Karolinska institute. Each WSI corresponds with a mask annotating the tumor region. Slides by Radboud have different masks for each Gleason grade and only annotate epithelial cell clusters. In contrast, slides by Karolinska combine epithelial cell clusters and surrounding stroma, forming more coarse grained masks. Consequently, we decided to only use slides provided by Radboud institute, reducing the total number of available data to 5160 WSIs.

We combined the masks corresponding with Gleason score 3 or higher and gave them the same label. This resulted in a binary mask with label 1 epithelial cancer cells and label 0 background, stroma and healthy epithelium. An example of which can be found in figure 1.

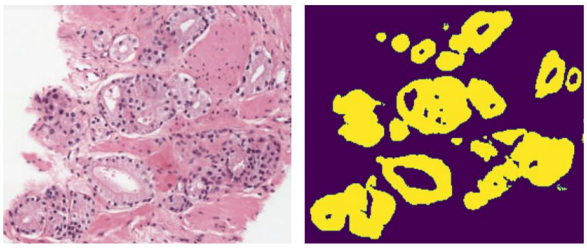

Fig. 1. Part of a WSI with its corresponding binary mask. Yellow is considered cancer tissue, purple is background or healthy tissue.

To reduce training time, we randomly selected 1000 WSIs from the Radboud dataset. We ensured that each Gleason score

is equally represented so that our model would be able to cope with the structural differences. We combined the Gleason grades into ISUP grade groups ranging from 2 to 5. Each ISUP grade group represents 25% of the selected 1000 WSIs.

B. SBRT dataset

The dataset contains trial data of 175 patients, from which 139 received SBRT. H&E stained WSIs were available for 72 of those patients. Tumors were annotated by a pathologist directly on the WSI with pen marks. The trial data contained metadata such as age, Gleason grade of the tumor and end-points indicating the result of the received treatment. We decided upon the binary end-point Prostate-Specific Antigen (PSA) failure as a label for implying SBRT success. The threshold for PSA failure was the increase of PSA levels above 4.0 ng/ml. According to this label, SBRT was unsuccessful (PSA failure) for 46 patients (63%). 26 patients (37%) received a successful treatment (no PSA failure).

III. Automatic tumor detection

Segmenting WSIs is a challenging task for three reasons. First, the high dimensions of a WSI prevent using the entire WSI as an input for a deep learning model. Second, the Gleason scores represent structural differences between WSIs. The model should be able to cope with those differences. Finally, the H&E staining causes colour differences between the WSIs based on the procedures used.

Taking these challenges into account, we propose our deep learning framework in figure 2. The WSI is divided into small non-overlapping tiles of 512×512 at 1.0 μm/pixel. Tiles consisting out of 50% or more white space (defined as brightness $\geq$230 of 255) are removed from the tile set. The remaining tiles are subjected to Reinhard stain normalisation proposed by Reinhard et al. [11]. This technique normalises the tiles in Lab colour space based on a target image. In our case, the target image is an artificial image created by averaging over all the tiles in the training set in Lab colour space. During training, stain augmentation is applied to the normalised tiles. This makes the model more robust by randomly changing hue, saturation, contrast and intensity. Parallel with stain normalisation and augmentation, we partially perform canny edge detection [12] to extract the gradient intensities from the tiles. By taking the image gradients into account, we hypothesised that the model would be able to automatically recognise blurring artefacts and achieve higher accuracy towards the edges of each segment. The normalised and augmented RGB tiles are combined with the gradient intensity map, resulting in 4 feature channels. Consequently, this data structure is used as input for a Unet model. A Unet is an encoder-decoder deep network specialised in segmentation, making use of skip-connections. We used a total of 4 layers: 3 contracting-detracting layers and 1 bottleneck. The output is a binary mask of 512×512 annotating cancerous epithelium. As a final step, the masks are put back together into their original position in the WSI. The result is a binary mask with the same dimensions as the original WSI.

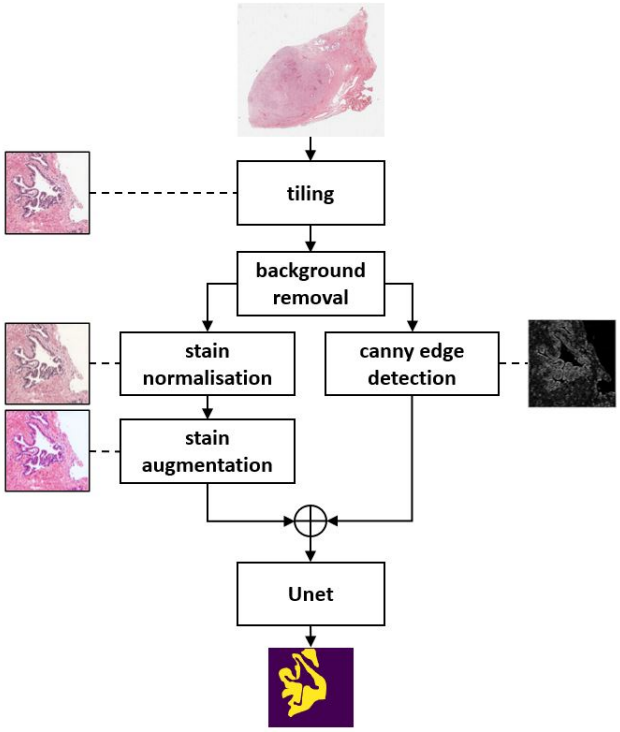

Fig. 2. The segmentation pipeline

We divided the 1000 WSIs into a train-validate set. 752 WSIs were used for training and 248 for validating. The distribution of the ISUP grade groups was kept the same in both sets. Additionally, a test set of 250 WSIs was created where no patient is part of the train or validate set. The 72 manually annotated slides from the SBRT dataset were used as an external test set. The Unet was trained for 20 epochs with batch size 15 and learning rate 0.0001 using the Adam optimiser [13]. We decided to use cross entropy loss as the objective function and regularised our network with a dropout of 50% during training.

We managed to achieve an average AUC of 0.9472 ± 0.0005 on the test set. The average AUC for ISUP group grade 2 to 5 were 0.95, 0.95, 0.95 and 0.94 respectively. The model does not seem to be affected by the structural differences that correspond with each Gleason score. To ensure generality and robustness, we tested the framework on the SBRT dataset after additional post-processing proposed in the next section.

IV. ROI extraction

To select the general tumor region (epithelium and stroma) and select the relevant tiles for therapy response prediction, we perform an extra post-processing step based on heatmaps. This allows for a better match with pathologists. Figure 3 illustrates the main flow with the WSI example originating from the SBRT dataset. The design of this framework is based upon two observations: First, the tumor region corresponds with the highest density regions in the binary masks. Second,

the false positives are generally speaking small isolated peaks with respect to the tumor region.

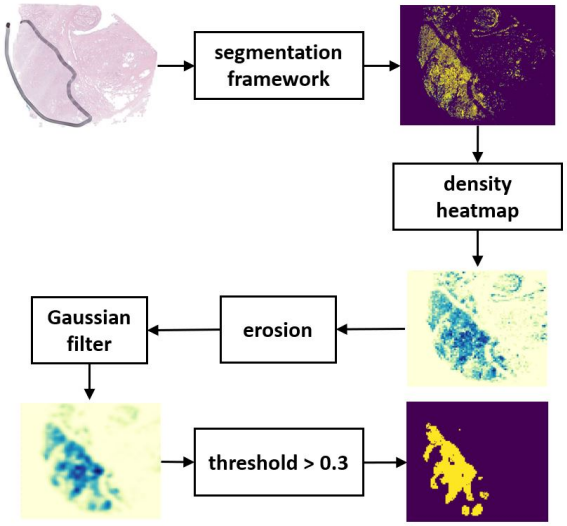

Fig. 3. The ROI extraction pipeline

First, a heatmap is calculated based on the local density of each output mask created using the framework of previous section. The mask density is calculated by dividing the sum of all the pixel values of the binary mask with the total tile area. The resulting heatmap is normalised by dividing with the maximum density in the WSI. The normalisation is necessary when the tumor is a small region. If no normalisation is applied, the next steps, erosion and Gaussian filtering, can potentially remove those regions. The morphological operator erosion is applied to remove small low density clusters from the heatmap. However, it also amplifies small gaps in the high density regions. For this reason, a Gaussian filter with 5×5 kernel size is applied to close the gaps without amplifying peaks. The final result is normalised using the same method as before. A threshold can now be applied on the heatmap to extract the general ROI. We found a threshold of 0.3 to be optimal for our use case.

The raw output on the external SBRT dataset resulted in precision 0.67 and dice similarity score 0.60 with respect to the manual coarse grained annotations. Using the proposed ROI extraction framework, we were able to achieve a dice similarity score of 0.82 and precision of 0.92 on the external SBRT dataset. In figure 4, we demonstrate the results for each patient in the SBRT dataset independently with and without the ROI extraction framework. It can be observed that the ROI extraction pipeline improves the results significantly. The entire framework (segmentation+ROI extraction) was able to achieve acceptable results for most patients. The resulting ROI allows us to reduce the number of relevant tiles in each WSI significantly for therapy response prediction.

V. Therapy Response Prediction

In our workflow, we first divide each WSI into small tiles of 512×512 at 0.5 μm/pixel so that the WSI can be processed by a deep learning model. The labels indicating SBRT outcome are on patient level and provide no local insights on the WSIs themselves. The tiled WSI form a bag of instances where each instance receives the same coarse grained label: PSA failure or not. We filter tiles not containing tumor out of each bag using the segmentation and ROI extraction pipeline. This also automatically removes background tiles and blurring artefacts. The reduction of amount of tiles lowers the chance that tiles are irrelevant towards the patient-level label. Still, we hypothesise that only small hotspots on the tumor will be relevant and that a part of the bag will serve as noise.

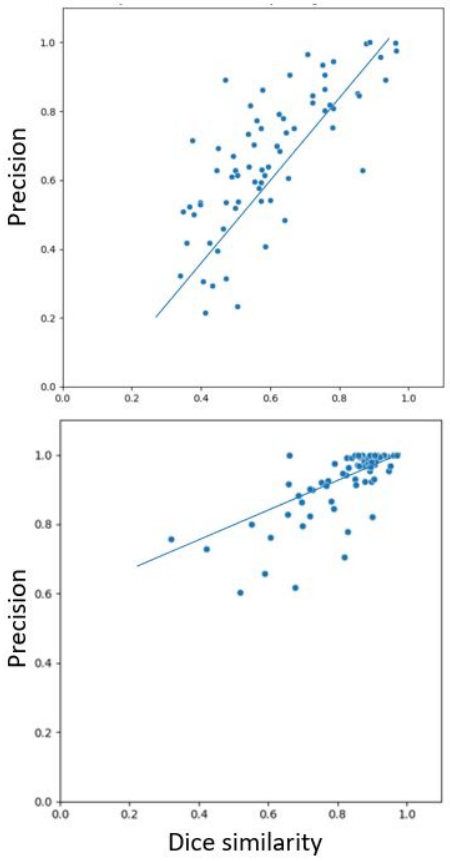

Fig. 4. The dice similarity put against the precision for each patient in the SBRT dataset independently. (Top) The results before post-processing (Bottom) The results after post-processing

The tiles are processed independently by a partially pre-trained RESnet18 [14] from ImageNet. A RESnet is an improved version of the traditional CNN that contains skip-connections to reduce the impact of the vanishing gradient. The last two layers are retrained. We make of use of transfer learning to reduce the change of overfitting and to automatically extract the high level features in the first layers. It also reduces the training time significantly.

The dataset containing 72 patients was divided into a train and validate set with 51 and 21 WSIs respectively. The class distribution was kept approximately the same in both sets. Because the dataset contains twice as much ineffective

treatments (63%) as effective treatments (37%), we tried to mitigate class imbalance. Two options are possible to prevent this issue. The first option is undersampling. It is a technique that keeps all of the data in the minority class and decreases the size of the majority class. The second option is to multiply losses corresponding with the majority class with a factor to increase its influence. We choose to do the latter and multiplied no PSA failure (effective treatment) samples with 2 since we did not want to reduce our dataset even further.

The RESnet18 was trained with batch size 15 and learning rate 0.001 using the Adam optimiser [13]. The binary cross entropy loss function was used as the objective function.

VI. Results

Training was stopped after 40 epochs because the model showed no sign of improvement. The model achieved an average AUC of 0.55. Figure 5 shows the learning curve. The training curve decreases slowly with low variance. In contrast, the validation curve is extremely irregular and does not decrease. The model is both overfitting and underfitting at the same time. This is a strong indication that the model is incapable to detect useful features towards SBRT outcome prediction.

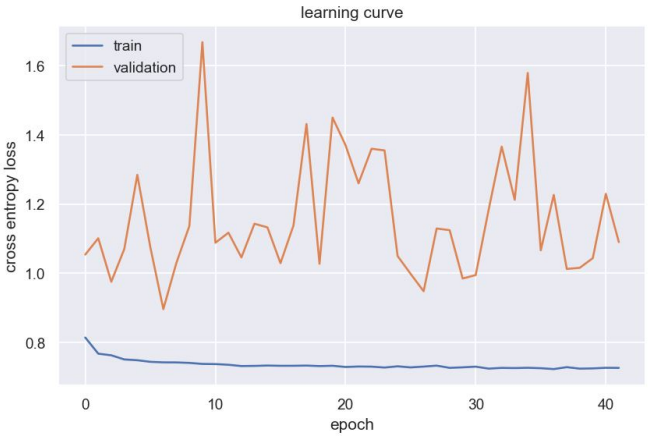

Fig. 5. The learning curve with epochs on the x-axis and binary cross entropy loss on the y-axis.

The output of the model can be interpreted as the probability or certainty towards the PSA failure class. Consequently, each tile corresponds with a value between 0 and 1. We used boxplots to visualise the output distribution of the tiles for each patient and compared the training set with the validation set. The result is shown in figure 6. The blue boxplots correspond with patients with a successful treatment (no PSA failure) and the orange boxplots with the reverse (PSA failure). The distributions in the training dataset show that the model can make a distinction between the two classes. However, the results in the validation set appear more random, further enforcing the observation of overfitting.

As final method of evaluation, we created a heatmap for each WSI were each tile corresponds with its respective output probability towards PSA failure. The goal of the evaluation is

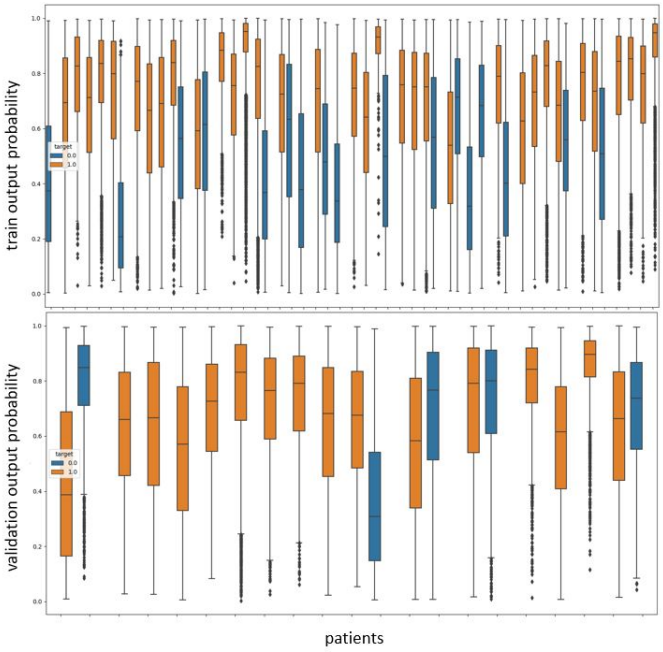

Fig. 6. The output distribution for each patient separately. Blue corresponds with no PSA failure and orange with PSA failure.

to find potential hotspots that could be used for further investigation. Figure 7 gives two examples for from each set for each ground truth label. For the no PSA failure sample from the validation set, every tile outputs low probability without showcasing specific hotspots. Instead of detecting hotspots, the output seems to result in an overall increase or decrease in tile probability. This again strengthens the observation that the model was incapable of detecting predictive features for SBRT outcome.

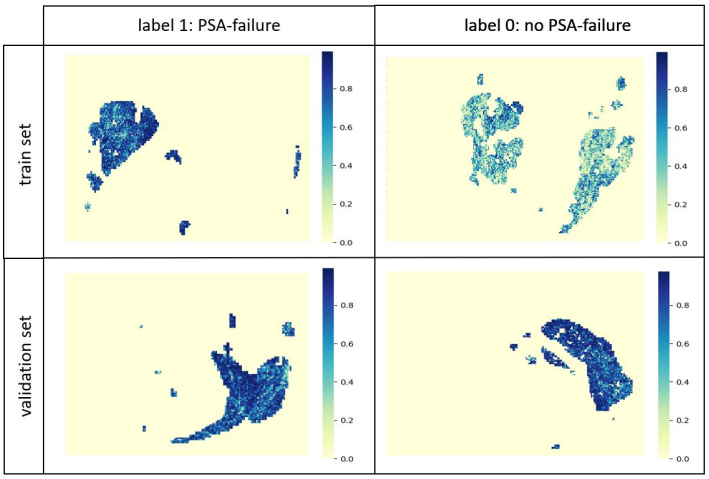

Fig. 7. Output probability visualised in a heatmap for each label and for each set.

VII. Conclusion

We developed a model based on a Unet to automatically detect tumor in H&E stained prostate biopsies. The model achieved an overall average AUC of 0.95 and performs equally well for different ISUP grade groups.

A post-processing pipeline based on heatmap was introduced to reduce the number of false positives and to extract the general ROI for the SBRT response prediction model. Using this method, we were able to achieve a dice similarity score of 0.82 and precision 0.92 on an external dataset, ensuring the generality of the segmentation pipeline.

The SBRT response prediction model based on coarse grained labels was after extensive evaluation not able to extract relevant features to predict the outcome. The model achieved an average AUC of 0.55.

VIII. Future work

To increase the overall performance of our segmentation model, different approaches can be taken. First, Reinhard normalisation is very dependent on the target image chosen. If this image is too different in distribution from the image to be normalised, it can fail. Improved techniques such as the method proposed by Macenko et al. [15], could partially alleviate this problem and in doing so improve the segmentation model. Another approach would be shift our single-model approach to a multi-model approach. Li et al. [16] compared 10 different approaches towards tumor segmentation in lung cancer and concluded that multi-model achieved on average a higher performance.

In recent years, attention-based Multiple Instance Learning (MIL) has gained in popularity in WSI analysis [17]. MIL is a weakly supervised technique where a single class label is assigned to a bag of instances. But instead of processing the instances independently, MIL processes the bag in its entirety using an advanced pooling technique. The method maintains the contextual information that is lost when processing the tiles independently. Attention-based MIL can be used for SBRT response prediction in WSIs.

IX. Acknowledgement

The external SBRT dataset used in this work is provided by Johns Hopkins and Ghent University.

X. Implementation and hardware

All methods were implemented using Python. Moreover, deep learning models were implemented using Pytorch. Training and inference were performed on one GeForce GTX 1080 Ti GPU with 12 Gb RAM.

References

[1] L. E. Schnipper, N. E. Davidson, D. S. Wollins, C. Tyne, D. W. Blayney, D. Blum, A. P. Dicker, P. A. Ganz, J. R. Hoverman, R. Langdon, G. H. Lyman, N. J. Meropol, T. Mulvey, L. Newcomer, J. Peppercorn, B. Polite, D. Raghavan, G. Rossi, L. Saltz, D. Schrag, T. J. Smith, P. P. Yu, C. A. Hudis, and R. L. Schilsky, "American society of clinical oncology statement: A conceptual framework to assess the value of cancer treatment options," *Journal of Clinical Oncology*, vol. 33, pp. 2563–2577, 8 2015.

[2] A. Sharma and R. Rani, "A systematic review of applications of machine learning in cancer prediction and diagnosis," *Archives of Computational Methods in Engineering*, vol. 28, pp. 4875–4896, 12 2021.

[3] A. Shalimova, V. Babasieva, V. N. Chubarev, V. V. Tarasov, H. B. Schiöth, and J. Mwinyi, "Therapy response prediction in major depressive disorder: Current and novel genomic markers influencing pharmacokinetics and pharmacodynamics," pp. 485–503, 6 2021.

[4] Y. Lu and J. Han, "Cancer classification using gene expression data," *Information Systems*, vol. 28, pp. 243–268, 6 2003.

[5] F. Galati, V. Rizzo, R. M. Trimboli, E. Kripa, R. Maroncelli, and F. Pediconi, "Mri as a biomarker for breast cancer diagnosis and prognosis," *BJR—Open*, 5 2022. [Online]. Available: https://www.birpublications.org/doi/10.1259/bjro.20220002

[6] S. E. Cohen, J. B. Zantvoord, B. N. Wezenberg, C. L. Bockting, and G. A. van Wingen, "Magnetic resonance imaging for individual prediction of treatment response in major depressive disorder: a systematic review and meta-analysis," *Translational Psychiatry*, vol. 11, 6 2021.

[7] A. D. Pizzi, A. M. Chiarelli, P. Chiacchiaretta, M. d'Annibale, P. Croce, C. Rosa, D. Mastrodicasa, S. Trebeschi, D. M. J. Lambregts, D. Caposiena, F. L. Serafini, R. Basilico, G. Cocco, P. D. Sebastiano, S. Cinalli, A. Ferretti, R. G. Wise, D. Genovesi, R. G. Beets-Tan, and M. Caulo, "Mri-based clinical-radiomics model predicts tumor response before treatment in locally advanced rectal cancer," *Scientific Reports*, vol. 11, 12 2021.

[8] P. G. Conaghan, M. Østergaard, O. Troum, M. A. Bowes, G. Guillard, B. Wilkinson, Z. Xie, J. Andrews, A. Stein, D. Chapman, and A. Koenig, "Very early mri responses to therapy as a predictor of later radiographic progression in early rheumatoid arthritis," *Arthritis Research and Therapy*, vol. 21, p. 214, 10 2019. [Online]. Available: https://arthritis-research.biomedcentral.com/articles/10.1186/s13075-019-2000-1

[9] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," pp. 436–444, 5 2015.

[10] "Prostate cancer grade assessment (panda) challenge — kaggle." [Online]. Available: https://www.kaggle.com/c/prostate-cancer-grade-assessment

[11] E. Reinhard, M. Ashikhmin, B. Gooch, and P. Shirley, "Color transfer between images," *IEEE Computer Graphics and Applications*, vol. 21, pp. 34–41, 9 2001.

[12] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, 1986.

[13] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization." International Conference on

Learning Representations, ICLR, 12 2015.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," vol. 2016-December. IEEE Computer Society, 12 2016, pp. 770–778.

[15] M. Macenko, M. Niethammer, J. S. Marron, D. Borland, J. T. Woosley, X. Guan, C. Schmitt, and N. E. Thomas, "A method for normalizing histology slides for quantitative analysis," 2009, pp. 1107–1110.

[16] Z. Li, J. Zhang, T. Tan, X. Teng, X. Sun, H. Zhao, L. Liu, Y. Xiao, B. Lee, Y. Li, Q. Zhang, S. Sun, Y. Zheng, J. Yan, N. Li, Y. Hong, J. Ko, H. Jung, Y. Liu, Y. C. Chen, C. W. Wang, V. Yurovskiy, P. Maevskikh, V. Khanagha, Y. Jiang, L. Yu, Z. Liu, D. Li, P. J. Schuffler, Q. Yu, H. Chen, Y. Tang, and G. Litjens, "Deep learning methods for lung cancer segmentation in whole-slide histopathology images - the acdc@lunghp challenge 2019," *IEEE Journal of Biomedical and Health Informatics*, vol. 25, pp. 429–440, 2 2021.

[17] M. Y. Lu, D. F. Williamson, T. Y. Chen, R. J. Chen, M. Barbieri, and F. Mahmood, "Data-efficient and weakly supervised computational pathology on whole-slide images," *Nature Biomedical Engineering*, vol. 5, pp. 555–570, 6 2021.

Contents

List of Figures

List of Tables

List of Abbreviations

WSI Whole Slide Image

SBRT Stereotactic Body Radiation Therapy

ROI Region Of Interest

PSA Prostate Specific Antigen

ADT Androgen Deprivation Therapy

AI Artificial Intelligence

ANN Artificial Neural Network

CNN Convolutional Neural Network

RESnet RESidual neural network

H&E Haematoxylin and Eosin

1 Introduction

1.1 Problem statement

Cancer has a major impact on today's society. According to the World Health Organisation (WHO), cancer is a leading cause of death worldwide, accounting for nearly 10 million deaths in 2020 [2]. The probability of being diagnosed with cancer is around 40% [3]. New insights in cancer can be very valuable in future development of treatments and can potentially be life saving.

This work aims to gain such valuable insights on a new treatment, Stereotactic Body Radiation Therapy (SBRT), against prostate cancer. It targets the tumor in the prostate with high-precision, high-dose radiation beams. This minimises the damage dealt to the surrounding tissue. It also beneficial for patients because SBRT requires far less sessions to achieve the same result as traditional radiation therapies. The effectiveness however is subject to high variance. Some patients react very well to the treatment while others do not experience improvement. This means that some patients receive a treatment that is almost ineffective. Since SBRT is a radiation technique, this implies that the patient is unnecessary exposed to radiation which is a non-negligible disadvantage. Unfortunately, the cause of non-responding tumors is not known.

We investigate the possibility to predict the outcome of SBRT using state-of-the-art deep learning

techniques on microscopic cancer tissue taken before the effective treatment. By focusing on the interpretability and feature extraction of such models, we could gain interesting insights in the cause of the observed variability such that the patients can receive a more personalised and effective treatment.

The work performed can be divided into two major parts: segmentation and therapy response prediction. Segmentation focuses on techniques to automatically detect tumor regions on microscopic cancer tissue. This process is performed under the assumption that the cause of the observed variability is found in the tumor and its immediate surroundings. Consequently, benign tissue has no value towards therapy response prediction. The therapy response prediction module builds further upon the results of the segmentation module and focuses on techniques to predict the SBRT effectiveness. Figure 1.1 represents a high-level overview of this thesis.

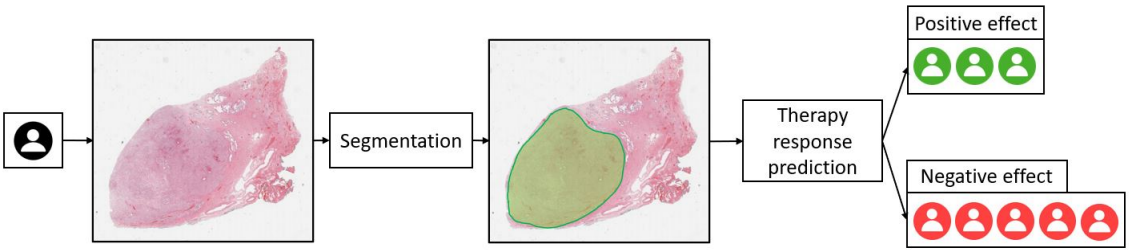

Figure 1.1: High-level overview

1.2 Main contributions

The main contributions of this thesis are:

- The development of a method able the automatically detect the tumor on microscopic images of tissue with high detail and with a focus on robustness. The method was tested on an external dataset with different characteristics to ensure generality.

- The development of a method that converts the segmentation maps into a general region of interest, with focus on removing false positives.

- The evaluation of different techniques which extract features from microscopic tissue to assess if they can be used as a predictive biomarker for the outcome of SBRT.

1.3 Outline

The rest of this work is organised as follows. Chapter 2 gives an overview of cancer mechanisms which are important to understand the challenges in later chapters. In chapter 3, we give an overview of the relevant methodologies in deep learning. Special focus is given to the models used in later chapters.

In chapter 4, the data-types and -sources used to train and evaluate our models are discussed. Chapter 5 provides an in-depth explanation of the design choices made during development of the segmentation process. Chapter 6 evaluates the results of the proposed models in chapter 5. Chapter 7 gives an overview of the different approaches tested and evaluated towards therapy response prediction. Finally, we conclude our work in chapter 8.

2

Cancer

Cancer is one of the most widespread diseases worldwide, accounting for nearly 10 million deaths in 2020 [2]. However, the disease affects even more people in their daily lives. It has been estimated that the probability of being diagnosed with cancer during the lifetime of an individual is around 40% [3]. Currently, many researchers focus on developing and improving technologies for faster diagnosis and better treatment. Recent advancements from the digital age have allowed information to be shared and accessed with relative ease, speeding up research. Specifically, the evolution and development of new cutting-edge technology have enabled the collection of huge amounts of valuable data. This paves the way for new insights that may very well lead to breakthroughs in cancer research.

2.1 What is cancer?

2.1.1 Cause

The human body consists out of trillions of cells each with their own function. Together they form a complex system to perform various specialised tasks. Cancer begins when one of the cells in an organ or tissue starts to malfunction. Malfunctioning cells, due to either damage or mutation, break free from the normal controls that allow our cells to work together in harmony.

A normal cell will divide only when it receives an extracellular chemical signal, called **mitogens** [4]. The signal is processed in the nucleus causing the cell to reproduce their genetic information and divide into two daughter cells through a process called **mitosis**. In addition, the human body also tells the cells when to stop dividing. This prevents too many cells from being made. However, cancer cells reproduce uncontrollably, regardless if they received chemical signals or not. Because malignant cells break free from the control of the human body, the amount of times the cells can divide has no limit and controlled cell death or **apoptosis** does not apply. This can lead to a mass of cells that accumulate to form a tumor [5].

Cells are composed of different components or organelles. One of the most important organelles, the **nucleus**, can be thought of as the brains of the cells. It is here that genetic information is stored in chromosomes. Each chromosome contains individual units or **genes** which, at a chemical level, consist of DeoxyriboNucleic Acid (**DNA**). They contain the blueprint that defines the function of the respective cells. All cancers are thought to result from changes in the DNA which are referred to as **mutations**. Mutations can be initiated through a variety of causes. Examples include chemicals that can be swallowed or inhaled such as those found in cigarette smoke. Consequently, people who smoke have a higher chance of developing lung cancer [6]. Another example that can cause mutations is radiation of the sun, resulting in skin cancer [7]. And sometimes mutations occur without any known external cause. Relevant mutations causing cancer occur in two types of genes.

Proto-oncogenes are genes that are involved in regulating normal cell division. When mutated, they are referred to as oncogenes. As such, the mutated cell starts to divide in the absence of proper signals. To stop this phenomenon, the cell also contains genes that are responsible for initiating apoptosis and are known as **tumor suppressors**. Cancer cells with sustained damage in proto-oncogenes and tumor suppressors, start to divide uncontrollably [5].

Because a multitude of different mutations can occur at the same time, the resulting tumor is a heterogeneous mass of cells [8]. This heterogeneity is one of the reasons why it is often so difficult to remove a tumor completely. The tumor may harbour cells with different levels of sensitivity to treatments. Consequently, it is possible that parts of the tumor are resistant to specific drug types.

An additional challenge is **metastasis**. The tumor sends signals asking for nutrients through **angiogenesis**. These messages cause nearby blood vessels to send over new extensions that deliver food and oxygen. Additionally, blood vessels serve as a passageway for cancer cells. The cancer cell travels through the blood vessel system to eventually nestle itself into a completely new part of the body. This creates a new tumor originating from the already existing one. Metastasis in the human body reduces the chance of survival depending on the location of the new tumor. It is therefore crucial to diagnose tumors before they evolve to that stage.

2.1.2 Treatments

Different types of cancer exist and no treatment is effective against all types and in every condition. Doctors attempt to create a specific treatment plan for each individual patient. It is not unusual for a patient to undergo different kinds of treatments in order to slow down tumor growth or to completely remove the tumor. In the following a non-exhaustive list is given of different treatments [9]:

- **Chemotherapy**: In chemotherapy, anti-cancer drugs (chemotherapeutic agents) are used to destroy cancer cells. The drugs target different phases in the cell cycle. They cannot make a distinction between healthy cells and cancer cells so this therapy is about minimising the healthy cell damage and maximising the cancer cell removal. This treatment is used for slowing down growth of tumors, complete removal of tumors and lessen the chance of a potential return of a tumor [10, 11].
- **Radiation therapy**: High doses of radiation are used to kill cancer cells and shrink tumors. This treatment specifically targets the DNA in cancer cells to stop their growth. Cancer cells whose DNA is damaged beyond repair stop dividing and die. This treatment needs to be performed with high precision to minimise the damage done to the surrounding benign tissue. This entire process takes multiple radiation sessions over the course of several weeks. [10].
- **Hormone therapy**: Some cancer types use hormones in order to grow. To stop this growth, hormone therapy alters or blocks the involved hormones. The used drugs travel throughout the human body to bind to the hormones. In this sense the treatment differs from the other treatments because they target only a specific part of the body. [10]
- **Immunotherapy**: The immune system attempts to trace and destroy foreign substances in the body, but cancer cells are sometimes capable of circumventing this system. Immunotherapy is a treatment that alters and/or boosts the immune system so that it becomes better capable of detecting and destroying cancer cells. [10]
- **Surgery**: Surgery is a procedure where the surgeon removes cancer directly from a patient. Either the complete tumor is removed or only a part of the tumor is removed because otherwise the organs are damaged as well. [10]

The treatment under investigation in this thesis is a specific radiation therapy and will be explored further on in this chapter.

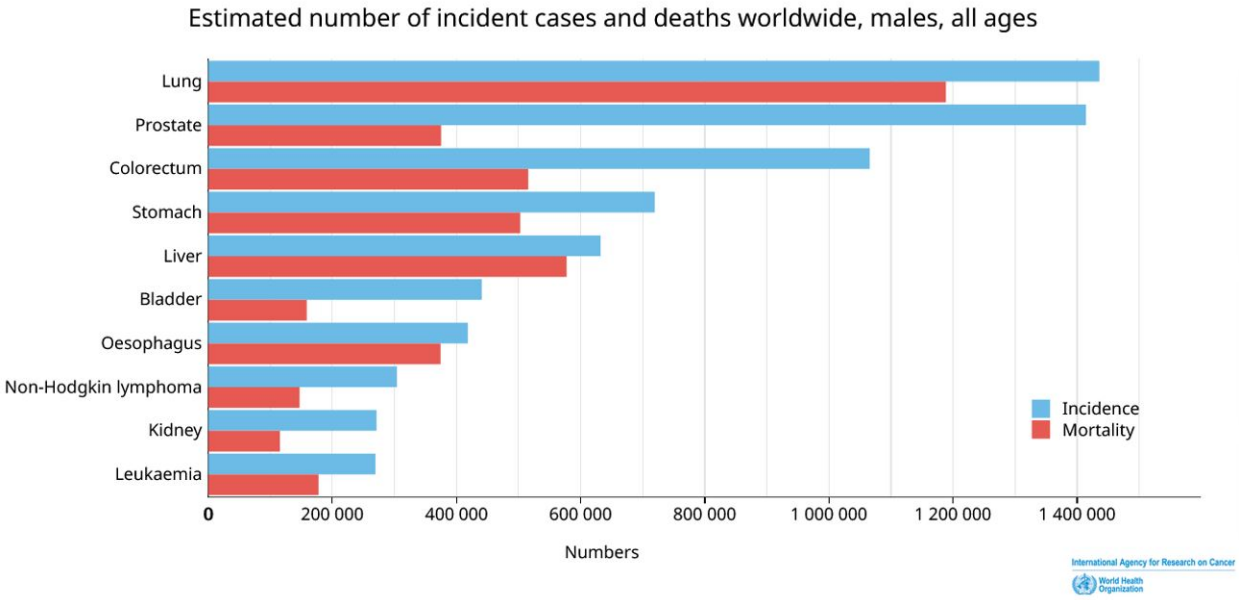

Figure 2.1: Number of occurrences grouped per cancer type [12]

2.2 Prostate cancer

Prostate cancer is one of the most common cancer diagnoses in men, just after lung cancer (figure 2.1). It is the fifth leading cause of death worldwide [13]. In 2020, approximately 1 400 000 new cases were reported and it caused approximately 300 000 deaths [12]. The development of prostate cancer is strongly correlated with age, with the highest incidence found for men over 65 years old. For unknown reasons, the incidence and mortality is the highest for African-Americans [14]. Other risk factors include: obesity and family history (genetic factors).

A developing tumor in the prostate causes it to dysfunction (figure 2.2) leading to difficulties in urinating, blood in urine or semen, pain and erectile dysfunction [15]. In recent times, prostate cancer can often be treated successfully. However if the cancer spreads (metastasis) new tumors can form, increasing the potential threat. This is why it is important to detect prostate cancer as early as possible. Since prostate cancer has no apparent symptoms in its early stages, early detection is a challenging task. As a result most prostate cancers are detected during standard screening before symptoms even occur.

Screening is done using a prostate-specific antigen (PSA) blood test and is performed for people with multiple risk factors. PSA is a protein made by both normal cells and cancer cells in the prostate gland. Because of the uncontrolled growth of cancer cells, PSA levels in blood typically rise indicating that a patient might have prostate cancer. There is no clear boundary that distinguishes normal PSA levels from abnormal PSA levels but typically doctors will start performing extra tests when the PSA is 4 nanogram per millilitre (ng/ml). Multiple elements

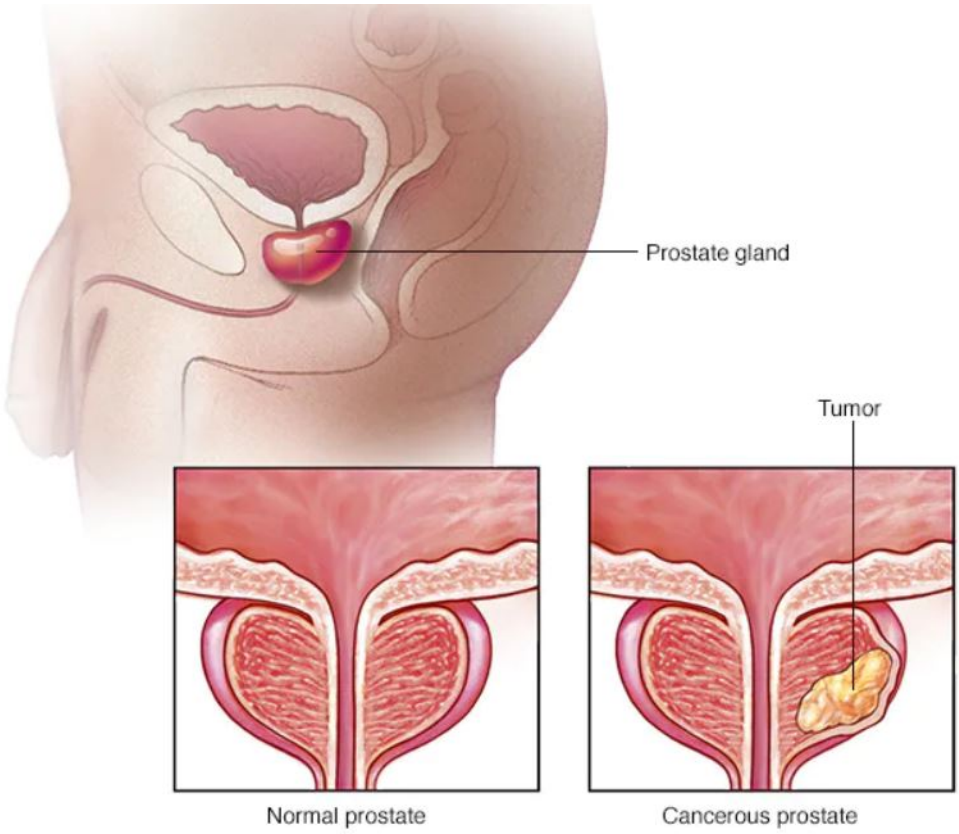

Figure 2.2: Illustration of prostate cancer [15]

affect PSA levels making it difficult to rely fully on that. People with enlarged prostates and older people for example will naturally produce more PSA. In contrast other factors lower the PSA levels, e.g. certain medicines [16].

Based on the stage the prostate cancer is found, different treatments exist. For example in some cases a doctor decides that the tumor could disappear on its own and will keep monitoring the evolution of the tumor. This is referred to as **expectant management**. Other treatments include **surgery** and **radiation therapy**.

2.2.1 Gleason grade

Gleason scores determine the degree in which the prostate cancer is a threat. Some tumors are not dangerous and are merely an inconvenience while others are more aggressive and have the potential to be deadly. When it comes to prostate cancer, **Gleason grade** is the golden standard to quantise the severity. This helps a doctor to determine which treatment would be appropriate [17].

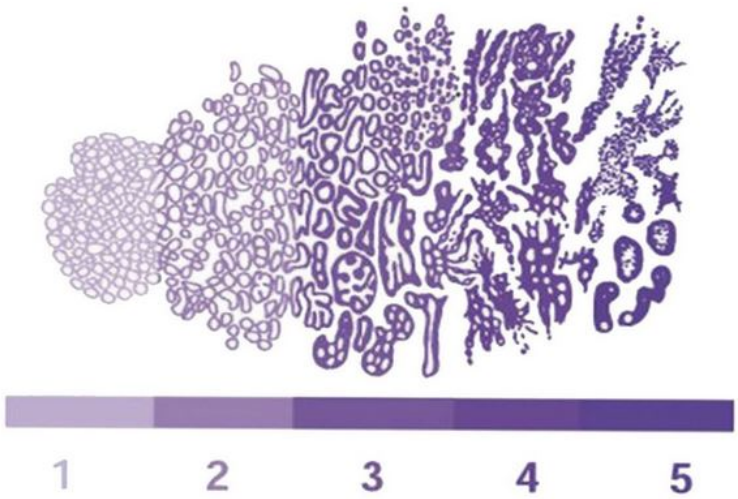

Figure 2.3: Schematic overview of the Gleason grade scoring system [18]

The Gleason grade is determined by pathologists using microscopic images of several samples of cells (biopsies) from the tumor. They assign a score to each cluster of cancer cells based on the overall shape and condition. Figure 2.3 illustrates this concept. In essence this score gives an indication on how similar clusters of cancer cells are to normal cells. A score of 1 and 2 indicate normal prostate cell clusters with the latter having more stroma between the glands. The upper spectrum ranging from 3 to 5 have increasingly more irregular cell clusters and are typically cancer cells. Examples for each grade are presented in figure 2.4 [17].

Given the heterogeneous nature of tumors, it comes with no surprise that multiple scores can be given to multiple parts of the tumor. In fact, the Gleason grade combines the two most frequent appearing scores. For example a patient with a high percentage of Gleason score 3 followed by Gleason score 4 will receive a Gleason grade 3+4 or 7. This way the grades represent a more nuanced picture. Everything below Gleason grade 6 is considered benign and does not require immediate treatment. An overview is given in table 2.1.

Note that there are no hard boundaries between the different Gleason scores and grades. Grading is partly subjective and can differ from pathologist to pathologist. Doctors often request the opinion of multiple pathologists in order the determine the best treatment possible.

Gleason grade	**ISUP grade group**	**meaning**
Gleason grade 6 ($3+3=6$)	Grade group 1	Well defined circular glands. The tumor is likely to grow very slowly, if at all.
Gleason grade 7 ($3+4=7$)	Grade group 2	Well defined circular glands. Some glands start to fuse. The tumor is likely to grow slowly.
Gleason grade 7($4+3=7$)	Grade group 3	Most glands start to fuse. Some circular glands. The tumor is likely to grow at a moderate rate.
Gleason grade 8 ($4+4=8$)	Grade group 4	Glands fuse but are still recognisable. The tumor might grow quickly or at a moderate rate.
Gleason grade 9 or 10 ($4+5=9$, $5+4=9$, $5+5=10$)	Grade group 5	No recognisable glands. Loose or strings of cells. The tumor is likely to grow quickly.

Table 2.1: The Gleason grades with its respective ISUP grade group. [1]

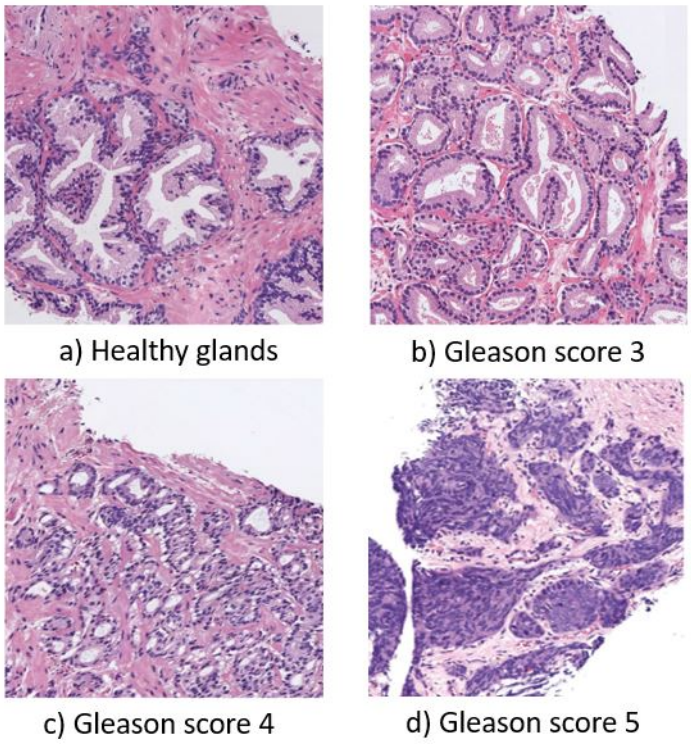

Figure 2.4: Examples of the Gleason grading system. a) Non-cancerous well differentiated glands, b) Gleason score 3, c) Gleason score 4 containing large fused granular patterns, d) Gleason score 5 containing nested cells in irregular formation. The images originate from Rodriguez et al. [17]

2.3 Stereotactic body radiation therapy

Stereotactic body radiation therapy (SBRT) is a relative new technique that is not as widespread as other treatments. SBRT delivers high doses of radiation with high accuracy onto the patients' body. The key difference between conventional radiation therapies and SBRT is in how the radiation is delivered [19]. The former delivers small doses of radiation performed over several weeks. Because SBRT is more targeted, doctors can deliver much higher doses of radiation each session. The treatment can be finished within five sessions.

A high dose of radiation can be damaging to healthy tissue. This is why a high accuracy is needed to minimise the damage done to surrounding tissue. Using sophisticated computerised images (e.g. CT-scan, MRI-scans or other imaging techniques) not only the precise location of the tumor is determined but also the shape and size. The accuracy is then achieved by applying radiation beams containing small doses from different angels onto the patients body. The tumor should lay at the cross-section of all those beams raising the combined doses where the tumor is located. This way collateral damage is minimised while still making effective use of radiation.

Figure 2.5 visualises the principle of using different radiation beams.

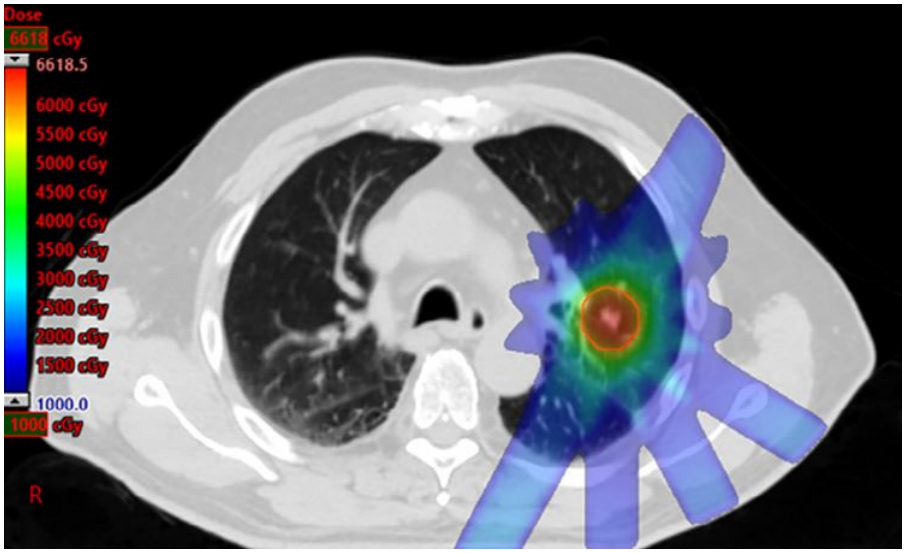

Figure 2.5: SBRT illustration [20]

3

Machine learning

Artificial Intelligence (AI) entails the technology that enables machines to perform tasks which require human thinking and intellect, like e.g. decision-making, speech recognition and visual perception. Advancements in technology boosted the collection and processing of data in almost every part of our society. The growth of available data and the increase in processing power prompted the introduction of AI into various fields: agriculture, retail, security, sport, health care etc. In health care, AI can assist doctors and contribute towards diagnosis, decision making, uncovering patterns and insights that humans could not find on their own.

In 1965, extensive research bore the fruit of making the first problem-solving program, Dendral [21]. The program was used to analyse chemical substances and hypothesise about the molecular structure. Dendral's performance rivalled that of chemists experts at this task, making it a valuable tool in the industry. This would inspire the next generation of AI applications in the medical field. In 1986, the knowledge-based system Eklavya [22] was created to assist a community health worker in dealing with symptoms of illness in toddlers. A few years later, systems like IBM's DeepBlue and Watson emerged, providing tools to help clients facilitate medical research and health care solutions through AI [23]. More recently in 2020, Google DeepMind applied AI to solve the 'protein folding problem' that existed for over fifty years and was successful in predicting a protein's three dimensional structure of its amino-acid sequence [24].

In this chapter, we introduce and discuss machine learning, a subset of AI systems. Machine

learning entails the set of programs or systems capable of performing a given task without explicitly programming any rules it has to follow. The programs learn on their own based on examples to perform their tasks. Machine learning on itself contains an even smaller subset: deep learning. This subset contains systems that attempt to mimic the human brain and are referred to as neural networks. A schematic representation is given in figure 3.1. First an overview is given of machine learning, discussing the different learning approaches. In the next section, one of these learning approaches, supervised learning, will be more in-depth explained. Finally, the basics of neural networks are given to introduce more advanced models in this thesis.

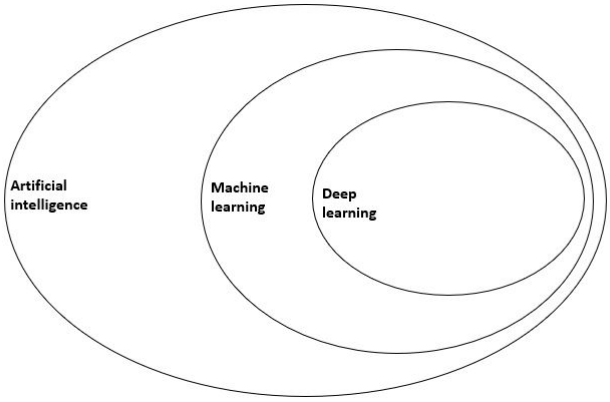

Figure 3.1: The field of AI

3.1 Machine learning

Machine learning models have the ability to learn from data without being explicitly programmed. A model consists of three main elements: a parametric mathematical model, a learning algorithm and an objective function [25]. The goal of a parametric model is to optimise (minimise/maximise) an objective function using a learning algorithm. Typically the objective function acts as a loss function that indicates the general performance of the model. To avoid explicit programming, the model needs to learn using examples. These examples or data represent a very important part of artificial intelligence and can greatly influence the performance of the model.

The ultimate goal of a model is to learn from the data it has seen. However, it should still perform well on unseen data. This can only be achieved if the data used to train and learn, represents the outside 'world' in good fashion. Data should be the sampled independently and identically distributed from the 'world'. This is called the i.i.d. assumption.

Machine learning can be divided into different categories based on the required task. Figure 3.2 represents the different categories with their most common applications [25]:

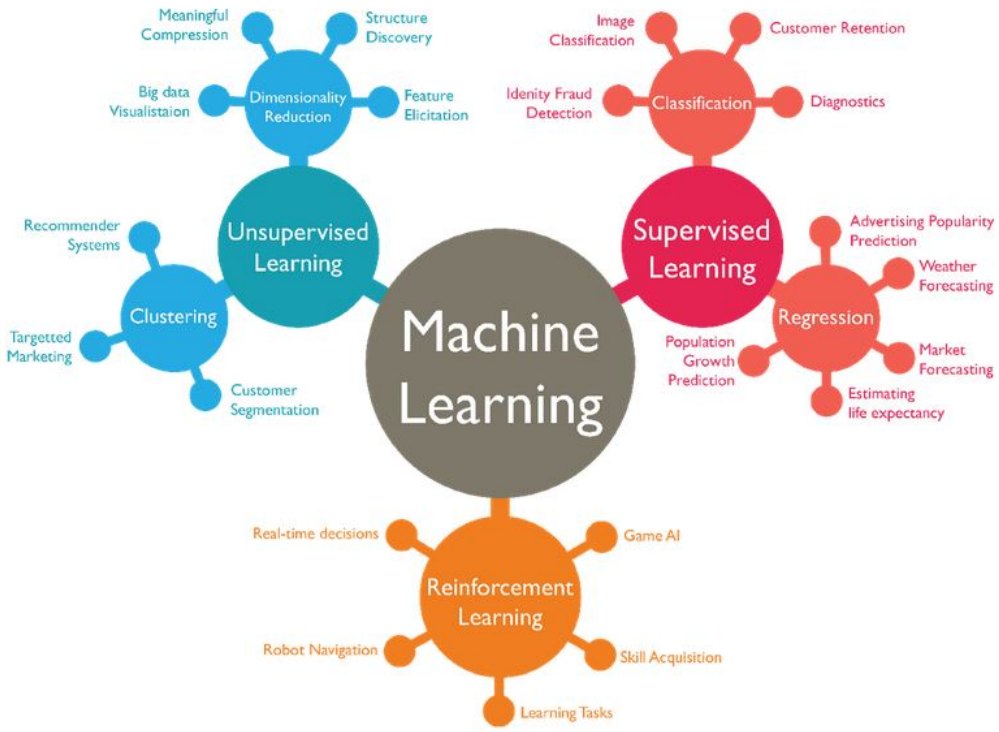

Figure 3.2: The general task types within machine learning [26]

- **Reinforcement learning**: Reinforcement learning is the training of models to make a sequence of decisions in an unknown environment. This is often used for models learning to play games (e.g. robot navigation). The model learns over multiple iterations. Each time a model successfully completes a task, it gets rewarded. If it makes a mistake it gets punished, reinforcing what it has learned earlier. By doing this, the model will improve each iteration by maximising future rewards.

- **Supervised learning**: This category is defined by its use of labeled dataset to train models. Supervised learning is used for classification and regression tasks. For classification, the model receives input data and attempts to classify the input in one of the given classes. While classification has discreet output values, regression is used to fit a function to the given input data. This means that the output values are per definition continuous. The input data is always combined with its respective output labels.

- **Unsupervised learning**: Contrary to supervised learning, labels are not used in unsupervised learning. Instead the goal of unsupervised learning is to find correlations between given data points. It is often used for clustering and dimensionality reduction tasks. The model receives input data but has no expected output labels.

3.2 Supervised learning

In this thesis we will almost exclusively make use of supervised learning techniques because the research question is inherently a classification task: classifying the response on SBRT using specific input data. A patient can be divided into two categories: a positive response of the patient (class 1) or a negative response of the patient (class 2).

Formalisation

A supervised machine learning model requires input data X and expected output labels R. The goal is to approximate an unknown target function $f(x) : x \rightarrow r$ which associates the input $x \in X$ to the output $r \in R$. The training data $(X, R) : \quad (\mathbf{x}_1, r_1), (\mathbf{x}_2, r_2), ..., (\mathbf{x}_N, r_N)$ is a collection of input vectors with their corresponding labels. Note that these labels are discrete for classification tasks and continuous for regression tasks.

The ideal machine learning model g is defined by input data X and tunable parameters θ. g should approximate the true function f.

$$g : \mathbf{x}, \theta \rightarrow y = g(\mathbf{x}, \theta) \tag{3.1}$$

Using this, we want to construct hypothesis h capable of performing the task on unseen data. The hypothesis h is the result of training using (X, R). It approaches the function g with parameters θ^*.

$$y = h(\mathbf{x}|\theta^*, X, R) \tag{3.2}$$

By performing a specific optimisation process, we approximate the optimal parameters θ for the supervised task using training data (X, R) as examples.

Data

A model does not only need to be trained but also validated and evaluated. One should make sure that the trained model is capable of performing well on completely unseen data. To this end, a dataset is divided into three non-overlapping subsets:

- **Training dataset**: This subset is mainly used for training and optimizing the tunable parameters θ of the model.

- **Validation dataset**: Typically multiple models are a good option for a specific task. To find the best model and to tune its hyperparameters, the validation subset is used.

- **Test dataset**: After the chosen model is trained with its optimal hyperparameters, tests are performed to ensure the quality and performance of the model. This subset should not be used or even viewed during development of a model and should only be used as a final test.

Figure 3.3 illustrates the typical workflow. Stage 1 is the development phase during which the dataset is split into the three subsets. After the model is chosen and the hyperparameters are chosen, one can train and validate the model. In stage 2, the validation and train datasets are combined again to perform a final training session after which the model performance can be estimated using test data.

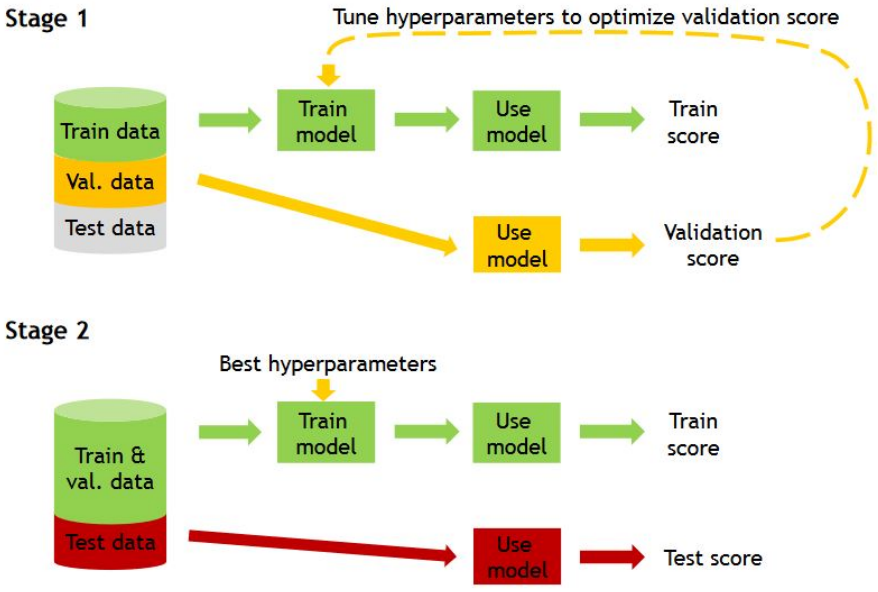

Figure 3.3: The typical workflow with the three splits [25]

Creating the splits needs to be done with caution. If the training subset gets too small, the model may not have enough samples to generalise with. A small validation subset may lead to poor model and/or hyperparameter choices because the validation score is not a good estimate of generalisation performance. A small testset may statistically not be enough to prove the generalisation performance. It is also important to avoid any kind of correlation between the sets. An example of this are tweets from the same person in different subsets. This leakage can cause the model evaluation to be optimistic and not be a good representation of the overall performance. Consequently it causes the model to perform worse on unseen data because of this model performance overestimation.

3.3 Deep learning

We mentioned before that deep learning is a subset of machine learning. Deep learning distinguishes itself from other machine learning models by the use of so-called Artificial Neural Networks (ANN). The previous section however still applies with supervised learning still the focus.

First off, we discuss the basic implementation of an ANN: architecture, gradient descent, backpropagation and loss functions. Afterwards we will dive deeper into more complex models: Convolutional neural networks (CNN), Residual neural networks (RESnet) and Unets. We finish this section with model evaluation methods.

3.3.1 Artificial neural network

An ANN can be seen as an abstraction of the human brain. The brain consists of millions of neurons, each with an input and output. The great interconnectivity between these neurons enable the brain to learn and adapt to its environment. This architecture allows not only parallel processing of information but also guarantees robustness to noise and failures. These properties are very desirable in machine learning. For this reason neural networks are modelled after the human brain.

Architecture

Assume that a simple ANN is used to classify images of handwritten digits. In figure 3.4 an image of the number three (in theory your brain should have classified this already) is used as raw input of a neural network. Each pixel of the image corresponds with a neuron in the input layer. The input layer is connected with one or multiple hidden layers. Hidden layers perform transformations which allow to detect patterns in the input. An ANN is per definition a black box model, meaning that the transformations in the hidden layer typically do not have any meaning for a human. But for this example, maybe the hidden layer detects specific edges in the image that correlate respectively with one or multiple output neurons. Finally the information is condensed into the output layer. This layer contains 10 neurons each representing the final prediction of the number. Ideally the neuron representing 3 should be activated, indicating that our brain is indeed correct.

Each neuron is fully connected with all the neurons in the next layer. The connections enable the network to ”send information” to the next layer. A weight is associated with each branch in the network. This weight indicates how relevant both neurons are with respect to each other.

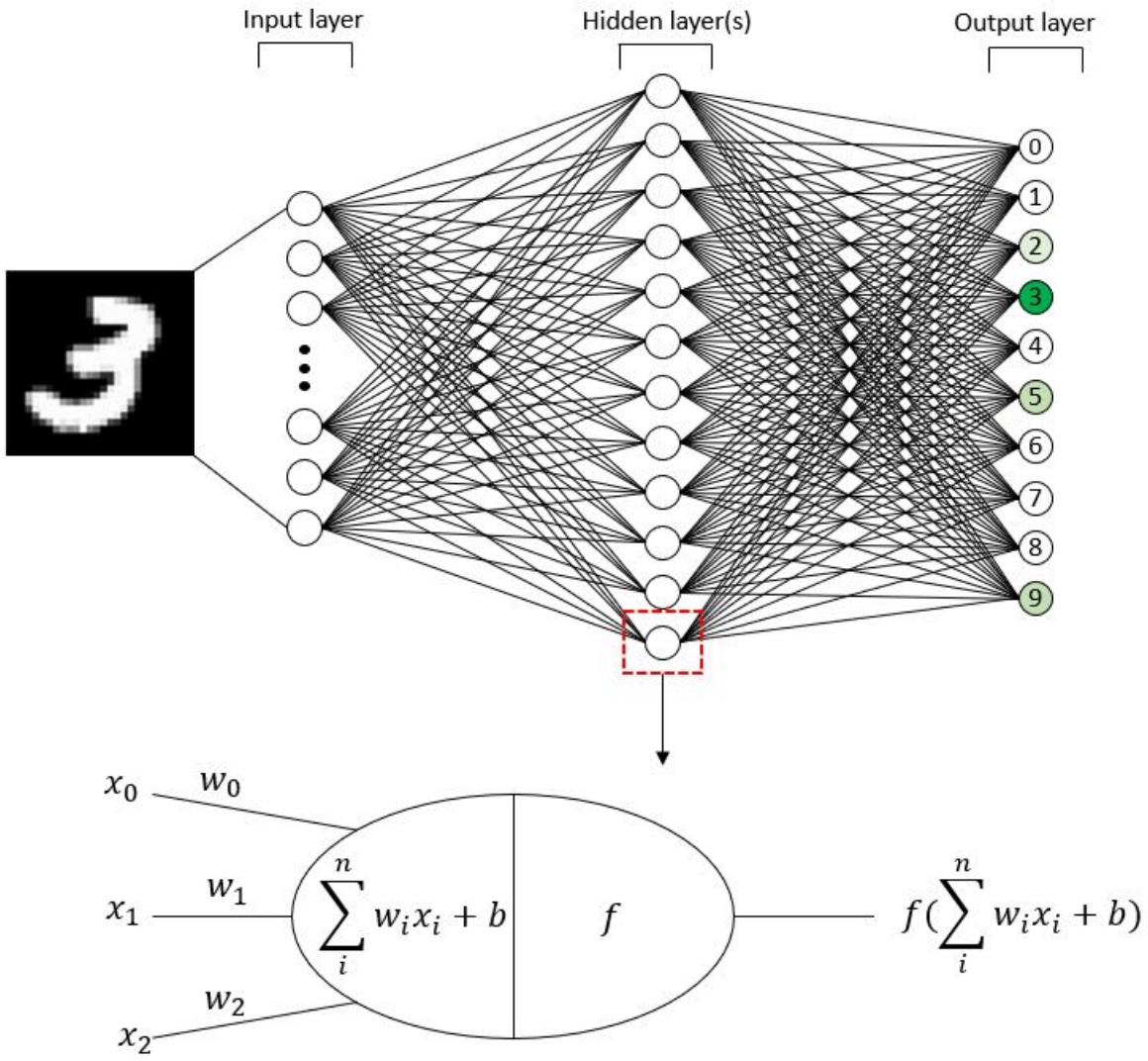

Figure 3.4: The core element of an artificial neural network

The learning phase attempts in fact to tweak those values to obtain an accurate and reliable prediction of the input.

Let us now take a look at one neuron in particular (figure 3.4). A typical neuron has an input vector $X \in \Re$ and a weight vector $W \in \Re$ related with each other in a one-on-one relation. The combined input of each neuron is a weighted sum $\sum_i w_i x_i + b$. To model non-linearities, the weighted sum is put through an activation function f. The activation function can be any function that is non-linear and differentiable. An example of such an activation function is the sigmoid function $f : \Re \to [0, 1]$:

$$f(z) = \sigma(z) = \frac{1}{1 + e^{-z}} \tag{3.3}$$

It is a continuous function mapping all values $\in \Re$ between zero and one. However in practice, another activation function is used called the ReLu-function

$$f(z) = R(z) = max(0, z) \tag{3.4}$$

Note that this function is not bounded by one and just maps negative values to zero. The reason ReLu is preferred over the sigmoid function has to do with a problem called vanishing gradient. We will explain this in greater detail later on in this chapter. For now, it is important to realise that ReLu partially resolves difficulties in optimisation of deep networks. Figure 3.5 compares the two functions. Finally, the bias b in the weighted sum is introduced to shift the activation function to its desired position based on the absolute values of the weighted sum of the neuron.

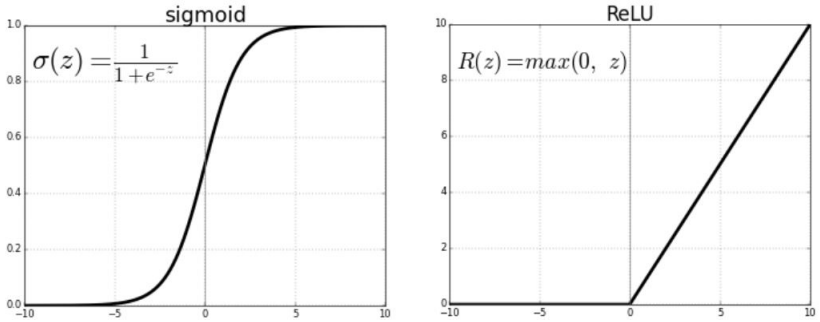

Figure 3.5: a) The sigmoid function, b) the ReLu function

All of this leads to a combined output value

$$y = f(\sum_i w_i x_i + b) \tag{3.5}$$

Each neuron in the network makes this calculation. Based on a specific set of weights and biases, the input image of the number three can be transformed such that the neuron corresponding with the number three in the output layer has the highest activation when compared to the other output neurons.

As mentioned before, this network does not need explicit programming in order to function. The weights and biases are determined through a learning and optimisation process and are randomly initialised.

Loss function

First an objective function needs to be chosen so that the performance can be calculated for the model (as seen with supervised learning). Learning in this environment is equivalent to

optimising this objective function (also called loss function) for a specific problem. Taking the example of the numbers again, a good loss function assigns a high cost to outputs that indicate the model is uncertain or wrong about the results. A vector output $[0.5, 0.5, ..., 0.5, 0.5]$ should have a relatively high cost. At the same time, if the model is certain about a number e.g. $[1, 0, 0, ..., 0, 0]$, a low cost should be assigned if the number is correct and a high cost should be assigned for the wrong number.

An example of a loss function meeting the requirements is the following:

$$L = -\frac{1}{N}[\sum_{j=1}^{N}(t_j log(p_j) + (1 - t_j) log(1 - p_j)] \tag{3.6}$$

Where t_j is either 1 or 0 depending on the ground truth and p_j is the output probability of the corresponding class of the ANN. By calculating the loss of each input image, the model knows when it performs good or bad and can act accordingly. The goal here is to minimise the loss function such that the model minimises the amount of mistakes it makes. Minimising a function proves to be a complex task because of the complex non-linearities combined with many parameters.

Gradient descent

Before going to the complete process of minimising the loss function, let us start with the basics. Gradient descent is an iterative optimisation algorithm to locate the (global) minimum in a function. It does this by starting at a random point on the function and 'walking' down step-by-step towards a minimum. To determine the direction we want to walk in, the derivative needs to be calculated at each step and in each dimension. The combined derivatives taken in each dimension are referred to as the gradient and indicate the direction and intensity of the slope.

Assume the resulting loss of the input image X and weights w, $L(w|X)$. Through gradient descent, we want to converge towards a minimum in the loss function L. From the loss function L the partial derivative can be taken with respect to each possible tunable weight w_i in the network. This results in a gradient:

$$\nabla_w E = [\frac{\delta E}{\delta w_0}, \frac{\delta E}{\delta w_1}, \frac{\delta E}{\delta w_2}, ...]^T \tag{3.7}$$

Based on the gradient, the weights w_i are updated in such a way that the loss decreases. Applying this process iteratively, improves the model with respect to the loss function. Figure 3.6 illustrates gradient descent. The black line represents the path taken in order to reach a local minimum after multiple iterations.

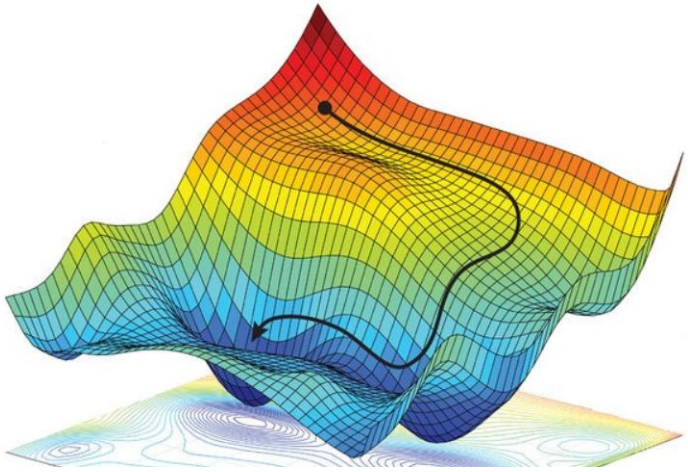

Figure 3.6: Gradient descent [25]

Gradient descent needs to used with care. Because the gradient is calculated in a local point in the function, it may not point towards the global minimum and can eventually get stuck in a local minimum. A solution to this could be to train the model multiple times starting from different randomly chosen points (corresponding with randomly initialised weights) on the curve and compare the results.

It is possible to adapt the step size or **learning rate** during gradient descent. The learning rate essentially determines how fast a model learns. If it is chosen too small, it may take a longer time to reach a minimum and can get stuck in a local minimum. If chosen too big, it can overshoot the minimum (figure 3.7). Both extremes result in a sub-optimal learning process. Choosing the optimal learning rate is of great importance during training.

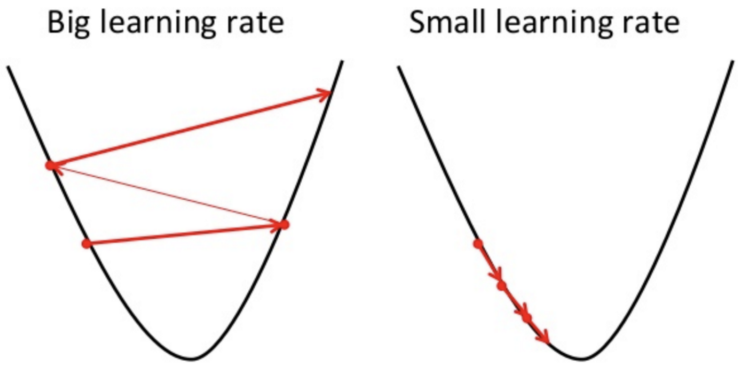

Figure 3.7: The impact of the learning rate [27]

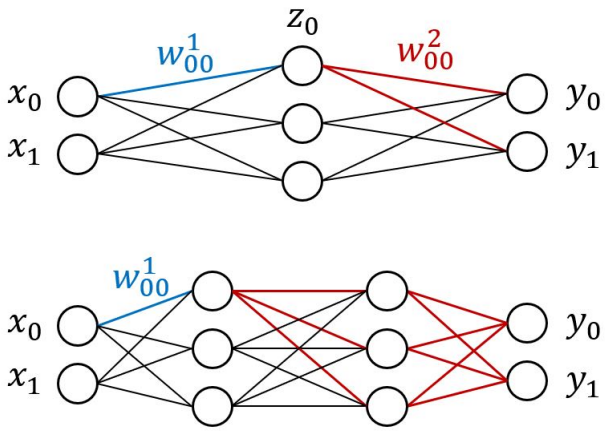

Figure 3.8: (top) Illustration of influenced branches when w_{00}^1 needs to be updated, (bottom) Multiple hidden layers increase the complexity exponentially

Backpropagation

Backpropagation uses gradient descent to update all the weights and biases in the network. Assume a network with 2 inputs and 2 outputs with only 1 hidden layer as given in figure 3.8. To finalise the construction of this example the loss function L is defined. Updating w_{00}^1 in such a way that L gets minimised, requires calculating the partial derivative $\frac{\delta L}{\delta w_{00}^1}$. The loss function however is only defined at the output layer y and thus w_{00}^1 can only be "reached" indirectly. The top network in figure 3.8 highlights the paths influenced by w_{00}^1. This is also reflected in the calculation of the gradient:

$$\frac{\delta L}{\delta w_{00}^1} = \frac{\delta L}{\delta y_0}\frac{\delta y_0}{\delta z_0}\frac{\delta z_0}{\delta w_{00}^1} + \frac{\delta L}{\delta y_1}\frac{\delta y_1}{\delta z_0}\frac{\delta z_0}{\delta w_{00}^1} \tag{3.8}$$

Adding more hidden layers increases the complexity exponentially since more paths can be followed for a given weight. This is illustrated in the bottom network of figure 3.8.

Backpropagation iteratively updates all the weights of the network starting from the output propagating to the input. In total the learning process consists of two components. First, the input gets passed through the network in the forward pass. This way the network can see examples and calculate the loss. Second, a backward pass is performed using backpropagation. The complete network gets updated and (hopefully) improved.

These ANNs can be very deep leading to an explosion of weights to be updated. In order to reach convergence, it is common to put the training data through the network multiple times. One such iteration is called an **epoch**. An epoch is not processed in its entirety but in **batches**.

Passing only one input sample per iteration causes way too many updates in the weights and can even degrade the performance of the model. This can happen when a sample is not correct or representative. Using batches to calculate the backwards pass gives a more averaged out representation of the direction (negative gradient of L on all training data) it has to go.

3.3.2 Convolutional neural network

A Convolutional neural network (CNN) is a special type of ANN used on any data that can be structured in a grid, e.g. image data. Algorithms that handle image processing typically work with specific filters or kernels that are used e.g. to detect edges. However, these kernels need to be manually defined by an engineer. A CNN attempts to learn and develop its own filters deciding on itself which features in the image are the most important when tackling a certain classification problem.

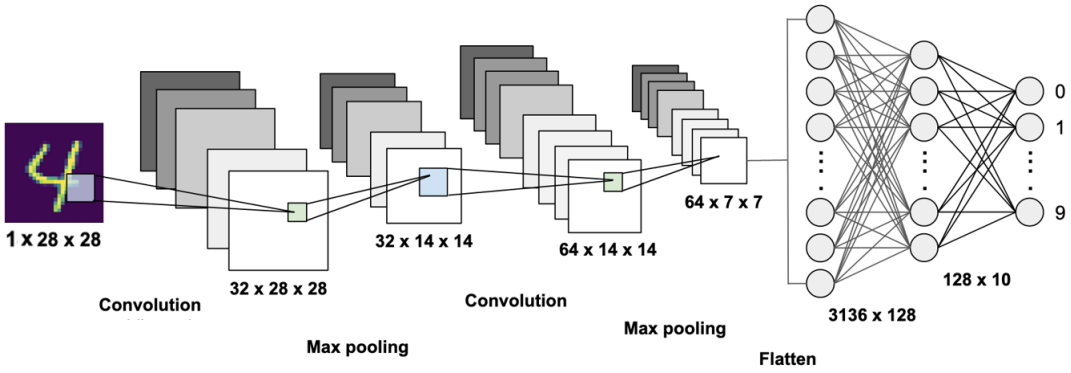

Figure 3.9: The architecture of a Convolutional neural network

Figure 3.9 illustrates the architecture of a CNN. The task at hand is again determining which number is on the image. The CNN typically consists out of 3 main components: the convolution layers, the pooling layers and a dense ANN.

Convolution layers

The convolution layers are the core building blocks of the CNN. Images have the property of being ”stationary” meaning that features learned in one part of the image can be reused in another part. For example an image taken from a city usually contains multiple horizontal edges. This suggests that a kernel of 3×3 features applied on a small part of the image can also be applied to another part resulting in the same activations for the same patterns.

In figure 3.10, a 3×3 matrix kernel is convolved with an image. This means that the kernel

gets shifted over the complete image and at each point calculates the dot product between the kernel and a part of the image. The output is a new matrix that represents the activations of the kernel at each position in the image.

The example in figure 3.9 takes as input a grey-scale image of the number 4. In total 32 different kernels are convolved over this image resulting in 32 so-called feature channels. The kernels are trained using backpropagation until they represent a specific feature that they have to detect in the image.

Note that the amount of features explodes exponentially since each kernel results in a new ”image”. This is the reason that convolution layers are not connected directly with each other but separated with pooling layers.

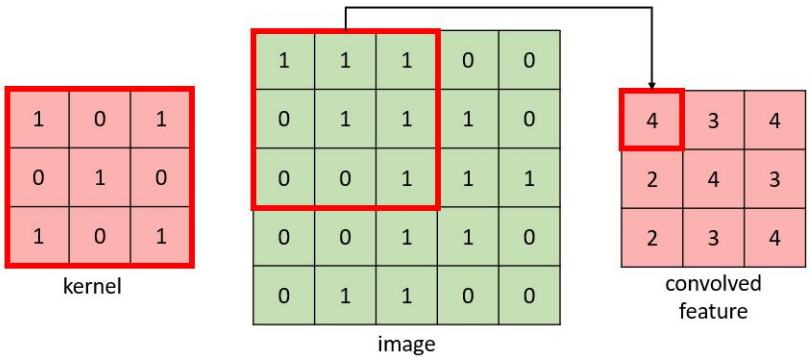

Figure 3.10: The kernel is convolved with an image resulting in a matrix of output activations

Pooling layers

Pooling layers are introduced to reduce the amount of features that are created and to decorrelate the feature channels. They do not only prevent the explosion of features but also reduce the chance of overfitting the model.

The pooling layers work by aggregating the activations of the convolution layer. This aggregation can be seen as a summary of the features and is performed with an operation referred to as **pooling**. Different types of pooling operations exist but the most used are **max pooling**, which takes the maximum of the presented feature area and **average pooling** which takes the average of the presented feature area.

The example in figure 3.9 uses a non-overlapping 2×2 kernel which effectively reduces the dimension of the feature channels by a factor 2. The total amount of features is reduced with a factor 4 in this case. An example of a detailed operation is given in figure 3.11.

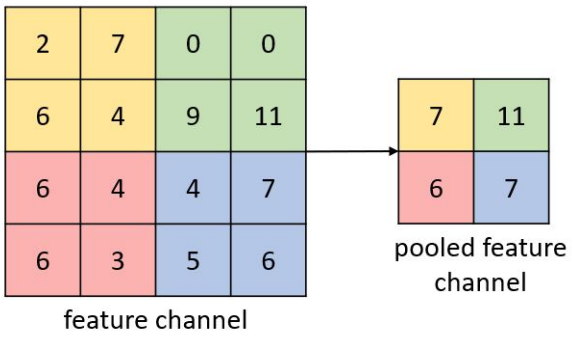

Figure 3.11: Max pooling is applied on the activation matrix.

Dense ANN

In the final part of the CNN a classic ANN is used to make the final classification. To this end, the feature channels at that point need to be flattened so that they can be used as raw input into the ANN. The multiple convolution and pooling layers can be seen as feature extractors specifically designed for images. The advantage is that the feature extractors or kernels do not need to be manually designed but can be learned using the theory presented in the section about neural networks.

3.3.3 Residual neural network

A RESnet can be seen as an extension of the traditional CNNs [28]. It has been observed that the usage of deep CNNs leads to lower performance and higher loss scores. However, deeper CNNs are required to solve increasingly complex systems. The reason traditional deep CNNs are so difficult to train is by a concept referred to as the vanishing gradients.

The vanishing gradient is a direct result from the backpropagation process. Recall that weights are updated using the gradient with respect to the loss function. The weights in the first layers of the model are updated using calculations that require multiple sequential derivatives. Assuming the model contains multiple layers, those derivatives tend to be extremely small when using the sigmoid function causing the initial layers to be updated with very small steps. These small gradients lead to a degradation of the model performance and limits the use of the traditional CNN.

A RESnet uses two solutions to optimise the training procedure. The first solution alleviates the vanishing gradient problem by using the ReLu function as activation function instead of the sigmoid function. The second solution is based on the observation the residual is easier to model by introducing skip-connections. As the name suggests, a skip-connection enables inputs

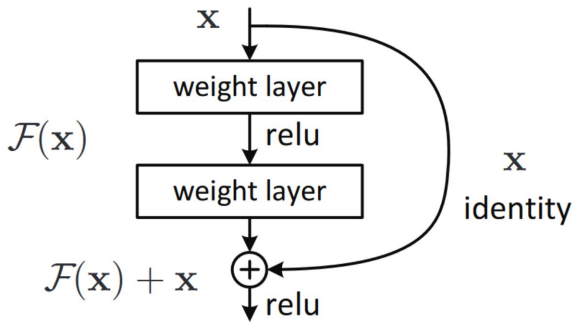

Figure 3.12: The building block of a Residual neural network

to skip several layers and provide a short cut to take during backpropagation. Figure 3.12 illustrates the basic building block of a RESnet. The information contained in the input vector x is not only transformed in the weight layer which is the same process taken in a CNN, but also directly combined with the output of the vector several layers later. Weights in the initial layers in the model will receive updates with higher values due to derivatives calculated via the skip-connections. A RESnet does not provide a complete solution against the vanishing gradient but is proved to be a more robust approach [28].

In figure 3.13, a RESnet is compared with two classic CNN models and illustrates the frequent use of skip-connections in the network.

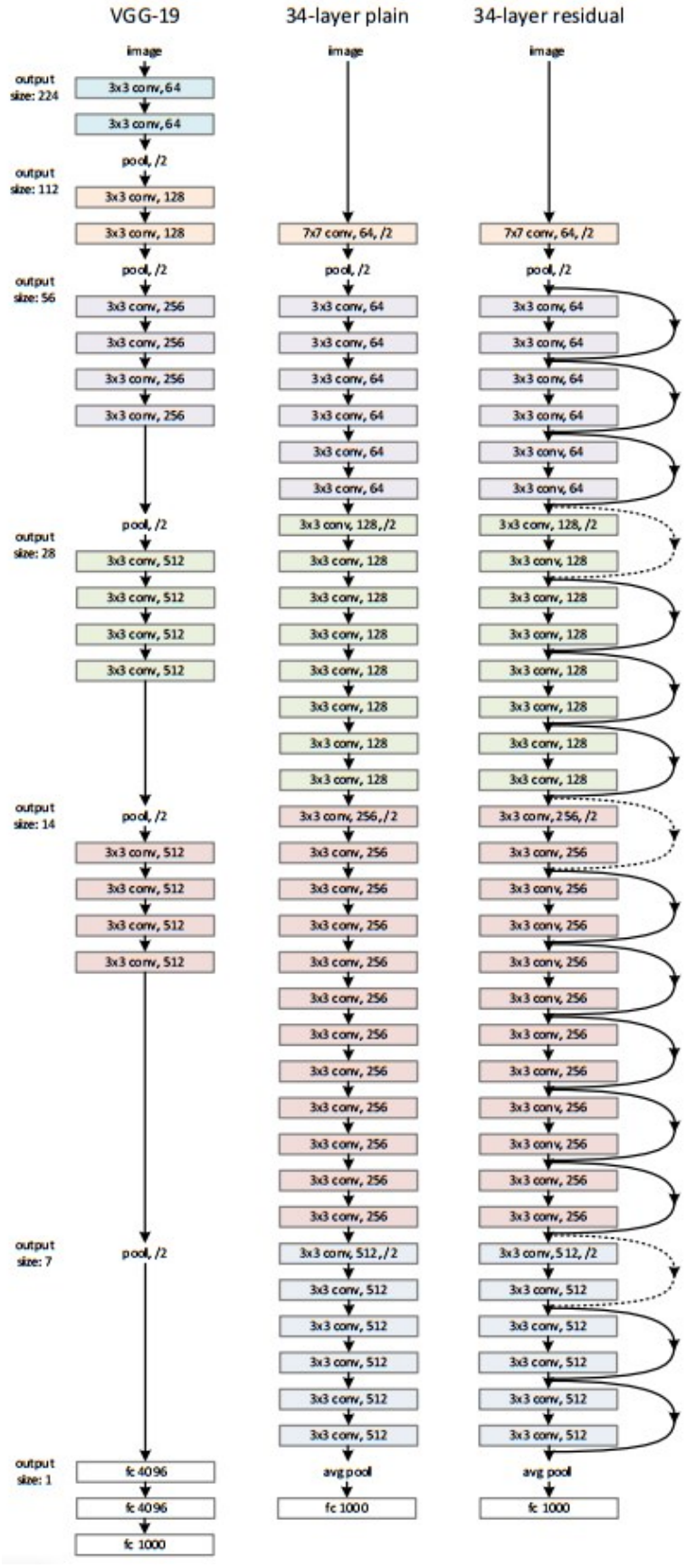

Figure 3.13: (left, middle) Classic CNN models, (right) RESnet34 model

3.3.4 Unet

One prominent part of image analysis is image segmentation. It involves dividing a visual input into different segments. In terms of supervised learning, segmentation is a special case of classification. Instead of classifying a complete image, segmentation assigns a class to each individual pixel, creating regions of similar pixels. Figure 3.14 gives an example of the segmentation approach to detect cells. In this image each pixel belongs either to a cell or to the background, effectively segmenting the images into small regions.

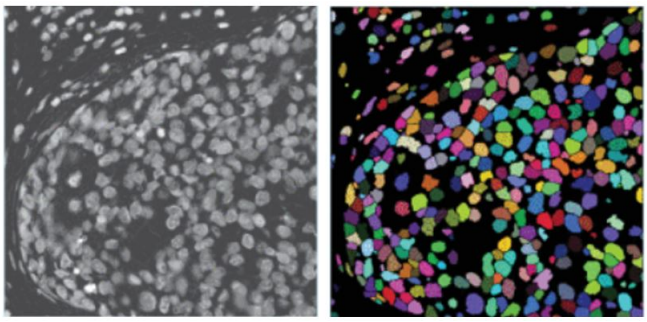

Figure 3.14: Segmentation example [29] (left) input image (right) output image

Segmentation can be achieved using multiple approaches but in deep learning, a Unet is often used [30]. This model is an adaptation of a RESnet and makes also use of skip-connections. A Unet consists of multiple levels with an encoder part and a decoder part (figure 3.15). The first level transforms the input image using multiple convolution layers before compressing the result and passing it to the next layer (red arrow). The grey arrow on each layer represents a skip-connection which are used with the same motivation as in RESnets, to model the residual. The following layers do exactly the same but on a decreasing number of features. This is the encoder step and effectively compresses the image in as few features as possible. The decoder process starts from the lowest layer and decompresses that image using inverse convolution layers (green arrow). In order to implement the skip-connections, each intermediate result is stored during the compression step and later on combined with the decompressed images in the same layer (grey arrows). Passing intermediate results to later stages in the network enables to use detailed information at each layer that otherwise would have been lost during compression.

This proves to be a powerful approach for segmenting images if detailed segmentation maps are available since this is a fully supervised method.

3.3.5 Evaluation

Evaluating a model is very important during development. It not only gives insight in the overall performance but can also be used to create a more nuanced picture in the mistakes the model

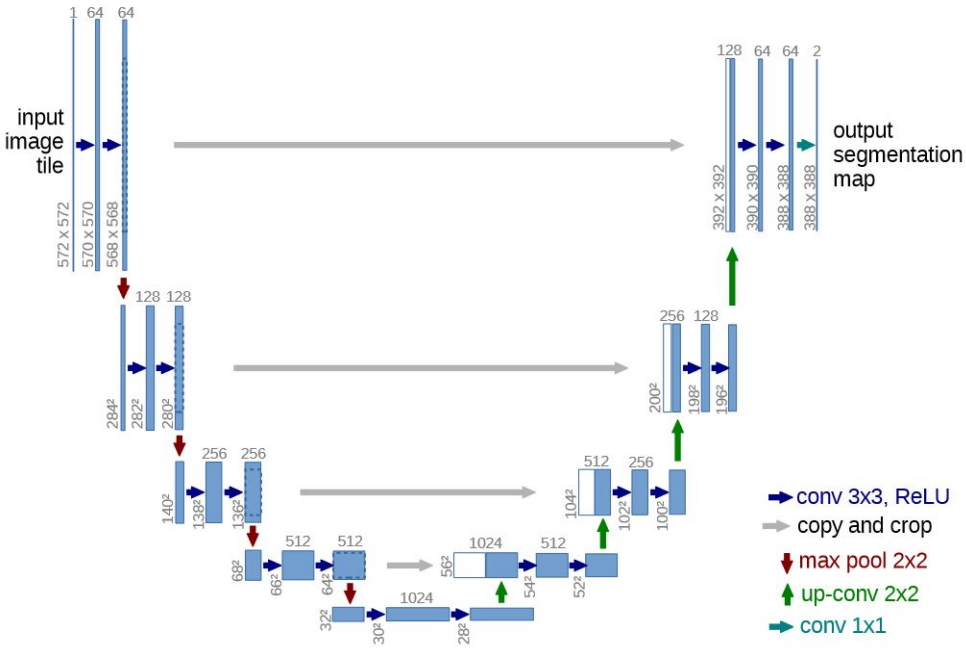

Figure 3.15: Unet

makes. If not done properly, this can result in sub-optimal models. We already mentioned in section 3.2 that a dataset needs to be divided into three non-overlapping subsets: the training dataset, the validation dataset and the test set. Setting the test set aside for now, the training and validation set in relation with each other provide a great deal of information.

We mentioned that data is passed through a model multiple times during training (epoch). By calculating the loss score for the training data and validation data after each epoch separately, a **learning curve** can be constructed (figure 3.16). The training loss decreases continuously over multiple epochs, indicating that the model is indeed improving. It can also be observed that the validation loss is higher than the training loss. This is a consequence of the fact that the model only updates its weights based on the training set while the validation set serves as unseen data. At a certain point the validation loss starts to increase. This phenomenon is called **overfitting**. When a model has been learning for a relatively long time, it will start to recognise all the samples in the training data. It starts to produces outputs not based on the features that we want to learn but because it recognises the image after so many iterations. Consequently, the training loss further decreases but the performance will degrade on unseen data causing the validation loss to increase. If a model is trained infinitely, the training loss will eventually drop to zero. As soon as the model starts overfitting, training can be stopped since it is not learning anything useful anymore and it will only hurt the overall performance of the model. Sometimes it happens that a model starts overfitting immediately. This can have multiple causes:

- The model is too complex for this task (too many weights to train).
- The training set is too small (too many weights with respect to the training data).
- The training data contains too much unimportant variability (e.g. label noise, too many outliers ...).

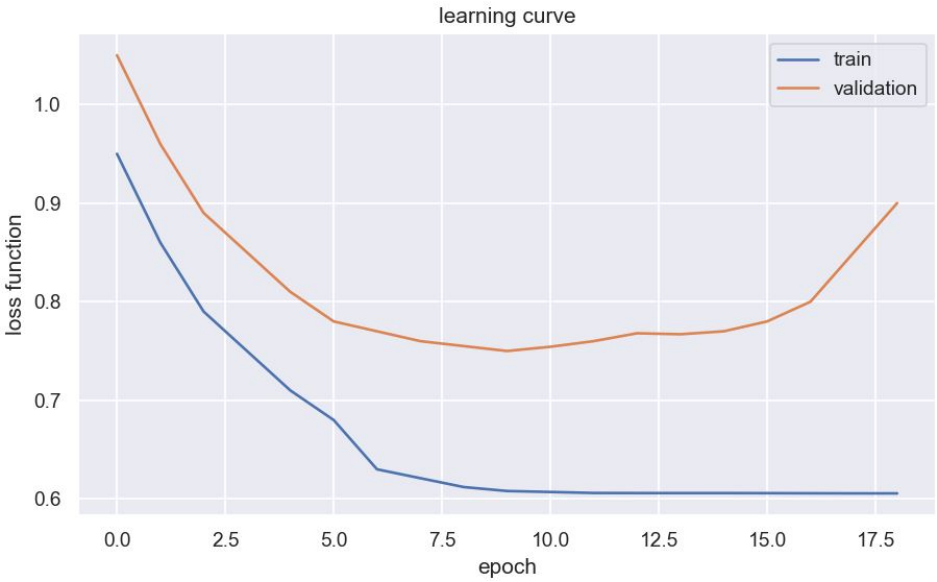

Figure 3.16: An example of a learning curve

On the other hand, **underfitting** can occur as well. When the training error is big, the model is not able to capture the features needed to make its classification. This can be caused by:

- The model is too simple for this task (not enough weights to extract useful information)
- The input does not contain enough informative information.
- The training dataset does not represent the distribution of the 'outside world'.

Overfitting and underfitting are related to the concepts of **bias** and **variance**. A model that is trained with the same hyperparameters but using slightly different datasets will behave differently. Indeed, gradient descent is an approximation technique and will not always find the global optimum causing different behaviour. The model bias indicates the capability of the chosen estimator to approximate the ground truth. Variance on the other hands indicates the sensitivity of the different estimators. In the bull's eye plots of figure 3.17, each blue dot represents the error of a model that is trained on a different set of data, measured on unseen data. The red dot represents the perfect model (able to make always a perfect classification). Overfitting on a different subset each time will lead to different validation scores depending on the training data

the model has seen. This leads to a high variance. Not being able to determine the relevant features leads to underfitting and results in a high bias. Note that underfitting and overfitting can occur at the same time.

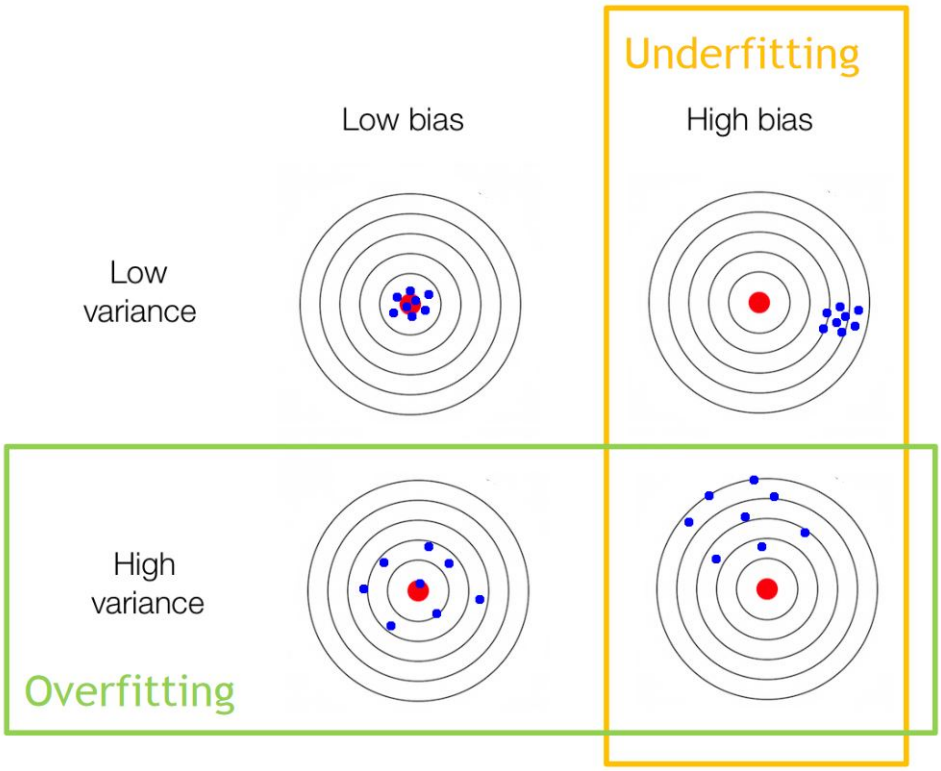

Figure 3.17: Bull's eye plots illustrating bias and variance [25]

For classification problems, evaluation metrics such as Area Under the Curve (AUC) and Receiver Operator Characteristics (ROC) curves are frequently used. The ROC curve is a probability curve and the AUC represents the degree of separability. They give an indication on how much a model is capable of distinguishing between classes. Higher the AUC, the better the model is at the classification task.

Assume now that we have a binary classification problem and our model outputs the probability of the input data belonging the class 1. One could say that when the model is at least 50% certain towards class 1, we classify it as class 1, class 0 otherwise. This would be the most obvious choice but in practice other thresholds can be chosen. By varying these thresholds, the ROC curve is constructed. The curve is created by putting the true positive rate (TPR) against the false positive rate (FPR).

$$TPR = \frac{TP}{TP + FN} \tag{3.9}$$

$$FPR = \frac{FP}{TN + FP} \tag{3.10}$$

With TP True Positives, TN True Negatives, FN False Negatives and FP False Positives. An example of ROC curves is given in figure 3.18. Starting with a threshold at 100%, nothing is being classified as class 1 ($TPR = 0$) but consequently, no FP occur as wel ($FPR = 0$). Lowering the threshold will increase the amount of TP (TPR increases) but will also start introducing FP (FPR increases). This process can be repeated until the threshold is at 0%, classifying everything towards class 1. In a well performing model, the TPR increases much faster compared to the FPR until a certain point (yellow plot in figure 3.18). On the other hand, a random classifier introduces an equal amount of TP and FP when changing the threshold (blue plot in figure 3.18). The AUC is the area under the roc curve and ranges between $[0.5, 1.0]$. Theoretically, an AUC lower then 0.5 is possible. In that case inverting the model output will map the AUC value between $[0.5, 1.0]$.

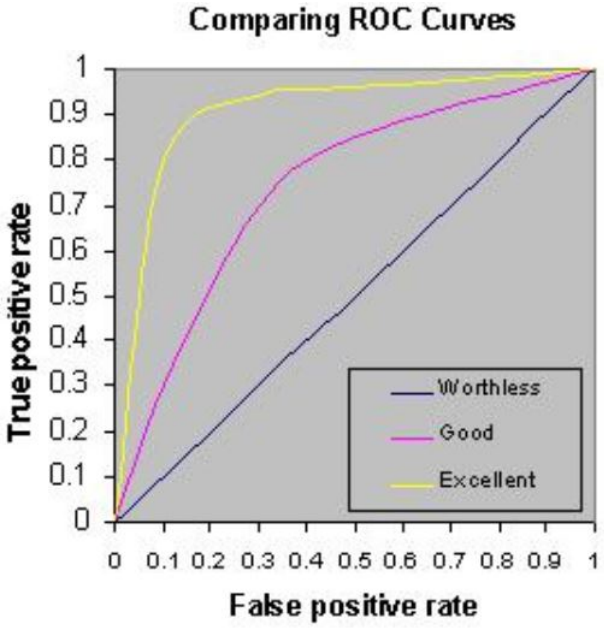

Figure 3.18: Comparing 3 ROC curves. The blue line corresponds with a random classifier [31]

3.4 Closing remarks

The goal of this chapter was to provide the reader with the necessary information to understand the later chapters. Special care was given to understanding supervised learning and deep learning

models. Of course, the field of deep learning, and by extension machine learning, is far more vast and diverse than given here.

4

Data

We want to predict the final outcome of SBRT based on microscopic cancer tissue. In this chapter we focus on the data needed for both segmentation and therapy response prediction. In the first section, colour staining is introduced. This is a technique to make sure the tissue is a viable product for further analysis. Because when the tissue is extracted from a patient, it appears colourless. Based on this foundation, we introduce the two datasets that will be used during development: the PANDA dataset for segmentation and the SBRT dataset for therapy response prediction.

4.1 H&E staining

One problem with tissue is that they are initially not suitable for use after extraction. They appear colourless and contain little useful information. **Staining** provides a way using chemical processes to improve the contrast, colour cells and tissue. This drastically increases the amount of useful information that can be extracted from such a microscopic image. Different types of staining techniques exist and they mainly differ by their chemical processes and elements that they highlight. The most used staining technique is **Haematoxylin and Eosin (H&E) staining** [32].

Haematoxylin is a dye extracted from the tree haematoxylum campechianum. By oxidising this product, hematein is created which is the actual dye used to colour tissue. It highlights the nuclear cells and their details in a purple/red-like colour. The depth of colouration not only relates to the amount of DNA present in the nuclei but also to the length of time the sample is exposed to haematoxylin. This creates variability in the final results which will be discussed later on in this chapter. **Eosin** is commonly used for counterstaining the tissue. It creates a clear distinction between the cytoplasm and the nuclei of the cells. Typically the resulting colour is pink with different shades of pink for different types of connective tissue fibres. Varying the doses and ratios of haematoxylin and eosin provides the ability to customise the desired results.

The actual colourisation process and protocol is far more extensive than explained here. We will leave this for the interested readers [32]. Figure 4.1 gives an example of successful staining on prostate tissue. Cells are the purple elements contrasted with the connective tissue which is dyed in pink.

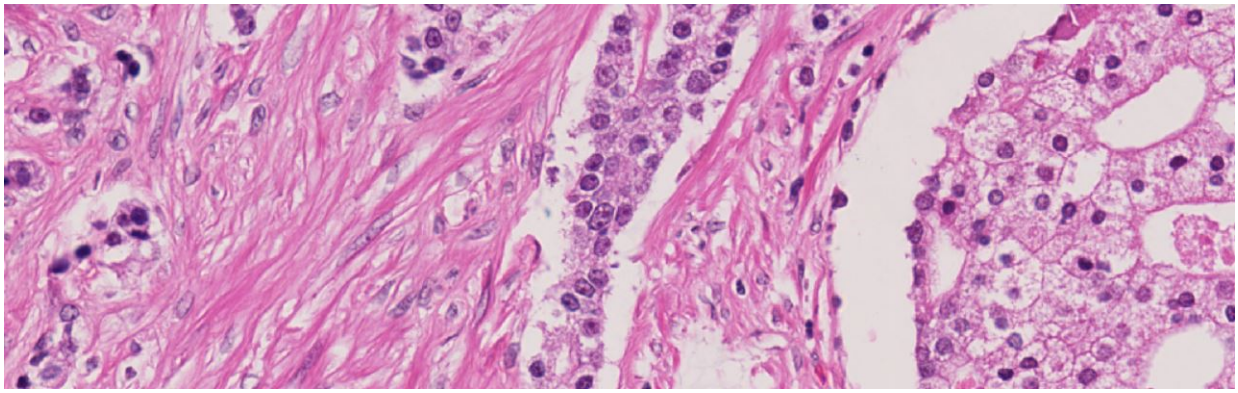

Figure 4.1: H&E staining example

4.2 Whole slide images

WSIs are digitised tissue sections that offer a microscopic view of the tissue [33]. Consequently, the resolution of these images is very high, resulting in a large file size. It is important to note that the camera will not scan the tissue at once. Instead it scans over the tissue, taking pictures of small individual tiles at high resolution. After the camera is finished, the tiles are stitched together in software in order the create a complete WSI.

The pictures are often taken on 20x or 40x magnification, which corresponds with a pixel resolution of 0.5 μm and 0.25 μm per pixel respectively [33].

For optimisation purposes, WSIs are stored in a pyramidal structure (figure 4.2). The structure contains different levels starting with the original high resolution image. This image is down-

Level	**dimensions**	**file size**	**ratio**	**percentage**
baseline	120000 × 80000	29 Gb	1:1	92.6%
1	30000 × 20000	1.8 Gb	4:1	6%
2	7500 × 5000	112 Mb	16:1	0.4%
3	1875 × 1250	7 Mb	64:1	-

Table 4.1: Storage usage for a typical WSI example

sampled with a factor 4 creating the second layer and so on. Using such a layered structure has the advantage that viewing in software can be done fluently. Only the part of the tissue that is zoomed in on, should display the high-res image. The higher layers are loaded when zoomed out, optimising the complete process. This is beneficial for image analysis since the layers and thus the resolution can be chosen freely without implementing the downsampling process manually.

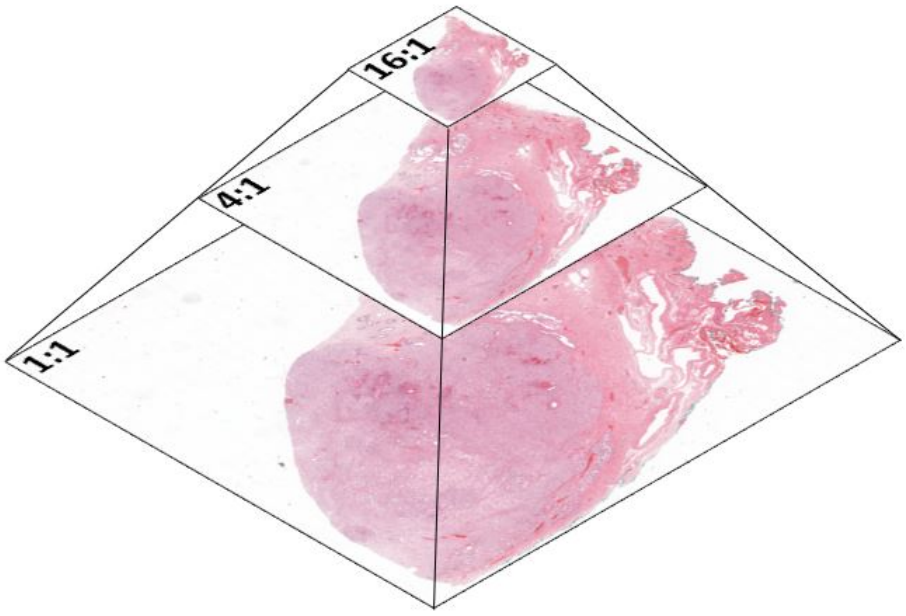

Figure 4.2: Pyramidal structure storage WSI

The image is stored multiple times, taking in more storage. Let us take a look at a typical example in table 4.1. The dimensions of this WSI are 120000 × 80000 (at 40x magnification) stored in RGB format corresponding to a file size of 29 Gb. Downsampling with a factor 4 reduces the area and the file size with a factor 16. Downsampling even further leads to file sizes in Mb. We can see that the file size reduces significantly with every layer. Consequently, the downsampled layers do not take in much storage relatively compared with the original high resolution image. The advantage of using this pyramidal structure outweighs by far the extra storage it requires.

4.3 PANDA dataset

The PANDA dataset is a public dataset originating from Kaggle [34], a crowd-sourced platform to solve data science, machine learning and predictive analytics problems. This dataset will be used for training models to automatically segment tumors from a WSI. The dataset contains 10616 prostate biopsy samples stored as WSIs each originating from a unique patient. Each WSI is given a Gleason score as well as a mask highlighting the relevant areas on the tissue. The dataset contains slides from two different institutions: Karolinska institute and Radboud university medical centre. Karolinska provided 5456 WSIs and Radboud 5160 WSIs. Figure 4.3 showcases the Gleason grade distribution while comparing the two institutions. "$0 + 0$" are biopsies that do not contain any tumor. From the figure, there is a clear class imbalance present in terms of Gleason score in both institutions.

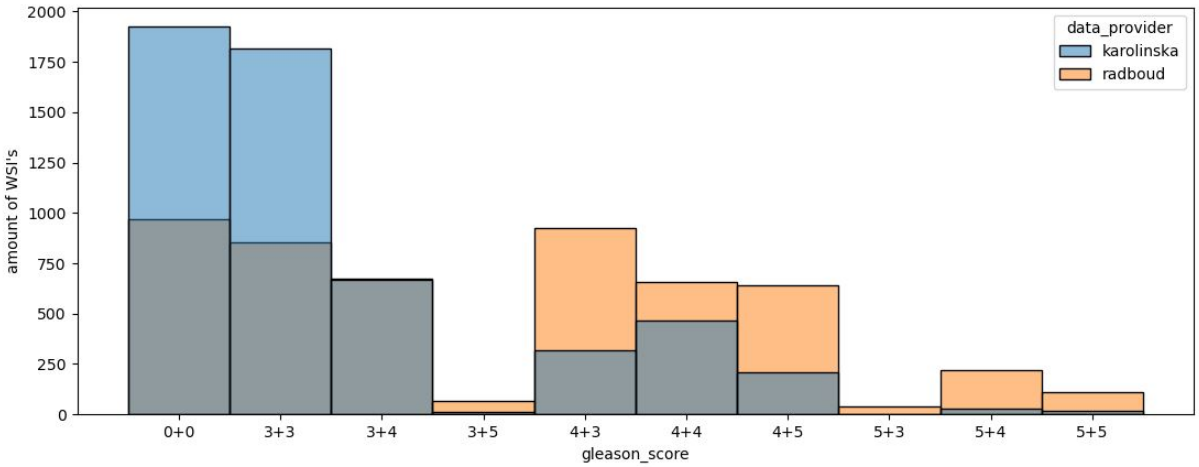

Figure 4.3: Gleason grade distribution of the PANDA dataset

The WSI's from both institution were created using different processes, resulting in some differences. We provide a short overview.

Radboud

The slides are taken at 20x magnification (or 0.5 μm/pixel) with an average dimension of around 20000×20000 pixels. As mentioned before the slides do not only contain metadata with the Gleason score, but also a mask. This mask assigns to each pixel an integer value between 0 and 5, which represents a unique class within the image:

- **label 0**: background (non tissue) or unknown
- **label 1**: stroma (connective tissue, non-epithelium tissue)

- **label 2**: healthy epithelium
- **label 3**: cancerous epithelium (Gleason score 3)
- **label 4**: cancerous epithelium (Gleason score 4)
- **label 5**: cancerous epithelium (Gleason score 5)

Epithelial cells make up the glandular portion of the prostate and stromal cells make up the connective tissues. Figure 4.4 illustrates the binary mask with its corresponding WSI. We can see that stroma takes in the most space but also overestimates the amount in some places incorporating background. The epithelial cells are grouped and classified with high precision. In this image only Gleason score 4 and 5 cancer cell clusters are present but Gleason score 3 can also be present at the same time.

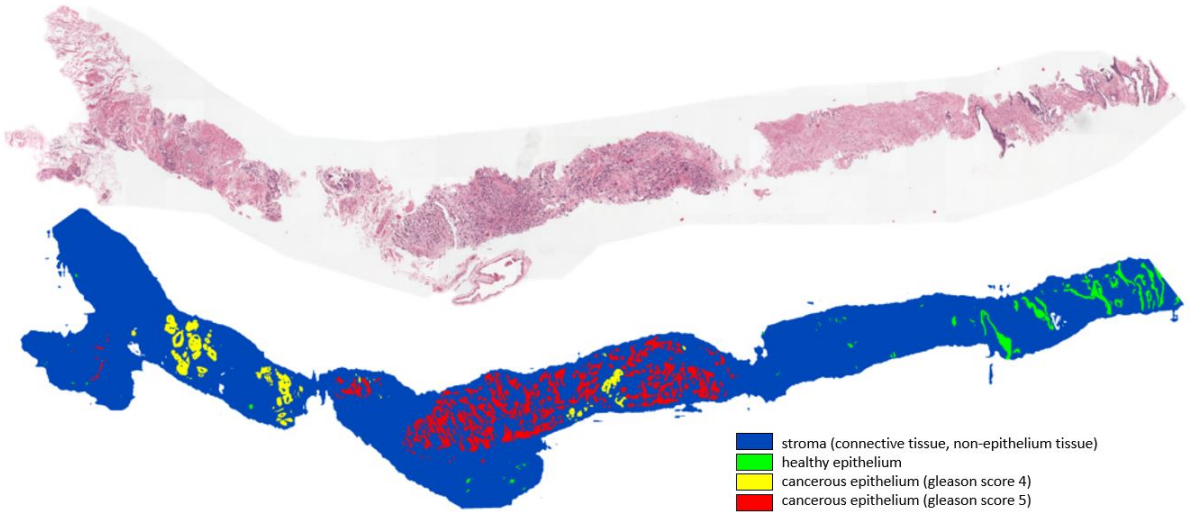

Figure 4.4: WSI provided by the Radboud institute

Karolinska

These slides were taken at 20x magnification as well but are slightly larger with dimensions of approximately 30000×30000 pixels. The masks are however less detailed with only three labels:

- **label 0**: background (non tissue) or unknown
- **label 1**: healthy tissue (stroma and epithelium combined)
- **label 2**: cancerous tissue (stroma and epithelium combined)

Stroma and epithelial cells are combined and clustered in cancer or non-cancer tissue. This dataset also does not provide a detailed Gleason score location as can be found in the slides provided by Radboud. An example is shown in figure 4.5. The dataset contains frequent artefacts which can also be seen on this image: a significant amount of tissue has no annotation and straight lines of healthy tissue can be found between cancerous tissue.

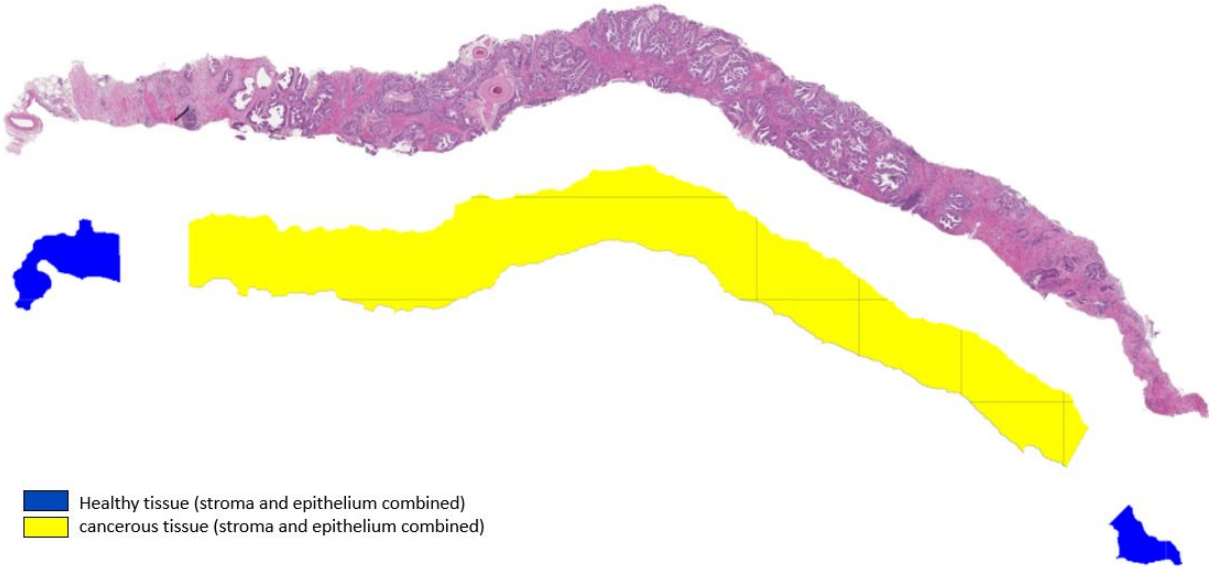

Figure 4.5: WSI provided by the Karolinska institute

4.4 SBRT dataset

The SBRT dataset is the result of a collaboration between Johns Hopkins and Ghent University. Both institutions organised trials [35]: patients were selected that have metachronous oligorecurrent prostate cancer. The term 'oligorecurrent' means that the patient develops a certain number of metastasised tumors after receiving a successful treatment and after a certain treatment-free time period. The selected patients in this trial were divided into two groups or **arms**. One arm received SBRT while the other arm received no radiotherapy and only went through observation. The patients in both arms received after some period of time androgen deprivation therapy (ADT) which is a hormone therapy.

Before the patients received any treatment, biopsies of the prostate were taken and WSIs were created containing part of the tumor. The WSIs together with the metadata containing trail results, make up the SBRT dataset. The metadata consists out of multiple values indicating the effect of the treatments on each individual patient in each arm. One of the endpoints of the study that will mainly be used in this work is PSA failure. This is a boolean value that is set to 1 if the PSA levels started to rise above 4.0 ng/mL and 0 if this not happens [36]. Recall

from section 2.2 that the rise of PSA levels indicates that the prostate tumor is still present. We assume SBRT treatment to be unsuccessful when PSA failure occurs and successful if not.

The total amount of patients monitored by both institutions amounts to 175 patients. From these patients, 139 received SBRT treatment. Not every patient, however, has a WSI available, further reducing the amount of valid patients to 72. Figure 4.6 gives the distribution of the Gleason grade with respect to the PSA end-point. The upper figure uses all the patients in the metadata while the bottom figure is the result after filtering the patients that are deemed to be valid for this work. The higher Gleason grades are underrepresented. This could result in a model that might underperform for WSIs with those features.

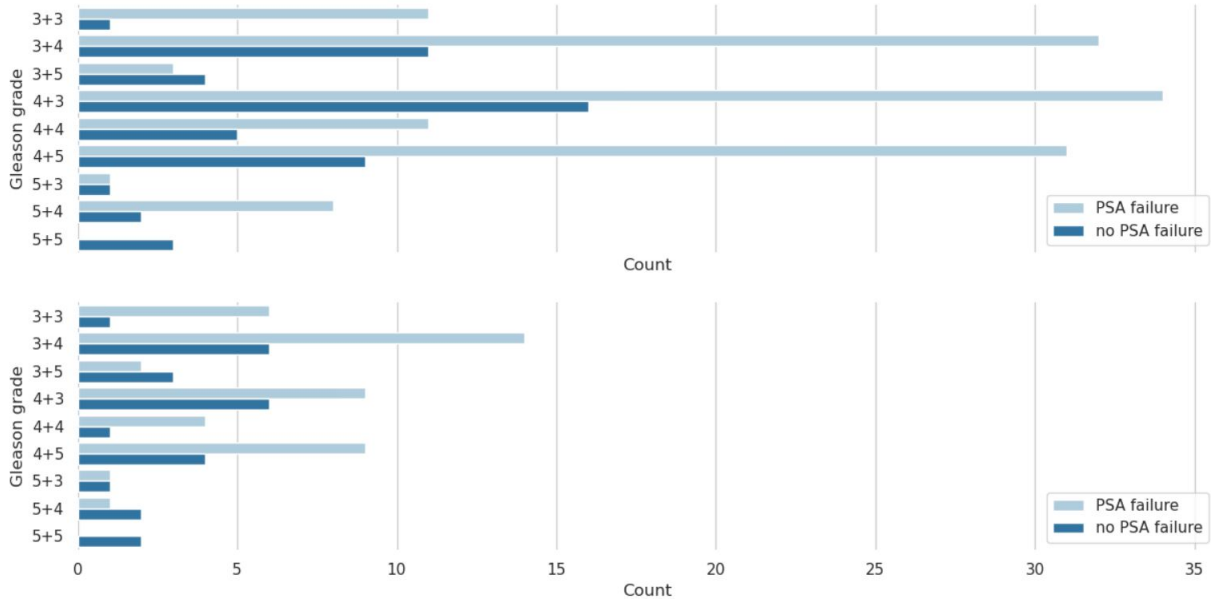

Figure 4.6: (top) Gleason grade distribution comparing the outcome of the treatment when all metadata is taken into account, (bottom) The distribution after the dataset is filtered, the filtering operation removes patients with no available WSIs or did not receive SBRT

4.5 Conclusion

The microscopic cancer tissue is stored in whole slide images, a data format specifically designed for digital pathology. To enable diagnosis based on WSIs, staining techniques are utilised. Both datasets used in this work make use of H&E staining. The PANDA dataset contains 10 000 WSIs, each with a respective high detail annotation mask. The data originates from 2 different sources: the Radboud and Karolinska institute. The SBRT dataset contains 175 patients and is the result of a collaboration between Johns Hopkins and Ghent University. It contains 2 trials which record end-points that indicate the effectiveness of SBRT.

5

Segmentation

The aim of this work is to determine if WSIs contain features that are predictive for response to SBRT treatment. Aside from tumor regions, WSIs can also contain a significant amount of healthy tissue. We assume that the cause can only be found in tissue consisting out of cancer cells and its immediate surroundings. Following this reasoning, the healthy tissue can function as noise in a deep learning model if it appears too much compared to the tumor region. In this chapter, we present our approach to automatically detect the tumor region to reduce the amount of noise for therapy response prediction (figure 5.1).

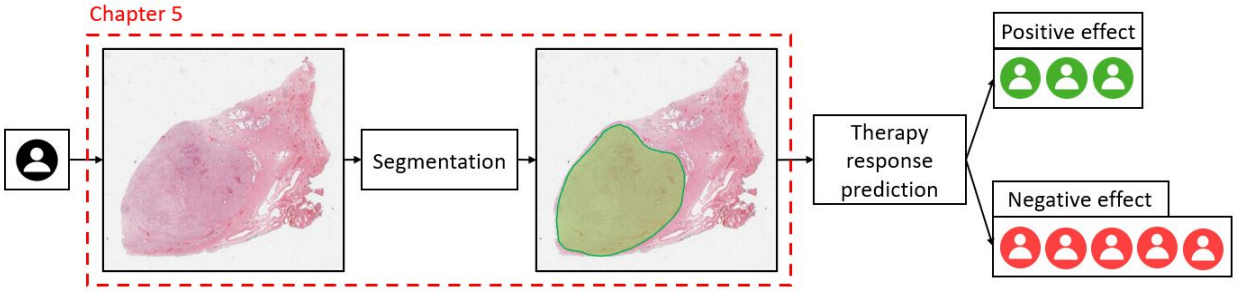

Figure 5.1: Situating chapter 5 in the overall pipeline

5.1 Segmentation

Before the rise of deep learning, segmentation on WSIs was mainly performed using handcrafted features based on the visual perception of cancer epithelium. Frequency analysis and filter banks proved to be an efficient approach for this task. Bianconi et al. [37] and Linder et al. [38] used features based on the coarseness, roughness and edges of the texture as input for a support vector machine (SVM). Using this method they were both able to successfully differentiate epithelium from stroma with 98% accuracy. However going one step further by making a distinction between cancerous epithelium and healthy epithelium cannot be easily done using texture features. Altunbay et al. [39] uses colour graphs within tiles of a WSI to determine if the tile in question contains no cancer, low-grade cancer or high-grade cancer. They were able to achieve 82.64% accuracy on their test set. High precision detection of cancer tumor is however a difficult task because of the handcrafted features and the extensive domain knowledge needed. For this reason we opt to use a deep learning approach in combination with the PANDA dataset. The organisers of the PANDA challenge Bulten et al. evaluated the top 15 models from the challenge both on Gleason grading and tumor detection [40]. On average the models for tumor detection achieved a sensitivity of 99.08% and specificity of 93.75% on a common test set. All of the top 15 approaches made use of deep learning models based on CNNs, Unets or RESnets.

We have seen that the PANDA dataset can be divided into two parts: data provided by Radboud and data provided by Karolinska. The WSIs from Radboud contain high detailed masks where the epithelial cell clusters are separated from the surrounding tissue. We decided to develop a model that was able to create high detailed masks. This is also beneficial for the therapy response prediction model. Figure 5.2 illustrates the difference between low detail and high detail segmentation. The annotation of the left figure over-represents the amount of tumor region (83%) on the tile while the high detail annotation only reach a coverage of 31%. If we want to extract tiles containing a high tumor coverage (the ROI), high detailed masks will be more representative.

To achieve high detailed segmentation, only data provided by Radboud will be used. The masks from the Karolinska dataset are not adequate because of the low detail in the annotations. From this point forward, we refer to the Radboud subset as the PANDAS dataset. Remember that the masks contain segments for each Gleason score. We decided to only make a distinction between healthy tissue (stroma and epithelium) and cancerous tissue (only epithelium) by converting the masks to binary masks with label 1 epithelial cancer cells and label 0 healthy tissue. Figure 5.3 illustrates the resulting binary mask after mapping all the classes corresponding with cancer (Gleason score 3, 4 and 5) to label 1 and the rest to label 0. Yellow corresponds with cancer cells and purple with background and healthy tissue.

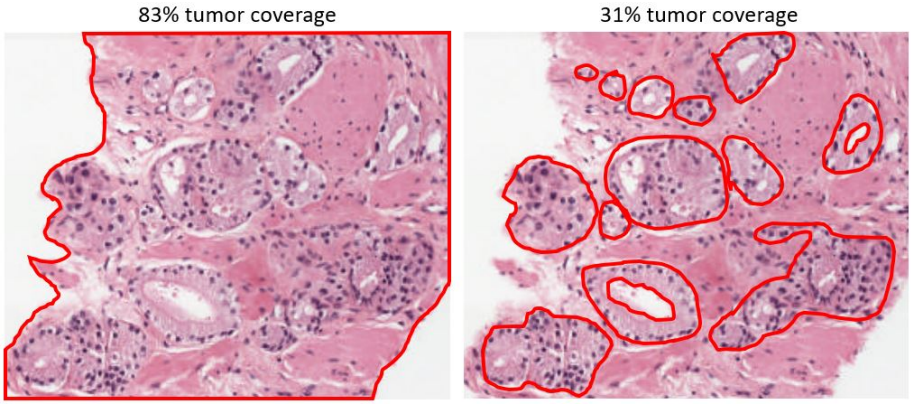

Figure 5.2: A sample of the PANDA dataset. (left) The low detail annotation corresponds with 83% coverage within the tile, (right) The high detail annotation corresponds with 31% coverage within the tile

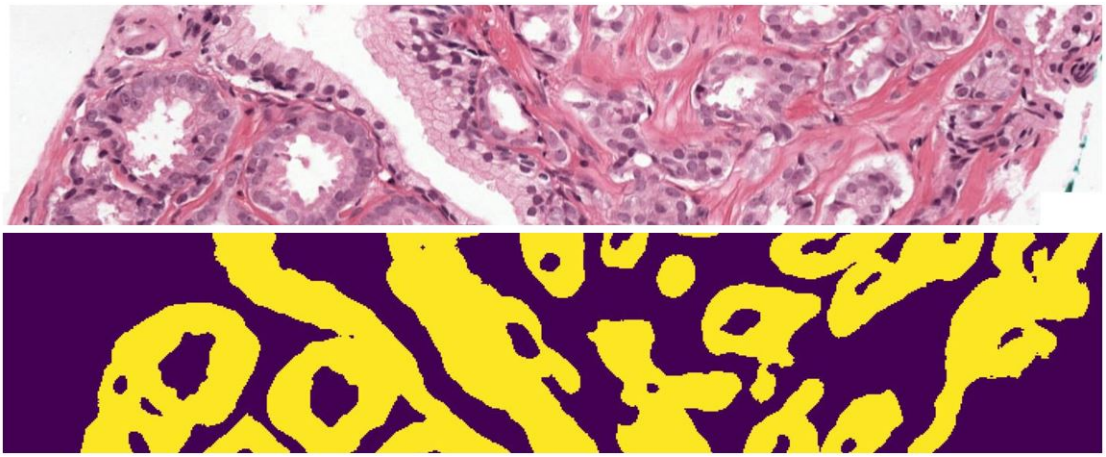

Figure 5.3: Example of the binary mask after conversion

Constructing the images and their corresponding mask in this way enables the use of state-of-the-art supervised deep learning techniques. The Unet model was chosen (see section 3.3.4) because of its powerful segmentation abilities. The main segmentation pipeline used is given in figure 5.4 and contains a pre-processing module and a post-processing module.

5.1.1 Challenges

Before discussing the relevant components for this task, we go over some interesting challenges to keep in mind when developing a model for segmentation. Identifying the potential complications are vital during development and evaluation of any kind of model.

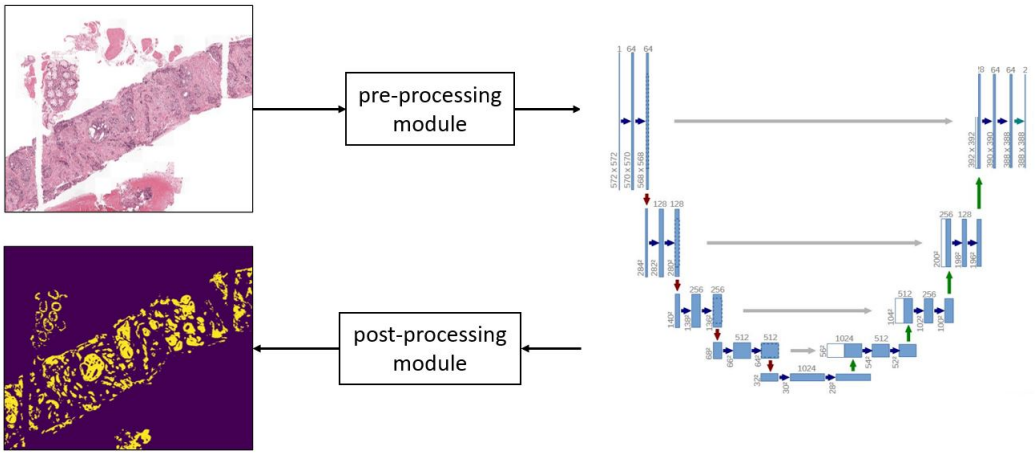

Figure 5.4: A high-level overview of the segmentation pipeline

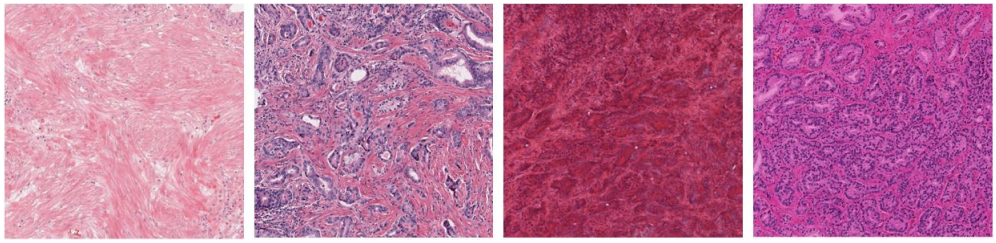

Figure 5.5: Stain variability

WSI variability

The hue, saturation and intensity of the slide after H&E staining suffer from a great deal of variability [41] (figure 5.5). This is not only due to inter-patient difference but is also caused by the different preparation methods. A staining that was given more time will appear significantly different than a staining process that is performed relatively fast. This is a problem for deep learning models which are very susceptible for changes in the input data. It is not rare to have a model that is capable of detecting tumors on one dataset but not on the other dataset. Variability is the main perpetrator of this phenomenon.

Under-representation benign epithelium

We observed a great deal of imbalance between healthy epithelium and cancer epithelium in the PANDA dataset. 76% of the annotated epithelial cell clusters is cancerous. This could be a problem for a model attempting to make a distinction between the two [42]. A possible consequence of this imbalance is a model that annotates all epithelial cells which is not a desired behaviour and will lead to an abundance of false positives.

WSI size

We mentioned earlier that WSI's have big dimensions. We have seen that these images can be multiple gigabytes containing millions of pixels. And although there are advantages of having such high resolution data, we simply do not have the computation power to apply a model on a complete WSI [43]. Even if it were possible one still has to deal with the variable sizes between the WSI's itself since most deep learning models expect a fixed input. Typically the way to deal with this problem is to divide the image in very small tiles of around 512×512 or 256×256 pixels which a model can handle [44, 45, 46]. Since using this method, each tile is considered to be independent from one another, this comes with the cost of losing spatial information [46]. For the purpose of segmentation however this problem is negligible and the tiling process can be performed without major consequences.

Gleason score

The Gleason score (see section 2.2.1) gives an indication about the aggressiveness of the tumor in the prostate. The severity corresponds with structural differences that can be observed in H&E stained tissue. For our application, the segmentation model should be able to cope with the different grades without the model knowing what the score actually is. Additionally, the Gleason grading system is partly subjective, due to soft boundaries between the different grades [47]. Finally, high Gleason grades are underrepresented in the PANDA dataset. This could potentially lead to a model performing well on low grade cancer but not on high grade cancer.

5.1.2 Pre-processing

The pre-processing module mentioned in figure 5.4 is the most important element of the pipeline to combat the challenges mentioned in the previous section. The module transforms raw-input WSI's into a format that the model can process. It is also an essential process for making the model robust. In figure 5.6, the complete pre-processing flow is given.

Tiling + background removal

Each WSI is divided into small tiles of 512×512 at 10x magnification (or 1.0 μm/pixel). Referring back to previous chapter, the average sized WSI in the PANDAS dataset is 20000×20000 pixels. Dividing it into tiles leads to a total amount of approximately 1525 tiles per patient. This results in a significant gain in data for each patient that gets added in the training set. A significant percentage of tiles is however background and are of absolutely no benefit in the learning process.

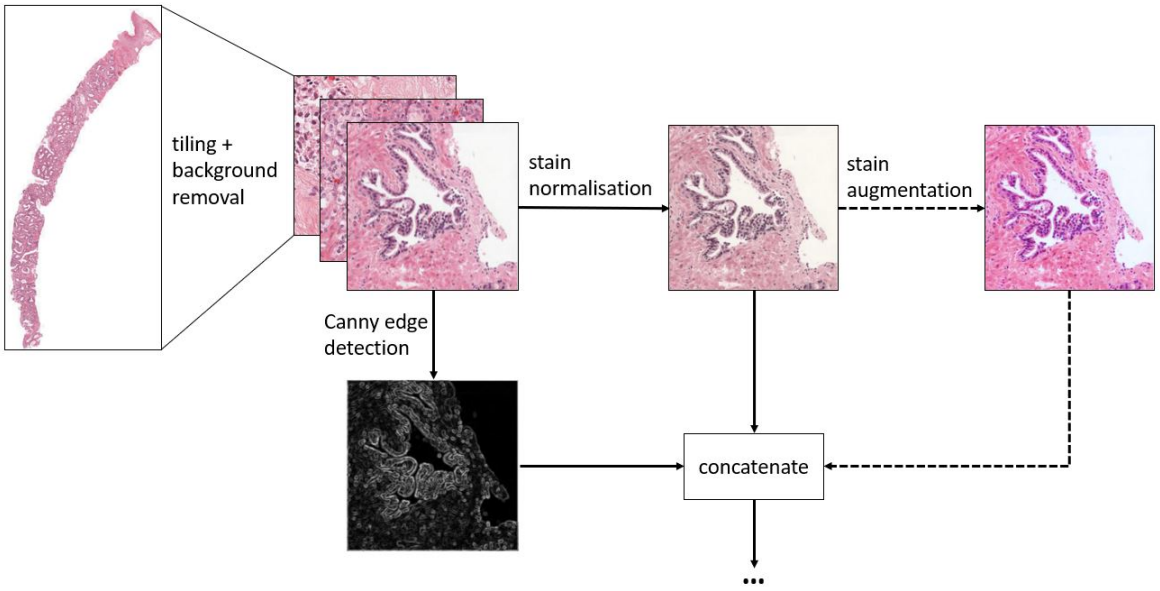

Figure 5.6: The preprocessing workflow

For this reason, we remove tiles that contain 50% or more white space.

Stain normalisation

To make the model more robust against the WSI variability, we apply a certain stain normalisation technique called **Reinhard stain normalisation**. This technique was developed in 2001 by Reinhard et al. [48] and makes use of the perceptual properties of the Lab colour space. The goal of Reinhard et al. is to transfer the general properties of a fixed target image to a given input image e.g. transform a photo taken in full daylight using a target image taken by night. This results in a daylight photo seemingly taken by night.

What makes the Lab colour space so useful is the fact it minimises the correlation between colour channels [49]. Transformations (such as normalisation) performed in this colour space reduce the amount of artefacts that are introduced as a consequence. The WSIs, however, are stored using the traditional RGB colour space. Transformations exist to convert the values in the RGB colour space to values in the Lab colour space. The transformation is done as follows [49]:

$$\begin{bmatrix} L \\ M \\ S \end{bmatrix} = \begin{bmatrix} 0.3811 & 0.5783 & 0.0402 \\ 0.1967 & 0.7244 & 0.0782 \\ 0.0241 & 0.1228 & 0.8444 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{5.1}$$

$$\begin{aligned} L' &= log(L) \\ M' &= log(M) \\ S' &= log(S) \end{aligned} \tag{5.2}$$

$$\begin{bmatrix} L \\ a \\ b \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{3}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{6}} & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -2 \\ 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} L' \\ M' \\ S' \end{bmatrix} \tag{5.3}$$

The L-channel represents an achromatic channel, while the a-channel corresponds with chromatic yellow-blue opponent channels and the b-channel with chromatic red-green opponent channels. Now we want to transfer the distribution of colour values in the Lab space from the target image to the input image. The mean $m_{input} = [\overline{L}_{input}, \overline{a}_{input}, \overline{b}_{input}]$ and standard deviations $s_{input} = [\sigma^L_{input}, \sigma^a_{input}, \sigma^b_{input}]$ are calculated of the input image. The same applies to the target image with $m_{target} = [\overline{L}_{target}, \overline{a}_{target}, \overline{b}_{target}]$ and $s_{target} = [\sigma^L_{target}, \sigma^a_{target}, \sigma^b_{target}]$. The normalisation is then performed using following equations:

$$\begin{aligned} L_{norm} &= (L - \overline{L}_{input})\frac{\sigma^L_{target}}{\sigma^L_{input}} + \overline{L}_{target} \\ a_{norm} &= (a - \overline{a}_{input})\frac{\sigma^a_{target}}{\sigma^a_{input}} + \overline{a}_{target} \\ b_{norm} &= (b - \overline{b}_{input})\frac{\sigma^b_{target}}{\sigma^b_{input}} + \overline{b}_{target} \end{aligned} \tag{5.4}$$

In the final step of this normalisation process, the resulting Lab values are converted back to RGB values.

To alleviate the disadvantage of potentially choosing a target image that is not representative, we instead calculated the mean and standard deviation over all the tiles in our training dataset. These values are stored for further use during validation and testing of the model (to avoid leakage). Figure 5.7 showcases the result of the normalisation operation using the complete training set as 'target image'. Observe that the WSI variability is visibly reduced using this method.

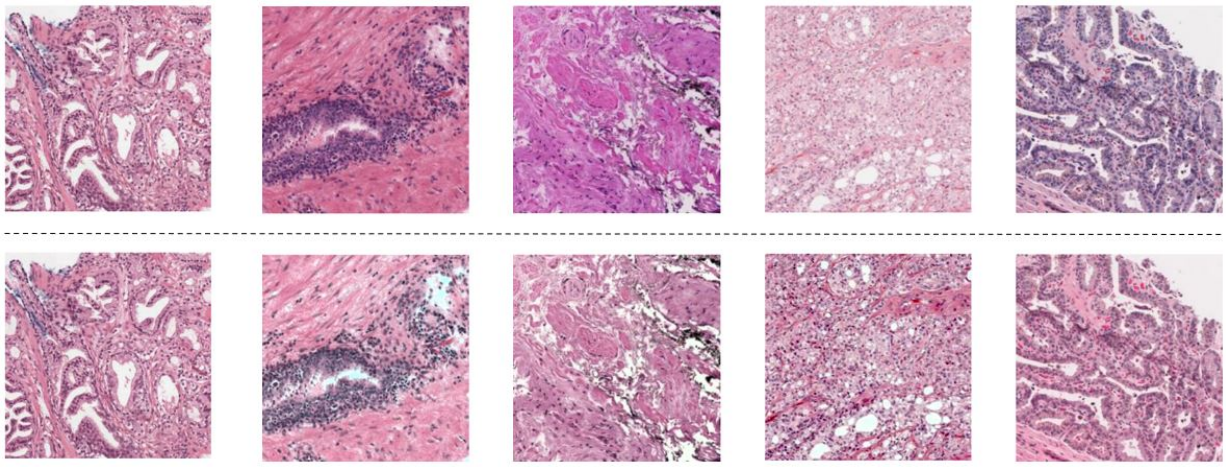

Figure 5.7: (top row) The original input images, (bottom row) The corresponding normalised images

Stain augmentation

Stain augmentation has been proposed as a method to increase generalisation performance by simulating realistic variations of the training data [50]. This step is only performed during training and is omitted during testing of the model. Stain augmentation randomly changes the brightness, contrast, saturation and hue of an image using small steps, creating a slightly different image. Introducing variability after normalising the tiles might seem odd, but it was done with focus on generality. We normalise the images because the WSI variability was simply too big for our model to cope with it (we will see this in a future chapter). However, in special cases the normalisation process is not entirely successful and some outliers (albeit close to the normalised cluster) can still be created. By stain augmenting the dataset, the model is able to cope with these special cases.

Canny edge detection

Apart from the 3 channels that correspond with an RGB image, we also create a 4th channel. This channel is the result of partially applying the multi-stage Canny edge detection algorithm on the original input image [51]. It consists out of 4 stages. First, a noise reduction technique is applied using a 5×5 **Gaussian filter**. This is needed because the algorithm is highly susceptible to noise. In the next step, the gradient of the image is calculated. Edges correspond with a sudden difference in pixel value and can be seen as a slope in a function. Consequently, the gradient has a high absolute value at each edge. Depending on how hard the edge is, the absolute value will be bigger or smaller. Because the actual function is not known and discreet (RGB), the derivatives are calculated using the **Sobel kernel**. The Sobel kernel is a 3×3 filter

and is a applied in two directions: horizontal K_x and vertical K_y. The corresponding filters are:

$$K_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$
$$K_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & 1 \end{bmatrix} \tag{5.5}$$

Convolving the image with the two kernels separately, approximates the gradient in the two directions: the derivative in the horizontal direction G_x and the derivative in the vertical direction G_y. From these two derivatives, the magnitude of the gradient $|G|$ and the angle θ of the gradient can be found:

$$|G| = \sqrt{G_x^2 + G_y^2}$$
$$\theta(x, y) = arctan\frac{G_y}{G_x} \tag{5.6}$$

$|G|$ is a 2D-image were the pixel values represent the absolute intensity of the gradient. It highlights the edges in the image. Additionally, θ represents the direction of the gradient at each pixel. The final steps in the algorithm are **non-maximum suppression** which finetunes the intensities based on the angle and **thresholding** to select the pixels certain to be contributing towards an edge. We do not apply the final two steps and instead use gradient intensities $|G|$ as the 4th channel. Examples of this channel can be found in figure 5.8 in the right column.

Note that the second row contains a blurring artefact that occurs frequently enough in other tiles to interfere with the overall model performance. The edge detection algorithm is not able to detect any edges in blurred area, giving the model a way to automatically detect areas that are of no interest. One could argue that the canny edge detection algorithm could be used beforehand to remove tiles that contain blurred artefacts. While this is a valid approach, we chose to provide the model with enough information to detect those areas on its own.

5.1.3 Post-processing

The Unet outputs two channels, corresponding with a specific class: the first class being background and healthy epithelium and the second class cancerous epithelium. If the model is certain about a class in a pixel, that pixel will receive a high value in the corresponding output channel and a low value in the other. In figure 5.9, we give an example output with the two output channels separated.

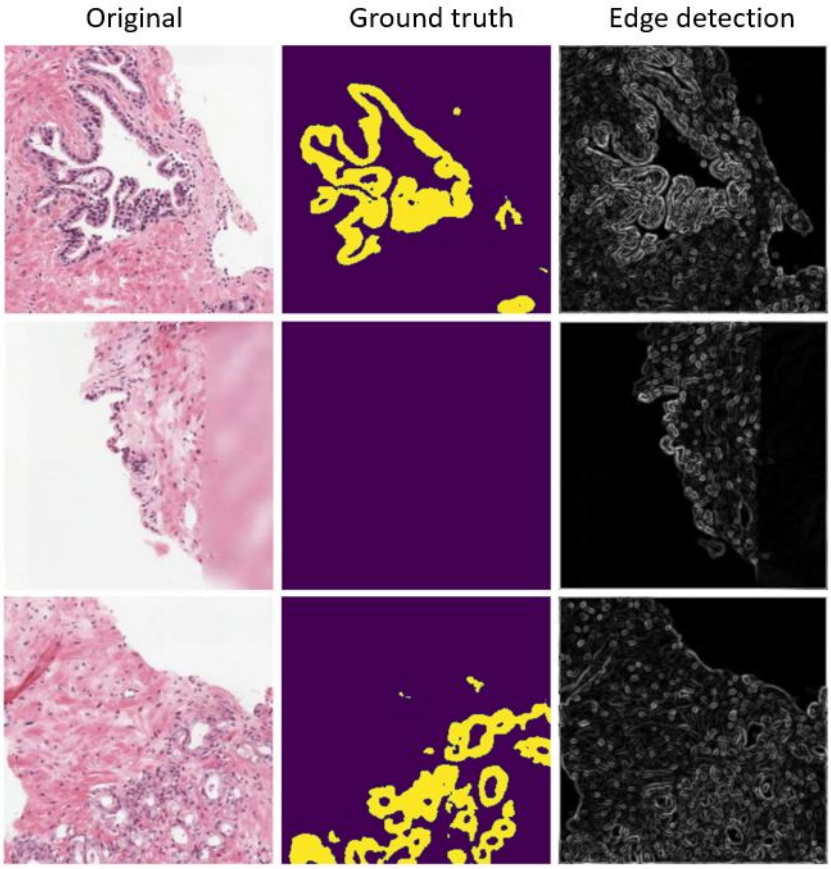

Figure 5.8: Three examples of the canny edge detection algorithm given with the corresponding input and binary mask

It is important to note that the traditional Unet outputs values in the range of $[-\infty, +\infty]$. In other words, no boundaries on the pixel values for each output channel exist. This can influence the optimisation process in a negative way because depending on the loss function the results can be overly optimistic due to the free choice of output values. To prevent this from happening we apply the **softmax function** which is often applied when working with neural networks. The function takes as input a vector $\overrightarrow{z}$ and turns it into a vector $\overrightarrow{z^*}$ where all the elements sum to 1:

$$\sigma(\overrightarrow{z})_i = \frac{e^{z_i}}{\sum_{k=1}^{K} e^{z_k}} \tag{5.7}$$

This function is applied on a vector with 2 elements. One element corresponds with the pixel value taken from the first output channel and the second element is taken from the pixel value at the exact same location but from the second output channel. Because the sum of all the elements is 1, the output values can be interpreted as probabilities or certainties of the network for a specific class. If a pixel location receives the output $[0.9, 0.1]$, the model is fairly certain

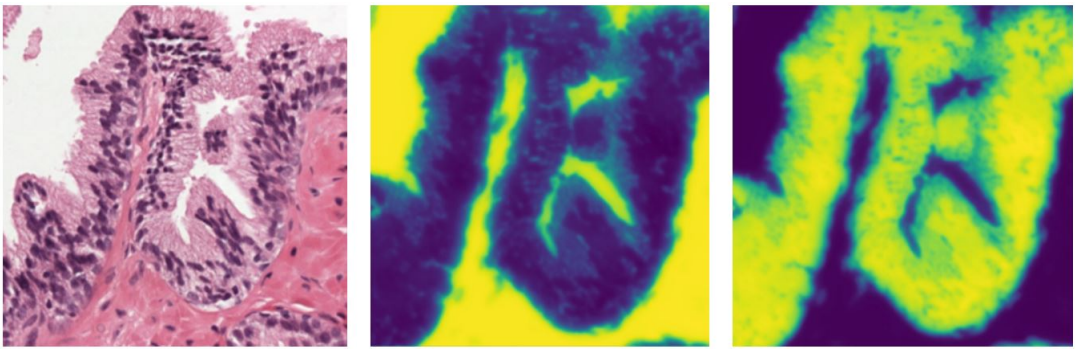

Figure 5.9: (left) The original input image (middle) Example output with emphasis on background and healthy tissue (right) Corresponding output for epithelial cancer cells

that this location needs to be labelled with the first class.

The last step of the post-processing module takes the feature channel corresponding with the class of the epithelial cancer cells (or the feature channel that can be compared with the ground truth binary masks). Because the pre-processing step converts the WSI into small tiles, the output prediction consists out of small tiles as well. The post-processing module stitches the output tiles together into one big binary mask corresponding with the complete original WSI.

5.2 Closing remarks

In this chapter, we proposed our segmentation pipeline and discussed the different challenges that we had to keep in mind during development of the model. Essentially the segmentation pipeline boils down to 3 components: pre-processing, Unet and post-processing. The pre-processing module is the most impactful component when dealing with the aforementioned challenges as we will see in the next chapter discussing the results of this approach.

6

Segmentation results

In the previous chapter, we gave an overview of the main segmentation approach taken based on a Unet. In this chapter we will compare two models: a model with and without the entire pre-processing pipeline. The purpose of this is to create an understanding and motivation for each individual part of the pipeline. We start with the results without the pre-processing pipeline. Afterwards, the model is compared with the final one which follows the complete pipeline of the previous chapter. We finalise this chapter with a discussion about the ROI extraction as preparation for the therapy response prediction model.

6.1 Initial model

The pre-processing module of the initial model made no use of stain normalisation and stain augmentation. Tiling as well as edge-detection for the 4th channel was still performed. We initially used a tile size chosen at 256×256 corresponding with 20x magnification (or $0.5\mu m$/pixel). Post-processing was also different from the second approach because the softmax function was only used during testing and not during training.

The PANDA dataset contains 5160 relevant patients that can be used to train the model. Training times tend to explode with increasing number of training data. We were able to process

256×256 tiles with a maximum speed of 11 tiles/second on a GeForce GTX 1080 Ti 12 Gb RAM. If the average size of a WSI in the PANDA dataset is 20000×20000 and 50% of the tiles are discarded because of the background removal rule, each patient corresponds on average with 1525 tiles. The train-test split reduces the amount of patients in the train-validate set to 4000. Training and validating the model for 1 epoch using 6 100 000 tiles, would take approximately 6.4 days at 11 tiles/second. Training the model until convergence can take months.

To be able to train and evaluate the model in acceptable amount of time, we randomly selected 200 patients from which 152 are used for training data while the others are kept for validation. The training set and validation set each consists out of 4 equally sized subsets corresponding with ISUP grade groups 2 to 5 (table 2.1). Remember from section 5.1.1 that the model should be able to cope with the different gleason grades. The total amount of data that needs to be processed in 1 epoch is 321 750 tiles which takes about 8 hours at 11 tiles/second. Note that each tile also corresponds with a binary mask.

The loss function used to train the Unet was **cross entropy loss** which is defined as:

$$L_{cross\_entropy}(\mathbf{x}, \mathbf{y}) = -\frac{1}{N}\sum_{n=1}^{N} \log(\frac{e^{x_{n,y_n}}}{\sum_{c=1}^{C} e^{x_{n,c}}}) \tag{6.1}$$

With x and y, the predicted output and the ground truth respectively. N is the total amount of output pixels per batch: $batch\_size * 256 * 256$. C is the amount of classes which in this case is 2 corresponding with the 2 output channels (section 5.1.3). Each pixel corresponds with a vector of size 2 containing probabilities for each class. x_{n,y_n} is the probability of pixel n for ground truth class y_n.

The Unet was trained until convergence with batch size 15 and learning rate 0.0001 using the Adam optimizer [52].

6.1.1 Results

After 10 epochs, the model converged (see figure 6.1) with a cross entropy loss around 0.14. The gap between train and validation curves is small leading us to believe that the model will generalise well on unseen data. For further evaluation, we use the 7th epoch having the lowest validation loss.

First, we provide an example output from the validation set in figure 6.2 which has been annotated with gleason grade 3+4. The middle figure contains the ground truth with yellow corresponding with label 1 epithelial cancer cells and purple corresponding with label 0 background or healthy tissue. The bottom figure contains the 2nd channel of the output after applying the

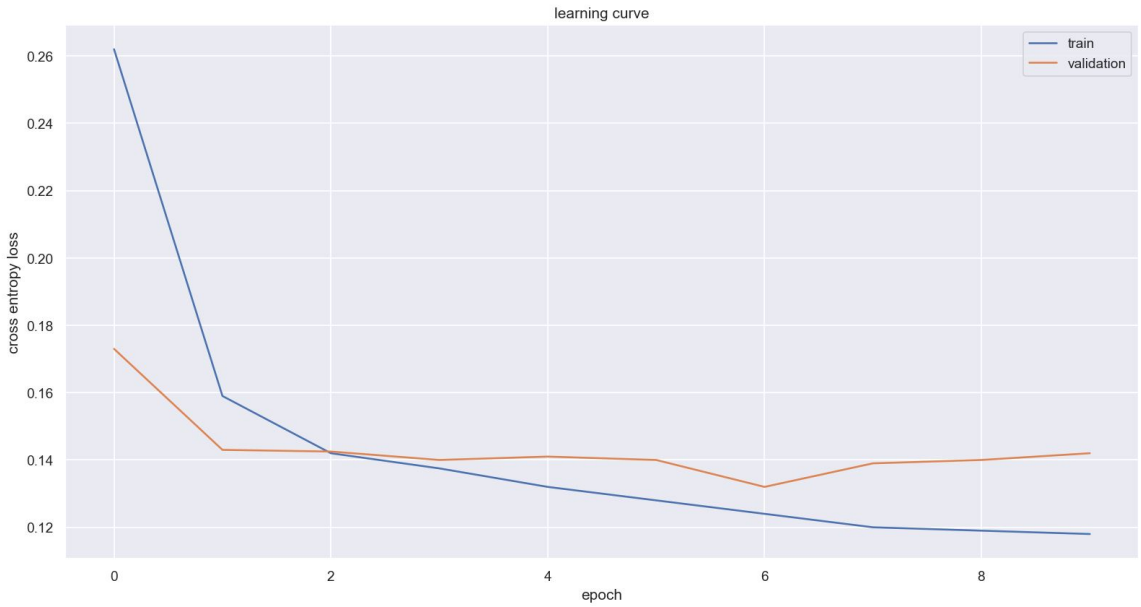

Figure 6.1: The learning curve of the initial model. Training was stopped after 10 epochs.

softmax function. It can be observed that each output pixel value takes on a continuous value in range $[0, 1]$. Visually, the general outlines of the ground truth match with the output.

Based on the transformed pixel values, the AUC on pixel-level was calculated using the validation set containing 48 patients (12 patients for each ISUP grade group). The mean AUC of this model is 0.915 ± 0.0087. To assure the performance of the model across all ISUP grade groups, we calculated the ROC-curve and AUC for each ISUP grade group independently. The result can be found in figure 6.3. The ROC-curves were created for each patient independently and aggregated based on the gleason grades. Using this calculation method, we were able to test for potential outlier patients and determine the inter-patient variability. Figure 6.3 not only showcases overall high model performance but also exhibits low inter-patient variance for each ISUP grade group independently.

Finally we tested the model on the SBRT dataset. Recall that this dataset originates from a different source and consequently used slightly different processes to dye the tissue. Highly detailed masks are not available but the tumor was manually marked on the slide itself. We tested our model on all 162 available slides in the SBRT dataset and manually checked every slide. We observed a significant degradation in model performance. Figure 6.4 illustrates the main problem. Although the tumor gets detected (green pen mark), the model outputs high values outside the tumor region. It appears that healthy epithelial cells are the main cause for the amount of false positives. We conclude that the model is not able to classify epithelium correctly and indiscriminately outputs high values for both healthy epithelium and cancerous epithelium.

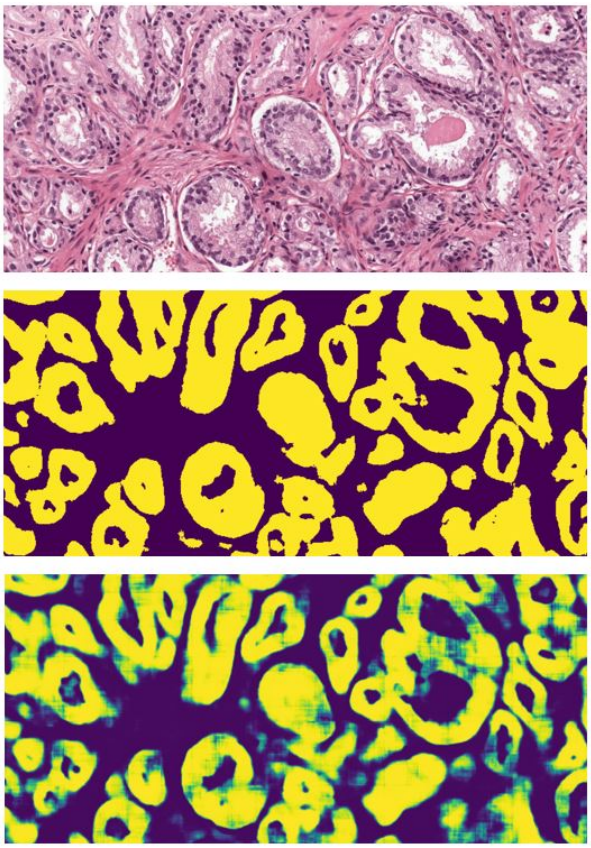

Figure 6.2: Example output of a tissue with gleason grade 3+4. (top) The original image, (middle) The ground truth, (bottom) The output of the model after applying the softmax function.

6.1.2 Possible improvements

We have seen that this model performs remarkably well on the PANDA dataset but fails to reach the same level of performance on the SBRT dataset. We identify different possible causes of this phenomenon.

- **Resolution**: 256×256 at 20x magnification could potentially be a resolution that is too high. A higher resolution essentially means that more detailed information is available but the area each tile covers is smaller. If the area is too small for a specific task, it does not have enough context to perform well. Therefore in our next model, we use tiles with dimensions 512×512 on 10x magnification. Figure 6.5 illustrates the differences between the tiles used in both models.

- **Stain normalisation and augmentation**: A dataset composed by one provider has the consequence of WSI's with relative low variability because they were all created using the

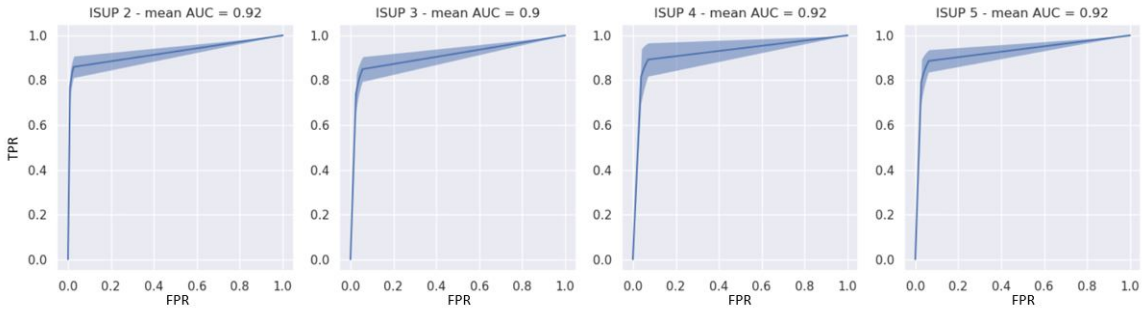

Figure 6.3: The ROC-curves corresponding with each ISUP grade group

same process. Other datasets however have different characteristics which this model is not able to cope with. We have seen in previous chapter that stain normalisation and augmentation are very powerful tools for this exact problem.

- **Tile selection**: We selected only a very small subset of the available data for training to reduce training time, essentially discarding valuable patients. This can be hurtful to the overall generality of the model. Aside from removing background tiles, we decided to randomly extract 50% of the tiles from each patient. This allows us to take more patients into account without introducing bias into the dataset or increasing the computation time too much.

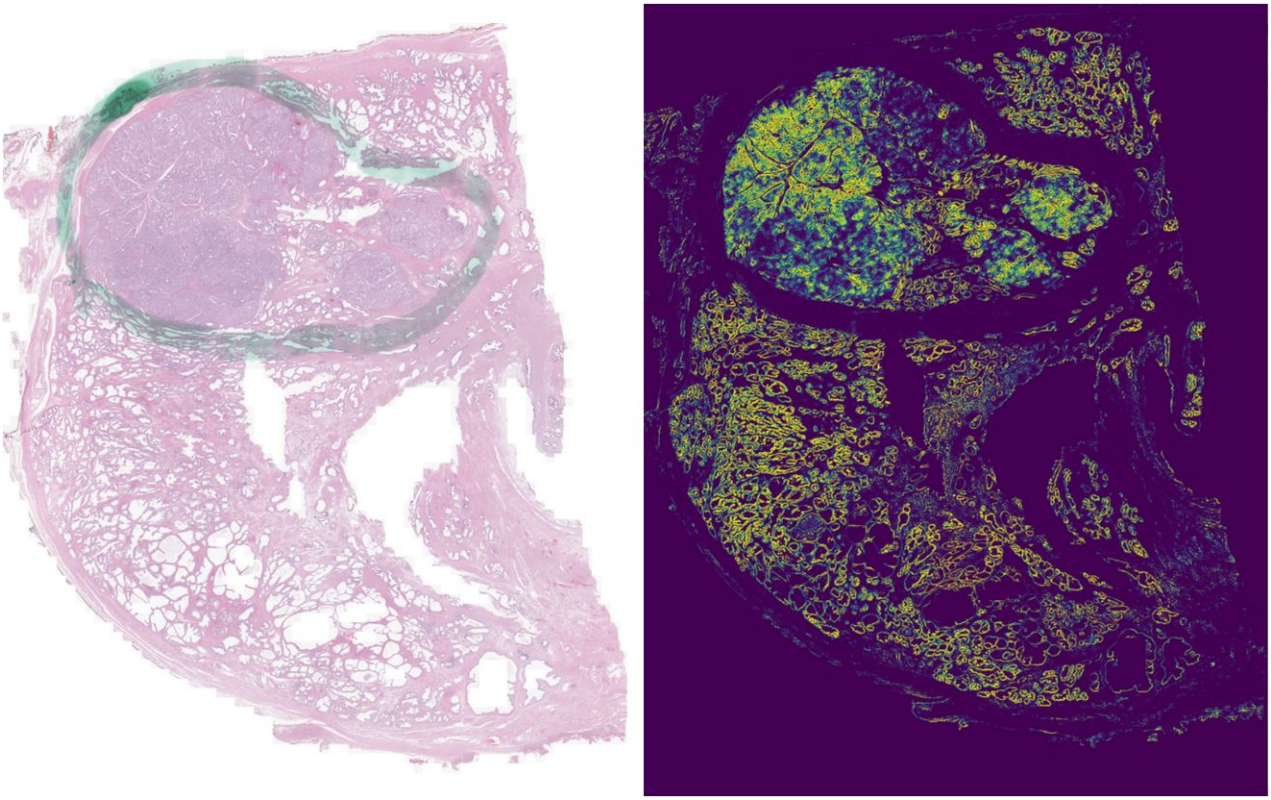

Figure 6.4: (left) The original image with the tumor annotated with a green marker (right) The output after applying softmax

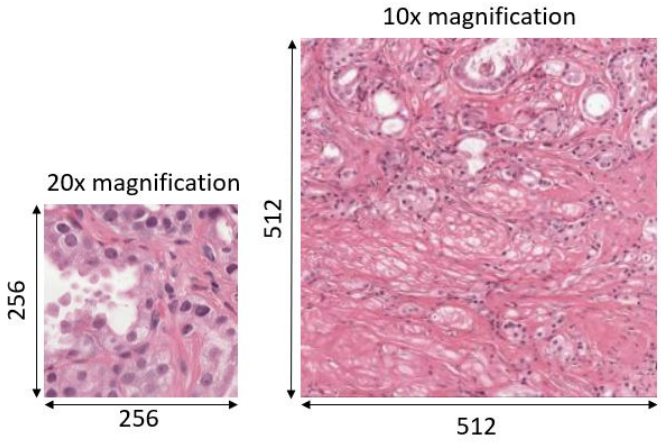

Figure 6.5: 20x magnification versus 10x magnification

6.2 Final model

The final model implements the complete pipeline given in the previous chapter. The tile dimensions are now 512×512 at 10x magnification which is a lower resolution compared to the initial model. Reducing the resolution has the added benefit of being able to take more patients into account without increasing the training time. 1000 patients are randomly selected while removing the background and keeping 50% of the tiles per patient. Because of the low resolution and the 50% rule, each patient corresponds on average with 95 tiles. This significantly reduces the overhead per patient. The training set consists out of 752 patients and the validation set out of 248 patients. Each set can be divided into 4 equally size groups representing the ISUP grade groups.

The Unet was trained until convergence with batch size 15 and learning rate 0.0001 using the Adam optimiser. The cross entropy function remains unchanged but the softmax function is now applied during training, influencing the absolute loss scores. We also used a dropout of 50% chance in the lowest layer (the bottleneck) of the Unet to make it more robust against overfitting. We highlight the changes made compared to previous model in table 6.1.

6.2.1 Results

The train and validation curves started to stagnate after 20 epochs (see figure 6.6 with a cross entropy loss around 0.36. This is significantly higher when compared to the initial model because this loss is calculated after softmax is applied. The absolute gap between the validation and training curves, however, remains unchanged. Epoch 15 will be used in further evaluation.

The AUC is calculated using the validation set containing 248 patients. The mean AUC is 0.9472 ± 0.0005. Not only did the performance increase but the variance decreased as well. The ROC-curves were created using the same method explained in previous section and can be found in figure 6.7. Each curve is created using 62 patients. The performance is again extremely similar between the ISUP grade groups.

Performing the same tests on the SBRT dataset, the result has improved vastly. Figure 6.8 visualises the output of the same input image as in figure 6.4. The current model is able to classify most normal epithelial cells from the cancerous ones. Additional post-processing can be applied using morphological filters to remove small isolated clusters of false positives. We will go deeper into this topic in the next section. Visual inspection of the results on the complete SBRT dataset proved that the overall model performance has improved drastically. We provide the reader with 5 additional examples in figure 6.9.

	Initial model	**Final model**
Dataset	200 patients	1000 patients (50% tile reduction each patient)
Tile	256 × 256 at 20x magnification	512 × 512 at 10x magnification
Pre-processing	background removal + 4th channel	background removal + 4th channel + stain normalisation + stain augmentation
Model	Unet with learning rate 0.0001, batch size 15 and objective function cross entropy loss	Unet with learning rate 0.0001, batch size 15, objective function cross entropy loss and drop-out 50%
Post-processing	softmax during testing	softmax during training and testing

Table 6.1: Model comparison

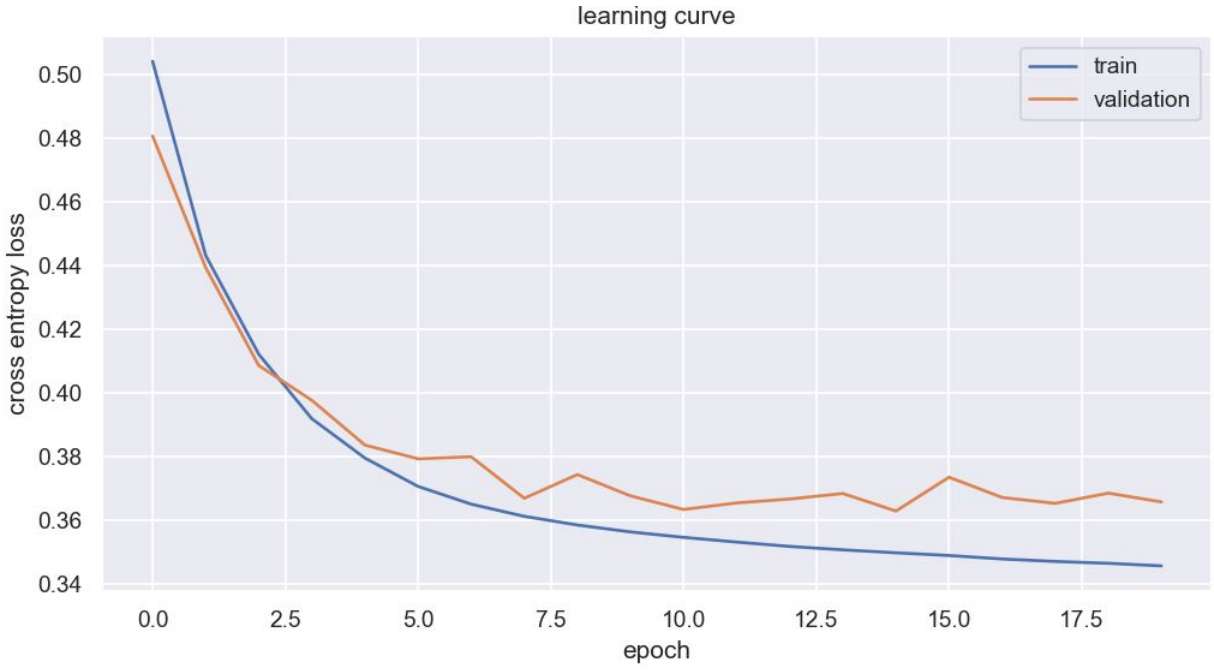

Figure 6.6: The learning curve of the final model. Training was stopped after 20 epochs.

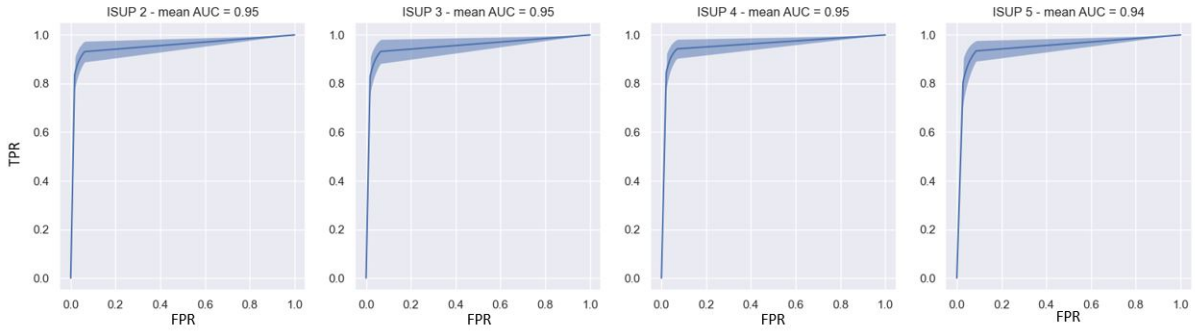

Figure 6.7: The ROC-curves corresponding with each ISUP grade group

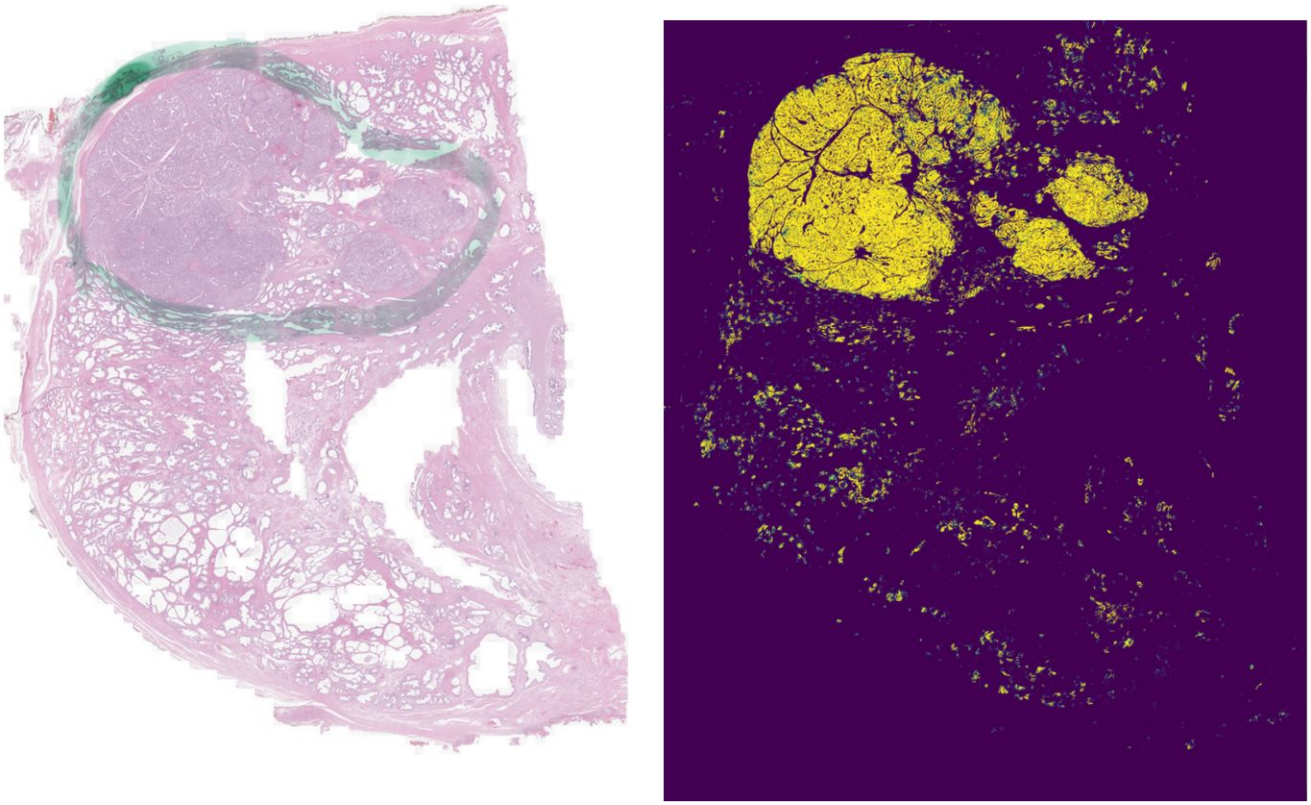

Figure 6.8: (left) The original image with the tumor annotated with a green marker (right) The output after applying softmax

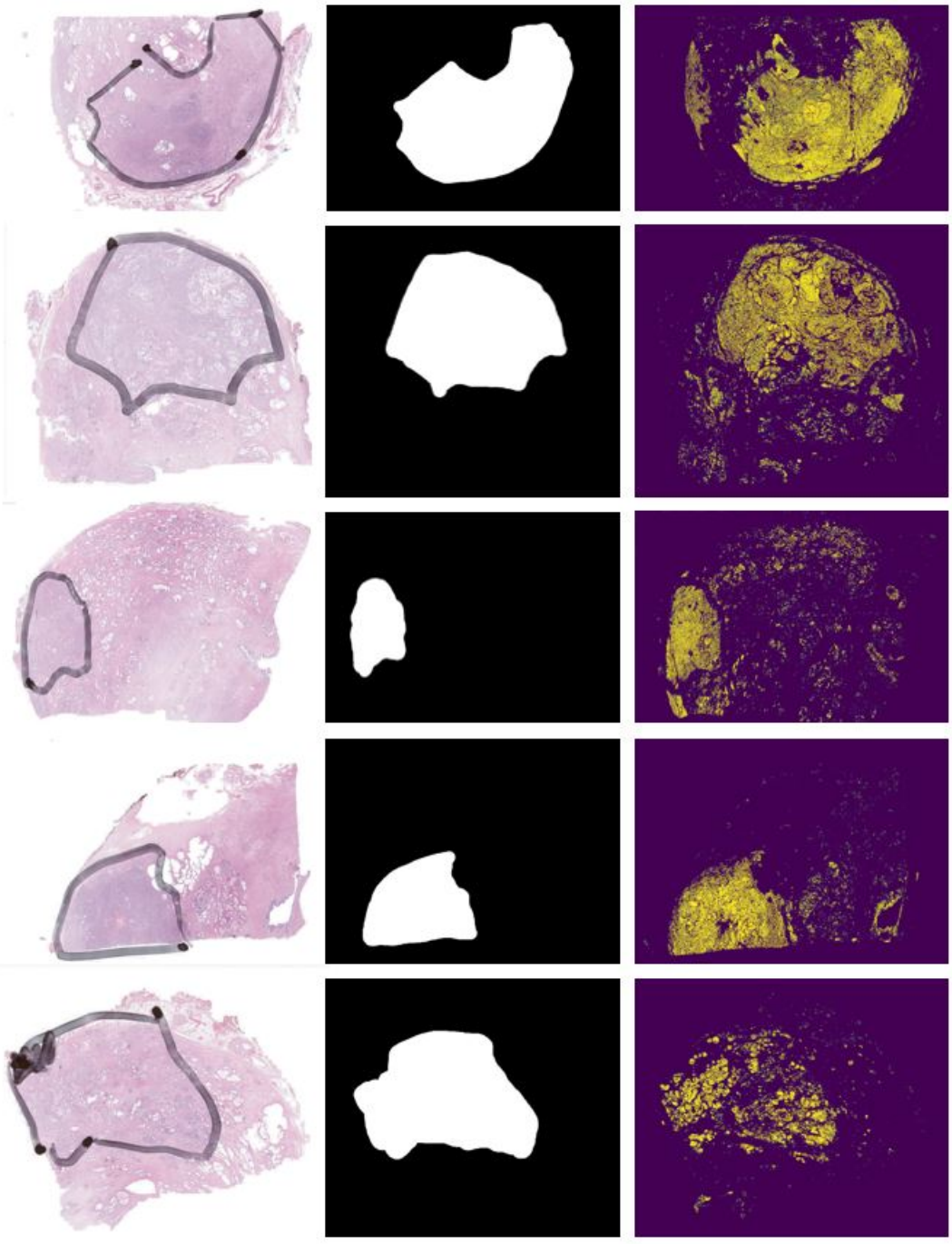

Figure 6.9: Results on the SBRT dataset

6.3 ROI extraction

The ROI extraction process is designed based on a few observations made on the raw output of the segmentation model:

- **Tumor density**: The highest density clusters in the output correspond with the annotated tumors on the original slide.
- **Healthy tissue**: Normal epithelial cells can result into false positives because of the soft boundaries between normal and epithelial cancer cells. Luckily, these cell clusters are more spread out and correspond with a lower density (compared to the tumor).
- **Isolated clusters**: False positives appear into small isolated clusters, clearly distinct and distant from the cluster consisting out of cancerous epithelial cells.

Based on these properties, we propose a pipeline with the goal of selecting tiles containing mostly tumor. Figure 6.10 illustrates the complete pipeline.

After the segmentation model calculates the detailed mask, a **heatmap** is created based on the density of each tile. The density D of each 512×512 output tile is calculated using the following formula:

$$D = \frac{\sum_{i=1}^{512} \sum_{j=1}^{512} x_{i,j}}{512^2} \tag{6.2}$$

With $x_{i,j} \in [0, 1]$, the pixel value of the output mask. The density D is extremely unlikely to reach 1 as this would mean that every pixel has value 1 within each output tile. To exaggerate the highest densities relative to the complete WSI, the densities D are divided by the maximum density that occurs in the WSI. This normalisation maps the highest occurring densities close to 1, effectively stretching the range in between $[0, 1]$. Given the normalised densities and the relative location of each tile, a heatmap can be constructed. It is important to realise that the highly detailed segmentation map is now reduced to 1 value for each tile. In the next step **erosion**, a morphological filter that removes isolated peaks in signals, is applied on the normalised heatmap. This is based on the observation that false positives are often small isolated clusters. If the isolated cluster is small enough to be considered a peak, erosion will not remove but lower the influence of the cluster. The peaks are eroded, but so are the valleys creating gaps in the heatmap. A **Gaussian filter** with 3×3 kernel size is applied with the goal of closing the small gaps without exaggerating the peaks. Figure 6.11 illustrates the impact of the different operators. In special cases were the tumor is very small, it could be that the erosion operation almost completely removes every cluster. For this reason, we apply a second

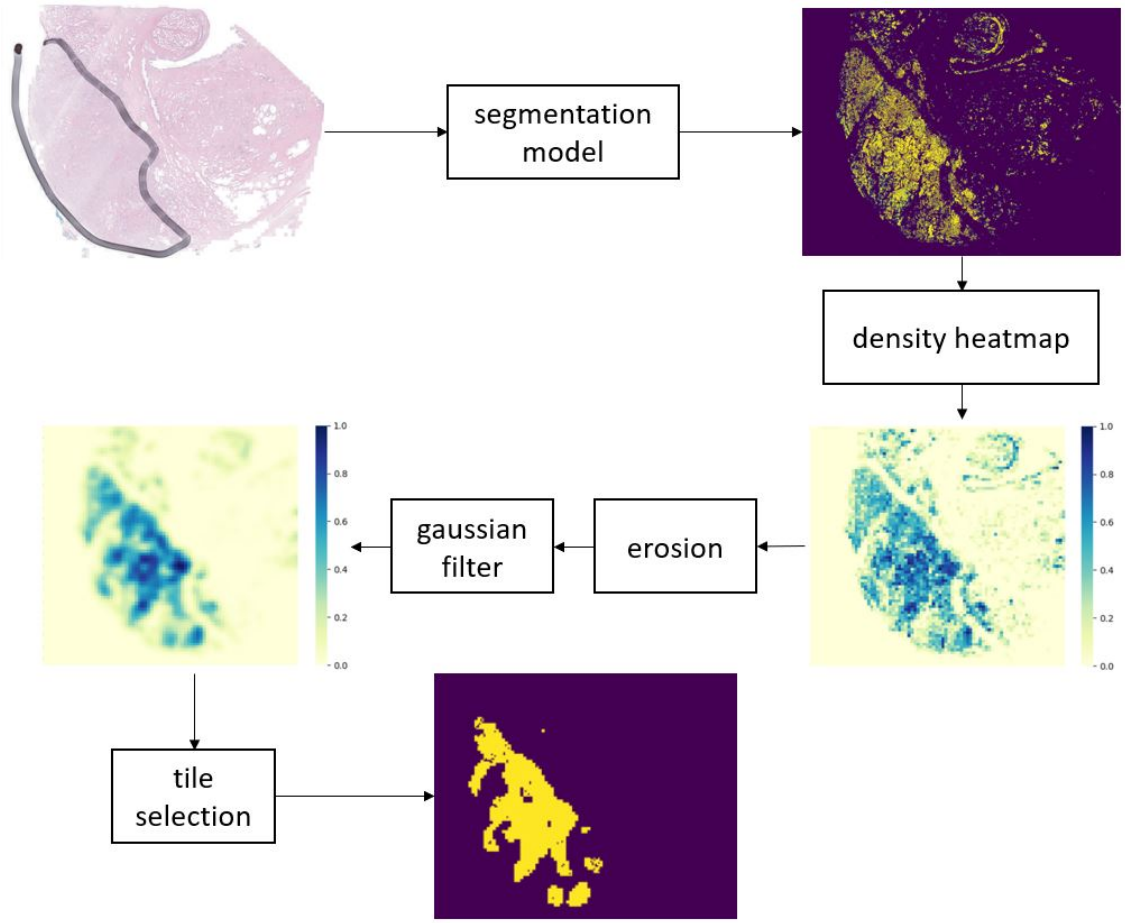

Figure 6.10: The ROI extraction pipeline

normalisation by dividing by the maximum density value of the current heatmap. The resulting heatmap is used to select the ROI by thresholding on the density values. In this work, every density value 0.3 is considered to be in the ROI. Figure 6.12 gives some examples of the pipeline applied on the SBRT dataset.

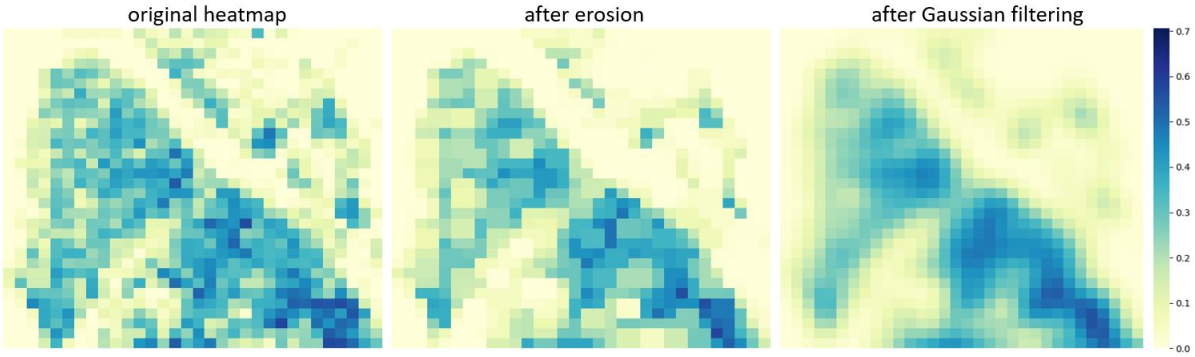

Figure 6.11: sample of the original heatmap (left) the result of the heatmap after erosion (middle) the result after both erosion and Gaussian filtering (right)

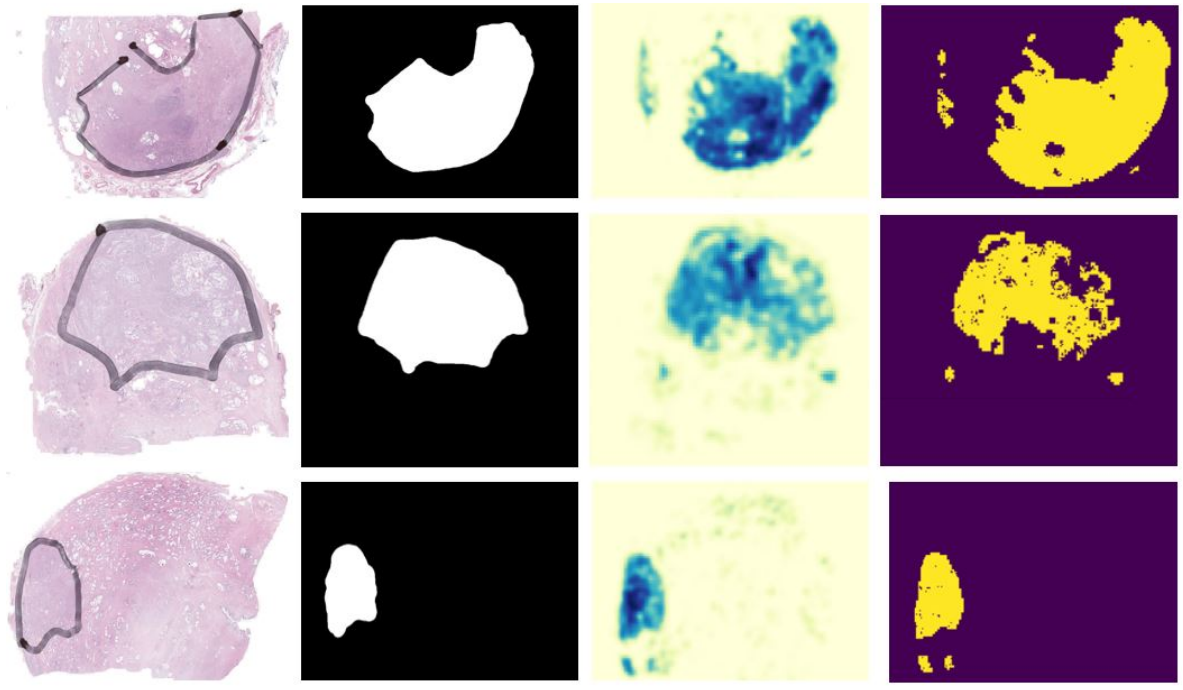

Figure 6.12: 3 examples from the SBRT dataset with respectively: the original image, the ground truth, the normalised heatmap, the ROI with threshold 0.3

6.4 Model comparison

To compare the two models more formally using the SBRT dataset, we made use of the method described in previous section to extract the ROI. We applied the ROI extraction pipeline on the output masks of the initial model en the final model. The resulting binary masks are compared with the ground truth, which is manually created by a pathologist (figure 6.12). To this end two evaluation methods are used: **dice similarity coefficient** and **precision**.

The **dice similarity coefficient** (DSC) also known as the Sørensen–Dice index, is a statistical tool which measures the similarity between two sets of data. it calculates the overlap between two structures respective to their total combined area. The DSC ranges between $[0, 1]$ with 0 corresponding with no overlap at all and 1 corresponding with two identical sets of data. The formula for set X and set Y is:

$$DSC(X, Y) = \frac{2|X \cap Y|}{|X| + |Y|} \tag{6.3}$$

X and Y correspond with the ROI binary mask and the ground truth respectively. We also use the **precision** metric to highlight the reduction in false positives between the 2 models. The formula for this metric is:

$$Precision = \frac{True\_positives}{True\_positives + False\_positives} \tag{6.4}$$

We calculated the DSC and precision for each patient of the SBRT dataset separately, using both models. The initial model which was not able to classify the epithelial cells, achieved an average DSC of 0.488 and an average precision of 0.402. The final model achieved an average DSC of 0.816 and an average precision of 0.920. The resulting scores are almost doubled between the two models, showcasing a significant improvement. In figure 6.13, two scatter plots are shown for each model respectively. Each patient is plotted with their corresponding DSC and precision score.

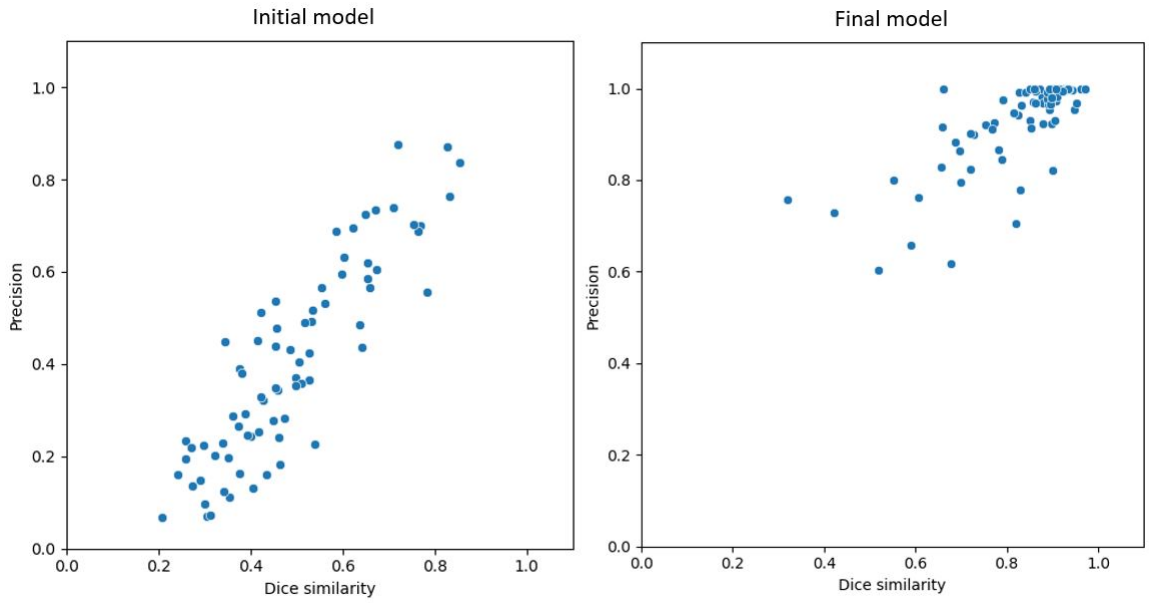

Figure 6.13: Comparison the intial model and the final model. The dice similarity is plotted against the precision.

6.5 Conclusion

After multiple iterations, a model was developed that achieved an AUC of 0.95 on the PANDA dataset. It is able to detect epithelial cancer cells with high detail independent of the gleason score. The model proved to be robust on a completely new independently created dataset. We showed that the generality can be achieved by combining different techniques (e.g. stain normalisation and augmentation). Next, we developed a pipeline to extract the ROI based on heatmaps with the focus on false positive reduction. Finally, we compared the two models after ROI extraction and observed almost a doubling in overall performance. A DSC of 0.816 and precision of 0.920 was achieved.

7

Therapy response prediction

Predicting the clinical response to specific treatments is a major challenge in cancer. To deliver personalised treatment with high efficacy, identifying the correct situation and matching them with the correct therapeutic interventions are essential. In the case for SBRT, doctors generally want to avoid excessive radiation that destroys healthy tissue and causes unwanted side-effects. In this chapter, we discuss our approach to assess whether WSIs can be used as a predictive biomarker for therapy response prediction. This discussion is situated as shown on figure 7.1 and follows after the ROI are extracted using the segmentation model and ROI extraction pipeline from chapter 5 and 6.

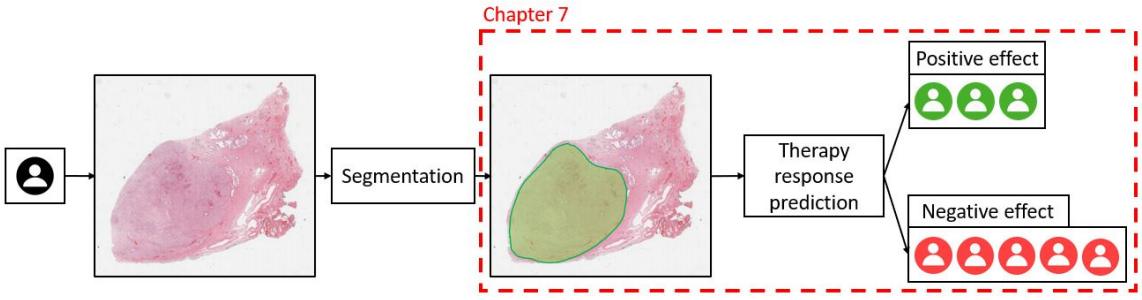

Figure 7.1: High-level overview

To start this chapter, an overview is given of the state-of-the-art in therapy response prediction. Following this, we provide the reader with the challenges when creating a model to predict SBRT

outcome. We group the techniques that we used and tested into two categories: tile-level based prediction and patient-level based prediction. These will be discussed in two different sections.

7.1 SBRT prediction

Cancer is a complex genetic disease involving various subtypes. Oncology has gained in attention over the past decade and researchers are trying to personalise the treatment therapies for cancer patients. Understanding the tumor micro-environment complexity is one of the challenging tasks. Patients receiving the same treatment for the same cancer subtype do not necessarily react in the same way [35, 53]. Given the genetic nature of cancer, models based on gene expression data have great potential in interpreting the correlation of a therapy with cancer [53, 54, 55]. Genes are known to contain the information relating to prevention and cure of diseases, biological evolution mechanisms and drug discovery [55, 56, 57, 58]. For example, Van't Veer et al. [59] used DNA microarray analysis to predict the clinical outcome of breast cancer. They were able to detect features in the genetic data indicating if chemotherapy or hormonal therapy would be a successful treatment for a given patient. Microarray data suffers however from high dimensionality and it requires significant domain knowledge to extract useful features. Traditional feature selection algorithms are often not scalable and robust [53]. This gave rise to deep learning in oncology which enabled automatic feature extraction for a given task [60, 61, 62].

In recent times, imaging data appears to be increasingly used as a potential predictive biomarker for therapy response prediction. For example, MRI-scans have recently been used to extract features for targeted therapies, adding to developments in genomics and molecular biology features [63]. Different studies exist that attempt to extract genetic information based on WSIs [64, 65, 66]. But to the best of our knowledge, using WSIs directly as predictive biomarkers to predict SBRT response is a novel approach.

7.2 Challenges

The SBRT dataset contains 72 patients who all received SBRT. We assume effective treatment for 26 patients when taking PSA failure (see chapter 4) as the decisive end-point. The other 46 experienced PSA failure and are not deemed to have cancer cured. This means that 63% of the patients did not receive the correct treatment and were needlessly exposed to radiation. Predicting this outcome, using the SBRT dataset as an example, comes with a few challenging aspects:

- **Unknown cause**: Researchers were unable to pinpoint the cause of the observed SBRT

instability. This makes it uncertain how WSI will be able to perform as a predictive biomarker. The cause can be of such nature that characterising signals cannot be observed on this level. Because we do not know which features can predict the outcome, we will depend on the automatic feature extraction techniques of state-of-the-art deep learning techniques.

- **Heterogeneous treatment**: The SBRT dataset is the result of 2 clinical trials. For ethical reasons, SBRT was not the only treatment given. Patients also received a hormone therapy called ADT [35]. From the perspective of SBRT response prediction, this pollutes the end-points since a possible recovery could also partially be the result of ADT.

- **Heterogeneous response**: Cancer patients show heterogeneous response against the same or similar treatments [53]. It is difficult to assess how effective SBRT is for each unique patient.

- **End-point selection**: No end-point gives a definitive conclusion if the prostate cancer is cured or not. Because PSA levels are strongly indicative for a possible tumor in the prostate [67], this seems like a valid end-point. However, patients naturally build up different amounts of PSA in the prostate which makes analysis purely based on PSA difficult. Another potential end-point is 'distant failure' and indicates if the cancer has spread to body parts other than the prostate (metastasis). We will see later that both end-points lead to the same results. For this reason, we will be using PSA failure as primary end-point.

- **WSI size**: Just like with segmentation, the high dimensionality of a WSI poses some challenges. Tiling the image comes with the risk of losing the contextual information. This is less of a problem in segmentation given that the tiles are not taken on a resolution that is too high. High resolution tiles cover a relatively small area that can be too small to identify specific cell clusters to detect epithelial cancer cells. But for therapy response prediction, the contextual information could potentially be key information because only patient-level labels are known. Alternatively, feature extraction and dimensionality reduction can be performed using techniques that do not use tiling. In this case, some form of compression needs to be executed, again resulting into loss of information. Because it is difficult to know what information needs to be captured, dealing with the WSI size is a challenging task.

- **Tile-level vs patient-level**: For each WSI, the label is a binary value: label 1 if the treatment is successful, label 0 if not. Tiling the WSI implicitly means that each tile is processed separately using the patient-level label. Not every tile will be indicative of the overall label and can introduce noise during training. ROI extraction partially alleviates this problem but may not be sufficient. In contrast, predictions taking the entire WSI into

account without tiling, require some form of compression. To that end, we group techniques into 2 groups: **tile-level based prediction** and **patient-level based prediction**.

7.3 Tile-level based prediction

As the name suggests, every technique in tile-level based prediction requires the WSI to be divided into tiles. To enable supervised learning, each tile needs to correspond with a label. In the case of SBRT response prediction, this label is the same for each tile that originated from the same patient. For example, tiles from a patient who experienced PSA failure all receive label 1. This label assignment is performed independent of whether the tile contains decisive information for the overall outcome or not. In the field of AI, this is called **weakly supervised machine learning** [68] where models work with coarse-grained labels. The advantage of this approach is that the WSI does not need to be compressed, avoiding information loss. There is also more data available, reducing the risk of overfitting the model. However, the chance that only certain hotspots on the WSI are relevant for the outcome is high. So this approach may introduce many noisy labels.

7.3.1 Proposed framework

We propose a framework based upon a RESnet18 [69] (see section 3.3.3: RESnet). In figure 7.2 the framework is illustrated. For now, we assume that PSA-failure is a sufficient label to define the outcome of SBRT. First, the ROI is extracted using the method from section 6.3. We do this based on the assumption that the cause of SBRT response variability is found in the tumor and its immediate surroundings. Next, the ROI is divided into tiles of 512×512 at 20x magnification. This creates a bag of tiles were each tile has the same global label. Each tile from the bag is processed independently using a RESnet18. The output is a 2-dimensional vector with the total sum of the elements being 1. This can be interpreted as the probability or certainty for each class (PSA failure and no PSA failure). We assume that a tile containing no decisive information will result in an output of [0.5, 0.5]. An interesting evaluation would be to visualise the tile output probability in the WSI to determine if hotspots are present.

From the 72 available patients, 51 were used for training and 21 patients for validation. We mentioned earlier that 63% of the patients experienced PSA failure which is almost double the amount of successful cases. We ensured that this distribution was maintained in both the train and validate set to avoid introducing discrepancies between the respective loss scores.

Training a RESnet18 is computationally expensive and can take a long time to stabilise on challenging datasets [70]. The usual way to go around this problem is through **transfer learn-**

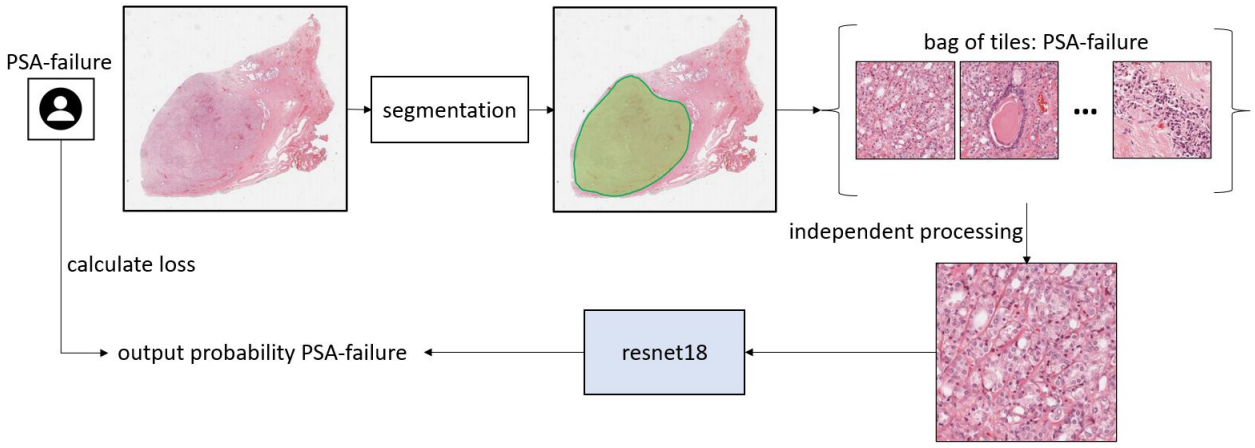

Figure 7.2: Pipeline using a tile-level prediction strategy

ing [71]. Transfer learning is a machine learning method where a model developed for a task is partially reused for another task. For example in a RESnet, the earlier layers detect the high-level features of an image. These layers could be reused on similar data. We decided to retrain the last 2 layers of a RESnet18 trained on ImageNet [72]. During training, the pretrained weights of the upper layers were kept fixed.

We trained the pretrained RESnet18 with batch size 15 and learning rate 0.001. The cross entropy loss between the input label and the output probability is chosen for the objective function which was optimised using the Adam optimiser [52]. To cope with the serious class imbalance, the loss score corresponding with the minority (no PSA failure) was multiplied with weight 2 during training.

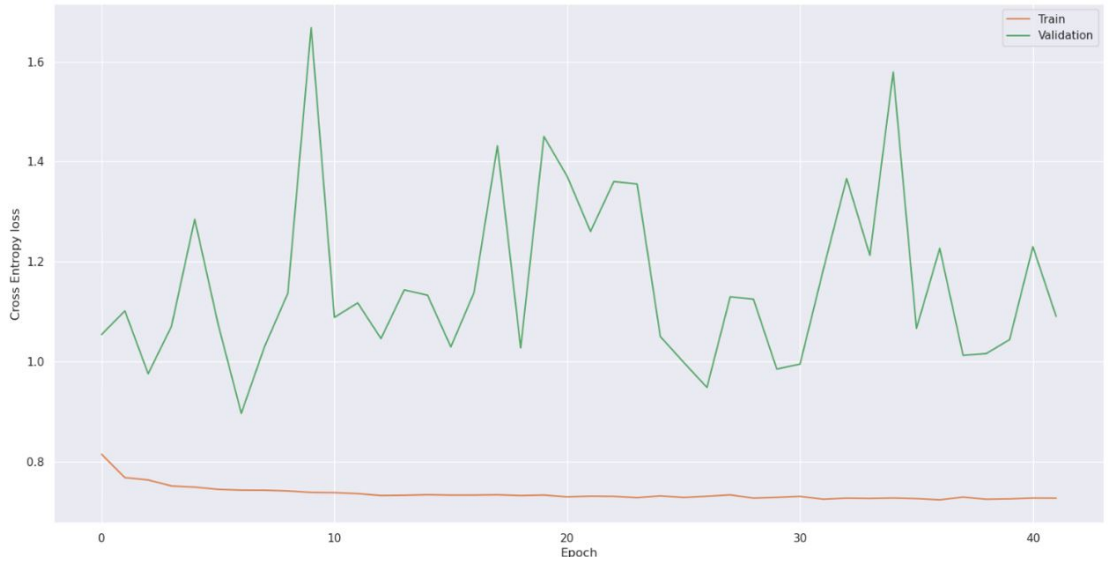

Figure 7.3: Learning curve of the proposed framework

7.3.2 Results

Training was stopped after 40 epochs after the model showed no indication of improvement. Figure 7.3 illustrates the learning curve. We observe that the train curve has the usual behaviour of decreasing and eventual flattening. The validation curve however is extremely irregular. And even worse it shows no sign of improvement. This is an example of both overfitting and underfitting (section 3.3.5). Overfitting can be observed in the high variance of the validation curve. The training curve on the other hand is very streamlined and eventually stagnates. Underfitting is represented by the high training error values. Based on the learning curve, we can conclude that the model was unable to extract the relevant features.

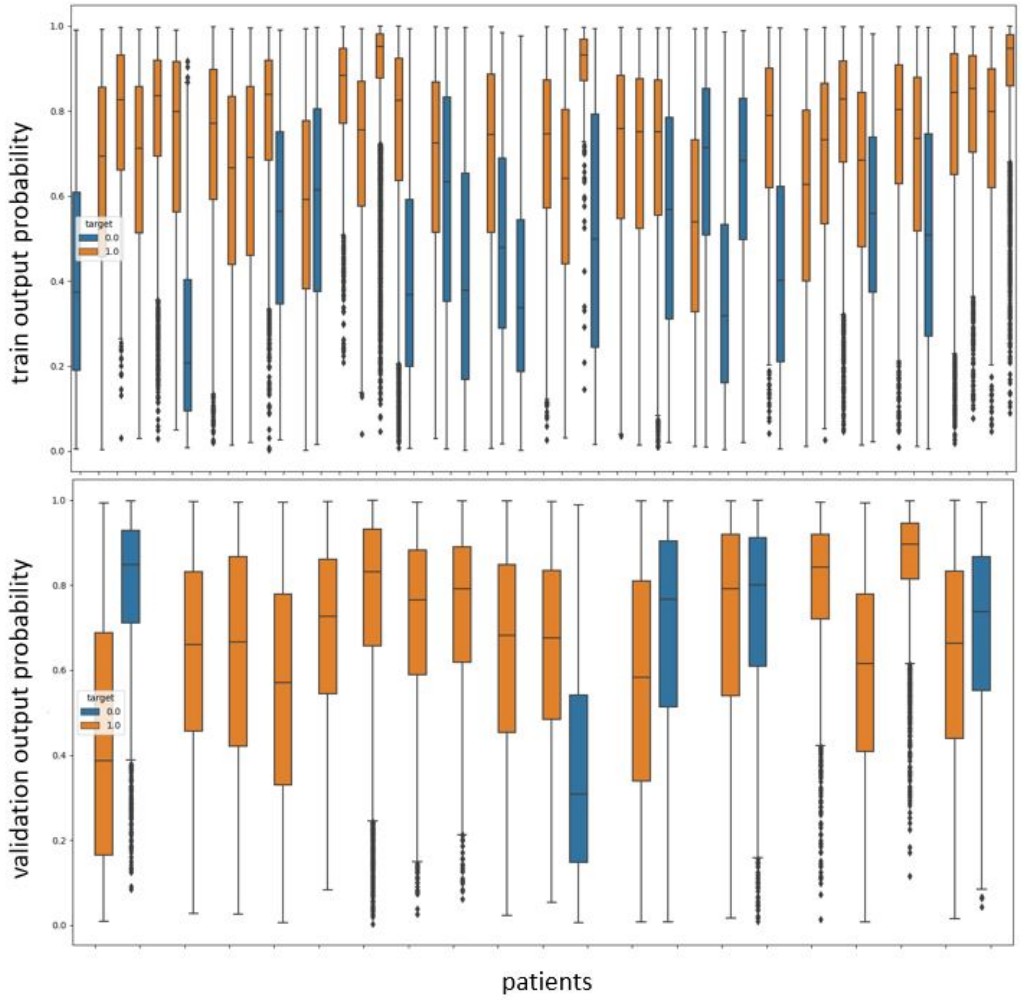

Figure 7.4: Each boxplot corresponds with the output probabilities of all the tiles of one patient. Blue patients are labelled with no PSA failure (label 0). Orange patients correspond with PSA failure (label 1)

We opted to evaluate the network further. In figure 7.4, we compared the results between the

train set (upper plot) and validation set (bottom plot). Each patient represents one boxplot in this figure. Each boxplot is created based on all the output probabilities for the PSA failure class of the tiles of the corresponding patient. Orange boxplots represent patients with PSA failure. The blue boxplots correspond with patients without PSA failure. The training set showcases as expected some difference between the two classes. The 'no PSA failure' class patients received on average a lower tile probability. The reverse is true of 'PSA failure' class patients. The validation set results, however, confirm the overfitting and underfitting. We can see that no distinction can be made between the two classes.

Earlier we mentioned that hotspots could be present which could indicate important areas. In figure 7.5 the tile probabilities for label 1 (PSA failure) are visualised on their respective location on the WSI slide. The upper row contains 2 examples from the train set. The patient with PSA failure shows as expected high probability tiles while the patient with no PSA failure corresponds with on average low probability tiles (which we already observed in figure 7.4). Also conform with figure 7.4, the validation examples both showcase on average high probability tiles. No peaks or hotspots can be observed from the heatmaps. This means that no region of tiles can be found that is more indicative towards the correct (or wrong) class. The figure highlights the observed overfitting and underfitting.

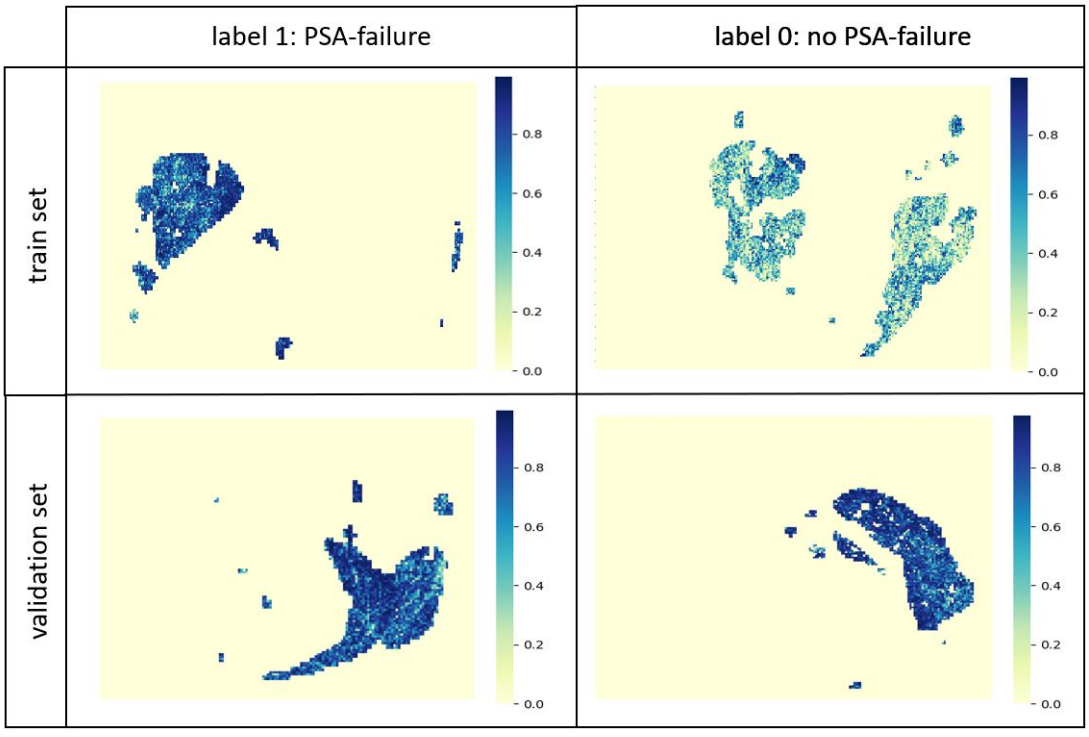

Figure 7.5: Comparsion between train and validation set

After applying 3-fold cross validation, the model reached an average AUC of 0.55, which is just above a random model. The extensive evaluation confirms that the model did not learn

anything of value. SBRT response prediction using the proposed framework is unsuccessful. Different factors can cause such a result:

- **Tile relevance**: A possibility is that the relevant region towards a certain class is too small. During training the model sees more noisy tiles then important tiles and is consequently not able to reach a high performance. The model forcefully attempts to make sense of the training data and in doing so starts to overfit.

- **Tile resolution**: The resolution of the input tiles could be too high or low to detect the relevant features. Our tests on higher and lower resolutions did not result in a higher performance.

- **End-point**: PSA levels could not be the correct choice to assess whether SBRT was effective. We tested other end-points such as 'distant-failure' (metastasis or not) and reached the same level of performance as the given model.

- **Pretrained weights**: Another possibility is that the data used to trained the pretrained resnet18, is too different from the SBRT data. As a consequence, the model detects the wrong high-level features. To ensure that this is not the case, we attempted to train a resnet18 from scratch. The model behaved the same way with a highly irregular learning curve and slightly worse performance.

- **WSI**: The WSI cannot be used as a biomarker to predict the outcome of SBRT.

7.4 Patient-level based prediction

With patient-level prediction, the WSI is not split into different regions or tiles which are independently processed. Using feature extraction and/or dimensionality reduction techniques, the WSI is processed as a whole, maintaining the spatial information. Compressing the WSI comes at a cost of potentially losing important local information. We want to extract the relevant local information (if any) during compression.

In this section we provide the reader with two approaches. These approaches are complex relative to the amount of available data. The 72 patients from the SBRT dataset were not enough to train the complex models discussed in this chapter. Using the models with the SBRT dataset, resulted in heavy overfitting where the validation loss instantaneously started to increase. Still, we wanted to list these promising approaches for future work.

7.4.1 Attention-based multiple instance learning

Multiple instance learning (MIL) is a weakly supervised technique where a single class label is assigned to a bag of instances [73]. Our proposed framework from the previous section makes use of this concept by grouping the tiles in a single bag of instances. Nonetheless, it differs from MIL because the tiles are processed independently from each other. Attention-based MIL is capable of automatically identifying tiles of high diagnostic value to classify the WSI on slide-level [74]. First, the WSI is divided into tiles. Then, MIL dynamically aggregates these tiles (one could see this as a dynamic pooling layer). During training and testing, the model ranks all the tiles, assigning an attention score to each tile. The attention score reflects the contribution or importance of the respective tile towards the designated output. Finally it calculates a weighted average of all tiles based on their respective attention score. The aggregated feature vectors can consequently be used to perform classification. If the classification is successful, the attention weights for each respective tile can be used to construct an attention based heatmap, highlighting the most relevant tiles.

7.4.2 Tile compression

The high dimensions of a WSI prevents it from using it as a direct input for e.g. a RESnet. One solution to this problem could be to tile the WSI and compress each tile separately. In figure 7.6, we illustrated such an approach. The WSI is divided into 512×512 tiles. 2 approaches were tested to encode (compress) the tiles. The first approach is an autoencoder were to goal is to compress the input and based on the compressed feature vector, to reconstruct the original again [75]. The second approach was based on a pretrained model of ImageNet where the feature vectors were extracted from higher layers. Each encoded tile was placed in a grid of 200×200 were the relative location of the tiles was maintained. If no tile was selected for a location, the corresponding feature vector was set to 0. In the example the intermediate result is a matrix of $200 \times 200 \times 1024$. The dimensions are reduced to such an extent that the matrix can be directly used as input for a RESnet.

Both the attention-based MIL model and the tile compression technique were tested on the SBRT dataset but were unsuccessful because of the lack of data.

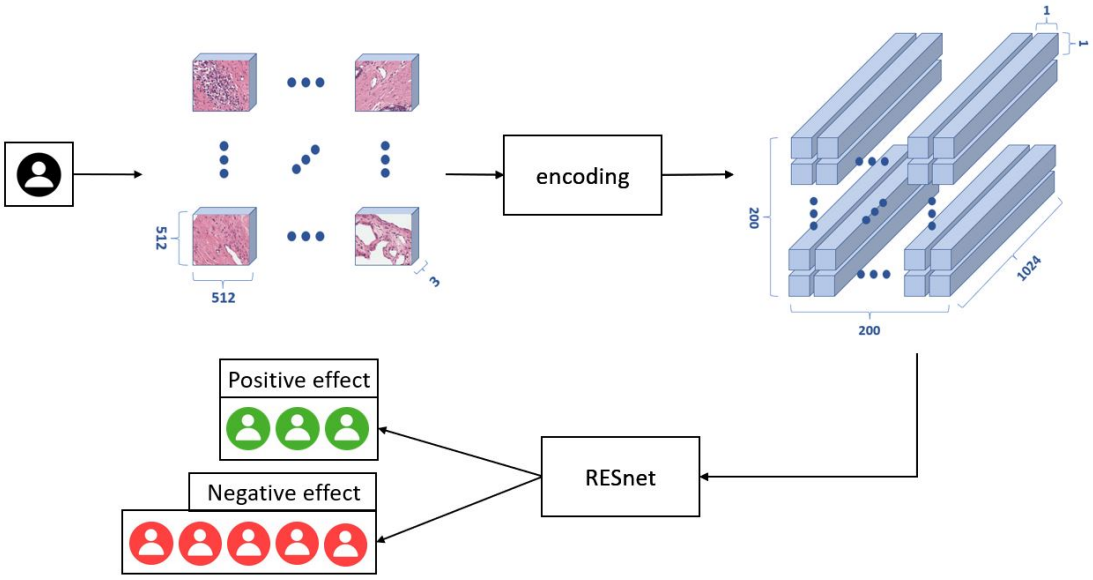

Figure 7.6: The tiles are encoded (compressed) and fed into a traditional network.

7.5 Conclusion

Even though several studies succeeded in using gene expressions to predict therapy response and several studies succeeded in predicting specific genetic alterations from WSIs, it seems WSIs have limited value for therapy response predictions. We divided our discussion into 2 parts: tile-level prediction and patient-level prediction. On tile-level prediction, a framework was proposed which processed the tile independently from each other, resulting in spatial information loss. Using this approach we were not capable to predict the SBRT outcome based solely on WSI's. The patient-level prediction strategies on the other hand are capable of maintaining the spatial information but require intricate aggregation strategies to make computation feasible. These models, however, were too complex relative to the amount of available data. As such, they were all subjected to heavy overfitting.

It is difficult to say for sure that a WSI cannot serve as a predictive biomarker. It is still possible that the solution can only be found in very small regions of the tumor. Patient-level prediction strategies are promising and especially MIL is gaining in popularity in the field of digital pathology. But to achieve acceptable results using this strategy, more data will be required.

8 Conclusion

In this chapter we give some conclusions on the work done and lessons learnt during this master thesis. First, we summarise our findings on the two parts of this work: segmentation and therapy response prediction. We start with outlining the main approach taken to achieve highly detailed tumor detection. We continue with our transformation from the detailed segmentation to the extraction of the ROI. Finally, we provide the results and conclusion about therapy response prediction on WSI's.

8.1 Summary of the master thesis

Segmentation In this work, we developed a method based on a Unet to detect epithelial cancer cells. This approach was possible because of the detailed masks in the public PANDA dataset and the amount of data available for each gleason grade. Although we achieved an average AUC score of 0.92 on our first attempt, the model did not perform on the independent SBRT dataset. It was not capable of differentiating between the different epithelial cell types. We hypothesised given the high AUC score, that our model imitated the unknown annotation algorithm used to create the mask from the PANDA dataset without regard for generalisable features. To achieve a robust model, we performed the following pre-processing steps:

- removing tiles at random to be able to take more patients into account without drastically increasing training time
- stain normalisation based on a technique by Reinhard et al. which makes use of the perceptual properties of the $l\alpha\beta$ colour space
- randomly changing the properties of the input using stain augmentation
- adding a 4th channel containing the gradient intensities after partially applying the canny edge detection algorithm

With these adjustments, we were able to reach an average AUC of 0.95 on the PANDA dataset and an average DSC of 0.82 on the SBRT dataset. We established that the model was able to differentiate between epithelial cancer cells and benign cells which was not the case in the first model.

ROI extraction The goal was to extract the general area of the tumor with both epithelial cancer cells and stroma included. The masks from segmentation are however too detailed. Additionally, the segmentation model occasionally detects small isolated areas outside the annotated tumor regions. To prepare the segmentation mask for ROI extraction and to remove the false positives, we proposed a heatmap-based pipeline:

- based on the local densities of the output of the segmentation pipeline, we constructed a heatmap
- on the resulting heatmap, the erosion operator followed by a Gaussian filter was applied; this removed isolated peaks in the heatmap
- by applying a threshold on the heatmap, we can extract the general ROI for later use

Therapy response prediction We divided our therapy prediction evaluation into two major parts: tile-level prediction and patient-level prediction. For **tile-level prediction**, a framework was proposed that minimised the amount of noisy tiles through ROI extraction. Each 512×512 tile was processed independently using a pretrained RESnet18. The learning curve showed signs of both overfitting and underfitting. Further evaluation, showed more clearly that the model did not learn relevant features. We hypothesised that this could have been caused by multiple factors: the presence of irrelevant tiles, polluted end-points and irrelevance of WSI in predicting the outcome of SBRT. **Patient-level prediction** takes into account the spatial information of the complete WSI but requires, given the complexity of the models, more data. Although we were not able to create a model without overfitting, we discussed possible approaches that could be taken if more data were present.

8.2 Future work

To increase the overall performance of our segmentation model, different approaches can be taken. We experienced that stain normalisation has a big impact on the overall performance of our segmentation model. Reinhard stain normalisation from Reinhard et al. [48] fails if the target image is too different from the input image. For example, if the input contains a high percentage of white pixels (due to presence of big glands or just plain background) and the target image contains a high percentage of stroma and epithelial cells, the result could map the white pixels to the color of the target stroma. More recently, Macenko et al. proposed an interesting technique based on optical density that partially alleviates this problem [76].

In 2019, the ACDC@LungHP (Automatic Cancer Detection and Classification in Whole-slide Lung Histopathology) challenge was held for evaluating different computer-aided diagnosis methods [77]. In the following paper of this challenge, the top-10 performing models were compared. It was observed that multi-model pipelines performed significantly better in segmenting the tumor compared to single-model pipelines. Multi-model pipelines achieved an overall higher DSC, sensitivity and specificity. We decided to focus on the pre-processing module of our pipeline and used a single model to perform the segmentation. It would be interesting to see the potential performance gain when using a multi-model approach with the same pre-processing pipeline.

Regarding therapy response prediction, attention-based MIL seems to be the most promising for further research. The dynamic pooling layer enables models to take the complete WSI into account which is useful when working with patient-level labels. Lu et al. developed a modified version of this approach to identify subregions on a WSI of high diagnostic value [74]. They achieved AUC scores as high as 0.991 based on coarse-grained labels. With more data, these approaches could be tested for therapy response prediction.

Bibliography

[1] "The gleason score and grade groups," https://www.cancerresearchuk.org/about-cancer/prostate-cancer/stages/grades, 2019.

[2] W. H. Organisation, "Cancer," https://www.who.int/news-room/fact-sheets/detail/cancer, 2022.

[3] R. L. Siegel, K. D. Miller, H. E. Fuchs, and A. Jemal, "Cancer statistics, 2021," *CA: A Cancer Journal for Clinicians*, vol. 71, no. 1, pp. 7–33, 2021. [Online]. Available: https://acsjournals.onlinelibrary.wiley.com/doi/abs/10.3322/caac.21654

[4] "Causes or initiation of cell division | actforlibraries.org." [Online]. Available: http://www.actforlibraries.org/causes-or-initiation-of-cell-division/

[5] B. N. Ames, L. S. Gold, and W. C. Willett, "The causes and prevention of cancer," pp. 5258–5265, 6 1995.

[6] F. C. Detterbeck, S. Z. Lewis, R. Diekemper, D. Addrizzo-Harris, and W. M. Alberts, "Executive summary: Diagnosis and management of lung cancer, 3rd ed: American college of chest physicians evidence-based clinical practice guidelines," *Chest*, vol. 143, pp. 7S–37S, 5 2013.

[7] "Skin cancer - symptoms and causes - mayo clinic." [Online]. Available: https://www.mayoclinic.org/diseases-conditions/skin-cancer/symptoms-causes/syc-20377605

[8] I. Dagogo-Jack and A. T. Shaw, "Tumour heterogeneity and resistance to cancer therapies," pp. 81–94, 2 2018.

[9] "Types of cancer treatment," https://www.cancer.gov/about-cancer/treatment/types, 2020.

[10] "Mayo clinic - mayo clinic." [Online]. Available: https://www.mayoclinic.org/

[11] D. Mitchison, "Basic mechanisms of chemotherapy," *Chest*, vol. 76, pp. 771–780, 12 1979.

[12] L. F. C. M. M. L. P. M. e. a. Ferlay J, Ervik M, "Global cancer observatory: Cancer today," https://gco.iarc.fr/today/home, 2020.

[13] P. Rawla, "Epidemiology of prostate cancer," *World Journal Of Oncology*, vol. 10, no. 2, pp. 63–89, 2019. [Online]. Available: https://www.wjon.org/index.php/WJON/article/view/1191

[14] G. K. Panigrahi, P. P. Praharaj, H. Kittaka, A. R. Mridha, O. M. Black, R. Singh, R. Mercer, A. van Bokhoven, K. C. Torkko, C. Agarwal, R. Agarwal, Z. Y. Abd Elmageed, H. Yadav, S. K. Mishra, and G. Deep, "Exosome proteomic analyses identify inflammatory phenotype and novel biomarkers in african american prostate cancer patients," *Cancer Medicine*, vol. 8, no. 3, pp. 1110–1123, 2019. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/cam4.1885

[15] "Prostate cancer," https://www.mayoclinic.org/diseases-conditions/prostate-cancer/symptoms-causes/syc-20353087, 2021.

[16] U. H. Stenman, J. Leinonen, W. M. Zhang, and P. Finne, "Prostate-specific antigen," *Seminars in Cancer Biology*, vol. 9, pp. 83–93, 4 1999.

[17] J. Silva-Rodríguez, A. Colomer, M. A. Sales, R. Molina, and V. Naranjo, "Going deeper through the gleason scoring scale: An automatic end-to-end system for histology prostate grading and cribriform pattern detection," *Computer Methods and Programs in Biomedicine*, vol. 195, p. 105637, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S016926072031470X

[18] P. Harnden, M. D. Shelley, B. Coles, J. Staffurth, and M. D. Mason, "Should the gleason grading system for prostate cancer be modified to account for high-grade tertiary components? a systematic review and meta-analysis," pp. 411–419, 5 2007.

[19] A. Martin and A. Gaya, "Stereotactic body radiotherapy: A review," *Clinical Oncology*, vol. 22, no. 3, pp. 157–172, 2010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0936655509004099

[20] W. R. Kennedy, P. Gabani, J. Nikitas, C. G. Robinson, J. D. Bradley, and M. C. Roach, "Repeat stereotactic body radiation therapy (sbrt) for salvage of isolated local recurrence after definitive lung sbrt," *Radiotherapy and Oncology*, vol. 142, pp. 230–235, 1 2020.

[21] B. G. Buchanan and E. A. Feigenbaum, "Dendral and meta-dendral: Their applications dimension," *Artificial Intelligence*, vol. 11, pp. 5–24, 8 1978.

[22] M. K. Chandrasekhara, B. Shanthi, and H. N. Mahabala, "Can community health workers screen under 5yr children with computer program," *The Indian Journal of Pediatrics*, vol. 61, pp. 567–570, 9 1994.

[23] M. N. Ahmed, A. S. Toor, K. O'Neil, and D. Friedland, "Cognitive computing and the future of health care cognitive computing and the future of healthcare: The cognitive power of ibm watson has the potential to transform global personalized medicine," *IEEE Pulse*, vol. 8, pp. 4–9, 5 2017.

[24] K. A. Dill and J. L. MacCallum, "The protein-folding problem, 50 years on," pp. 1042–1046, 11 2012. [Online]. Available: https://www.sciencemag.org/lookup/doi/10.1126/science.1219021

[25] J. Dambre and T. Dhaene, "Machine learning," Ghent University, 2020.

[26] "Types of machine learning algorithms | 7wdata." [Online]. Available: https://7wdata.be/visualization/types-of-machine-learning-algorithms-2/

[27] "Educative: Interactive courses for software developers." [Online]. Available: https://www.educative.io/

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[29] E. Meijering, "Cell segmentation: 50 years down the road [life sciences]," *IEEE Signal Processing Magazine*, vol. 29, no. 5, pp. 140–145, 2012.

[30] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," 2015. [Online]. Available: https://arxiv.org/abs/1505.04597

[31] "The area under an roc curve." [Online]. Available: http://gim.unmc.edu/dxtests/roc3.htm

[32] A. T. Feldman and D. Wolfe, "Tissue processing and hematoxylin and eosin staining," *Methods in Molecular Biology*, vol. 1180, pp. 31–43, 2014.

[33] N. Kumar, R. Gupta, and S. Gupta, "Whole slide imaging (wsi) in pathology: Current perspectives and future directions," pp. 1034–1040, 8 2020.

[34] "Kaggle: Your machine learning and data science community." [Online]. Available: https://www.kaggle.com/

[35] R. Phillips, W. Y. Shi, M. Deek, N. Radwan, S. J. Lim, E. S. Antonarakis, S. P. Rowe, A. E. Ross, M. A. Gorin, C. Deville, S. C. Greco, H. Wang, S. R. Denmeade, C. J. Paller, S. Dipasquale, T. L. Deweese, D. Y. Song, H. Wang, M. A. Carducci, K. J. Pienta, M. G. Pomper, A. P. Dicker, M. A. Eisenberger, A. A. Alizadeh, M. Diehn, and P. T. Tran, "Outcomes of observation vs stereotactic ablative radiation for oligometastatic prostate cancer: The oriole phase 2 randomized clinical trial," *JAMA Oncology*, vol. 6, pp. 650–659, 5 2020.

[36] "Prostate-specific antigen (psa) test - nci." [Online]. Available: https://www.cancer.gov/types/prostate/psa-fact-sheet

[37] F. Bianconi, A. Álvarez Larrán, and A. Fernández, "Discrimination between tumour epithelium and stroma via perception-based features," *Neurocomputing*, vol. 154, pp. 119–126, 4 2015.

[38] N. Linder, J. Konsti, R. Turkki, E. Rahtu, M. Lundin, S. Nordling, C. Haglund, T. Ahonen, M. Pietikäinen, and J. Lundin, "Identification of tumor epithelium and stroma in tissue microarrays using texture analysis," *Diagnostic Pathology*, vol. 7, p. 22, 3 2012. [Online]. Available: https://diagnosticpathology.biomedcentral.com/articles/10.1186/1746-1596-7-22

[39] D. Altunbay, C. Cigir, C. Sokmensuer, and C. Gunduz-Demir, "Color graphs for automated cancer diagnosis and grading," *IEEE Transactions on Biomedical Engineering*, vol. 57, pp. 665–674, 3 2010.

[40] W. Bulten, K. Kartasalo, P. H. C. Chen, P. Ström, H. Pinckaers, K. Nagpal, Y. Cai, D. F. Steiner, H. van Boven, R. Vink, C. H. van de Kaa, J. van der Laak, M. B. Amin, A. J. Evans, T. van der Kwast, R. Allan, P. A. Humphrey, H. Grönberg, H. Samaratunga, B. Delahunt, T. Tsuzuki, T. Häkkinen, L. Egevad, M. Demkin, S. Dane, F. Tan, M. Valkonen, G. S. Corrado, L. Peng, C. H. Mermel, P. Ruusuvuori, G. Litjens, M. Eklund, A. Brilhante, A. Çakır, X. Farré, K. Geronatsiou, V. Molinié, G. Pereira, P. Roy, G. Saile, P. G. Salles, E. Schaafsma, J. Tschui, J. Billoch-Lima, E. M. Pereira, M. Zhou, S. He, S. Song, Q. Sun, H. Yoshihara, T. Yamaguchi, K. Ono, T. Shen, J. Ji, A. Roussel, K. Zhou, T. Chai, N. Weng, D. Grechka, M. V. Shugaev, R. Kiminya, V. Kovalev, D. Voynov, V. Malyshev, E. Lapo, M. Campos, N. Ota, S. Yamaoka, Y. Fujimoto, K. Yoshioka, J. Juvonen, M. Tukiainen, A. Karlsson, R. Guo, C. L. Hsieh, I. Zubarev, H. S. Bukhar, W. Li, J. Li, W. Speier, C. Arnold, K. Kim, B. Bae, Y. W. Kim, H. S. Lee, and J. Park, "Artificial intelligence for diagnosis and gleason grading of prostate cancer: the panda challenge," *Nature Medicine*, vol. 28, pp. 154–163, 1 2022.

[41] B. Ehteshami Bejnordi, N. Timofeeva, I. Otte-Höller, N. Karssemeijer, and J. van der Laak, "Quantitative analysis of stain variability in histology slides and an algorithm for standardization," vol. 9041, 02 2014.

[42] N. Japkowicz and S. Stephen, "The class imbalance problem: A systematic study," *Intelligent Data Analysis*, vol. 6, pp. 429–449, 1 2002.

[43] X. Hu, L. Chu, J. Pei, W. Liu, and J. Bian, "Model complexity of deep learning: a survey," *Knowledge and Information Systems*, vol. 63, pp. 2585–2619, 10 2021.

[44] L. Duran-Lopez, J. P. Dominguez-Morales, A. F. Conde-Martin, S. Vicente-Diaz, and A. Linares-Barranco, "Prometeo: A cnn-based computer-aided diagnosis system for wsi prostate cancer detection," *IEEE Access*, vol. 8, pp. 128 613–128 628, 2020.

[45] K. Fan, S. Wen, and Z. Deng, "Deep learning for detecting breast cancer metastases on wsi," vol. 145. Springer Science and Business Media Deutschland GmbH, 2019, pp. 137–145.

[46] J. Ye, Y. Luo, C. Zhu, F. Liu, and Y. Zhang, "Breast cancer image classification on wsi with spatial correlations," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 1219–1223.

[47] K. Nagpal, D. Foote, Y. Liu, P. H. C. Chen, E. Wulczyn, F. Tan, N. Olson, J. L. Smith, A. Mohtashamian, J. H. Wren, G. S. Corrado, R. MacDonald, L. H. Peng, M. B. Amin, A. J. Evans, A. R. Sangoi, C. H. Mermel, J. D. Hipp, and M. C. Stumpe, "Development and validation of a deep learning algorithm for improving gleason scoring of prostate cancer," *npj Digital Medicine*, vol. 2, pp. 1–10, 12 2019.

[48] E. Reinhard, M. Ashikhmin, B. Gooch, and P. Shirley, "Color transfer between images," *IEEE Computer Graphics and Applications*, vol. 21, pp. 34–41, 9 2001.

[49] D. L. Ruderman, T. W. Cronin, and C.-C. Chiao, "Statistics of cone responses to natural images: implications for visual coding," *Journal of the Optical Society of America A*, vol. 15, p. 2036, 8 1998.

[50] D. Tellez, G. Litjens, P. Bándi, W. Bulten, J. M. Bokhorst, F. Ciompi, and J. van der Laak, "Quantifying the effects of data augmentation and stain color normalization in convolutional neural networks for computational pathology," *Medical Image Analysis*, vol. 58, p. 101544, 12 2019.

[51] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, 1986.

[52] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2015.

[53] A. Sharma and R. Rani, "A systematic review of applications of machine learning in cancer prediction and diagnosis," *Archives of Computational Methods in Engineering*, vol. 28, pp. 4875–4896, 12 2021.

[54] A. Shalimova, V. Babasieva, V. N. Chubarev, V. V. Tarasov, H. B. Schiöth, and J. Mwinyi, "Therapy response prediction in major depressive disorder: Current and novel genomic markers influencing pharmacokinetics and pharmacodynamics," pp. 485–503, 6 2021.

[55] Y. Lu and J. Han, "Cancer classification using gene expression data," *Information Systems*, vol. 28, pp. 243–268, 6 2003.

[56] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander, "Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring," *Science*, vol. 286, pp. 531–527, 10 1999. [Online]. Available: https://www.science.org/doi/10.1126/science.286.5439.531

[57] A. A. Alizadeh, M. B. Elsen, R. E. Davis, C. L. Ma, I. S. Lossos, A. Rosenwald, J. C. Boldrick, H. Sabet, T. Tran, X. Yu, J. I. Powell, L. Yang, G. E. Marü, T. Moore, J. Hudson, L. Lu, D. B. Lewis, R. Tibshirani, G. Sherlock, W. C. Chan, T. C. Greiner, D. D. Weisenburger, J. O. Armitage, R. Warnke, R. Levy, W. Wilson, M. R. Grever, J. C. Byrd, D. Botstein, P. O. Brown, and L. M. Staudt, "Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling," *Nature*, vol. 403, pp. 503–511, 2 2000.

[58] K. Balázs, L. Antal, G. Sáfrány, and K. Lumniczky, "Blood-derived biomarkers of diagnosis, prognosis and therapy response in prostate cancer patients," p. 296, 4 2021.

[59] L. J. V. Veer, H. Dai, M. J. V. de Vijver, Y. D. He, A. A. Hart, M. Mao, H. L. Peterse, K. V. D. Kooy, M. J. Marton, A. T. Witteveen, G. J. Schreiber, R. M. Kerkhoven, C. Roberts, P. S. Linsley, R. Bernards, and S. H. Friend, "Gene expression profiling predicts clinical outcome of breast cancer," *Nature*, vol. 415, pp. 530–536, 1 2002.

[60] H. Shimizu and K. I. Nakayama, "Artificial intelligence in oncology," *Cancer Science*, vol. 111, pp. 1452–1460, 5 2020. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1111/cas.14377

[61] Y. C. Chiu, H. I. Chen, A. Gorthi, M. Mostavi, S. Zheng, Y. Huang, and Y. Chen, "Deep learning of pharmacogenomics resources: Moving towards precision oncology," *Briefings in Bioinformatics*, vol. 21, pp. 2066–2083, 11 2020.

[62] A. D. Trister, "The tipping point for deep learning in oncology," pp. 1429–1430, 10 2019.

[63] F. Galati, V. Rizzo, R. M. Trimboli, E. Kripa, R. Maroncelli, and F. Pediconi, "Mri as a biomarker for breast cancer diagnosis and prognosis," *BJR|Open*, 5 2022. [Online]. Available: https://www.birpublications.org/doi/10.1259/bjro.20220002

[64] Z. Chen, X. Li, M. Yang, H. Zhang, and X. S. Xu, "Optimize deep learning models for prediction of gene mutations using unsupervised clustering," 3 2022. [Online]. Available: http://arxiv.org/abs/2204.01593

[65] J. S. Reis-Filho, F. Pareja, F. Derakhshan, D. N. Brown, J. Sue, P. Selenica, Y. K. Wang, A. D. C. Paula, M. Banerjee, Z. Ebrahimzadeh, M. Isava, M. Lee, R. Godrich, A. Casson, R. Padron, G. Shaikovski, A. van Eck, A. Marra, H. Dopeso, H. Y. Wen, E. Brogi, M. G. Hanna, C. Kanan, J. D. Kunz, F. C. Geyer, C. Leibowitz, D. Klimstra, L. Grady, and T. J.

Fuchs, "Abstract pd11-01: An artificial intelligence-based predictor of cdh1 biallelic mutations and invasive lobular carcinoma," *Cancer Research*, vol. 82, pp. PD11–01–PD11–01, 2 2022.

[66] S. Arslan, D. Mehrotra, J. Schmidt, A. Geraldes, S. Singhal, J. Hense, X. Li, C. Bass, J. N. Kather, and P. Raharja-Liu, "Deep learning can predict multi-omic biomarkers from routine pathology images: A systematic large-scale study," *bioRxiv*, p. 2022.01.21.477189, 4 2022.

[67] W. J. Catalona, D. S. Smith, T. L. Ratliff, K. M. Dodds, D. E. Coplen, J. J. Yuan, J. A. Petros, and G. L. Andriole, "Measurement of prostate-specific antigen in serum as a screening test for prostate cancer," *New England Journal of Medicine*, vol. 324, pp. 1156–1161, 4 1991.

[68] Z.-H. Zhou, "A brief introduction to weakly supervised learning," *National Science Review*, vol. 5, pp. 44–53, 1 2018. [Online]. Available: https://academic.oup.com/nsr/article/5/1/44/4093912

[69] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 12 2015.

[70] C. A. Ferreira, T. Melo, P. Sousa, M. I. Meyer, E. Shakibapour, P. Costa, and A. Campilho, "Classification of breast cancer histology images through transfer learning using a pre-trained inception resnet v2," vol. 10882 LNCS. Springer Verlag, 2018, pp. 763–770.

[71] K. Weiss, T. M. Khoshgoftaar, and D. D. Wang, "A survey of transfer learning," *Journal of Big Data*, vol. 3, p. 9, 12 2016. [Online]. Available: http://journalofbigdata.springeropen.com/articles/10.1186/s40537-016-0043-6

[72] "Imagenet." [Online]. Available: https://www.image-net.org/

[73] M. Ilse, J. M. Tomczak, and M. Welling, "Attention-based deep multiple instance learning," 2 2018.

[74] M. Y. Lu, D. F. Williamson, T. Y. Chen, R. J. Chen, M. Barbieri, and F. Mahmood, "Data-efficient and weakly supervised computational pathology on whole-slide images," *Nature Biomedical Engineering*, vol. 5, pp. 555–570, 6 2021.

[75] D. Bank, N. Koenigstein, and R. Giryes, "Autoencoders," 3 2020. [Online]. Available: http://arxiv.org/abs/2003.05991

[76] M. Macenko, M. Niethammer, J. S. Marron, D. Borland, J. T. Woosley, X. Guan, C. Schmitt, and N. E. Thomas, "A method for normalizing histology slides for quantitative analysis," 2009, pp. 1107–1110.

[77] Z. Li, J. Zhang, T. Tan, X. Teng, X. Sun, H. Zhao, L. Liu, Y. Xiao, B. Lee, Y. Li, Q. Zhang, S. Sun, Y. Zheng, J. Yan, N. Li, Y. Hong, J. Ko, H. Jung, Y. Liu, Y. C. Chen, C. W. Wang, V. Yurovskiy, P. Maevskikh, V. Khanagha, Y. Jiang, L. Yu, Z. Liu, D. Li, P. J. Schuffler, Q. Yu, H. Chen, Y. Tang, and G. Litjens, "Deep learning methods for lung cancer segmentation in whole-slide histopathology images - the acdc@lunghp challenge 2019," *IEEE Journal of Biomedical and Health Informatics*, vol. 25, pp. 429–440, 2 2021.