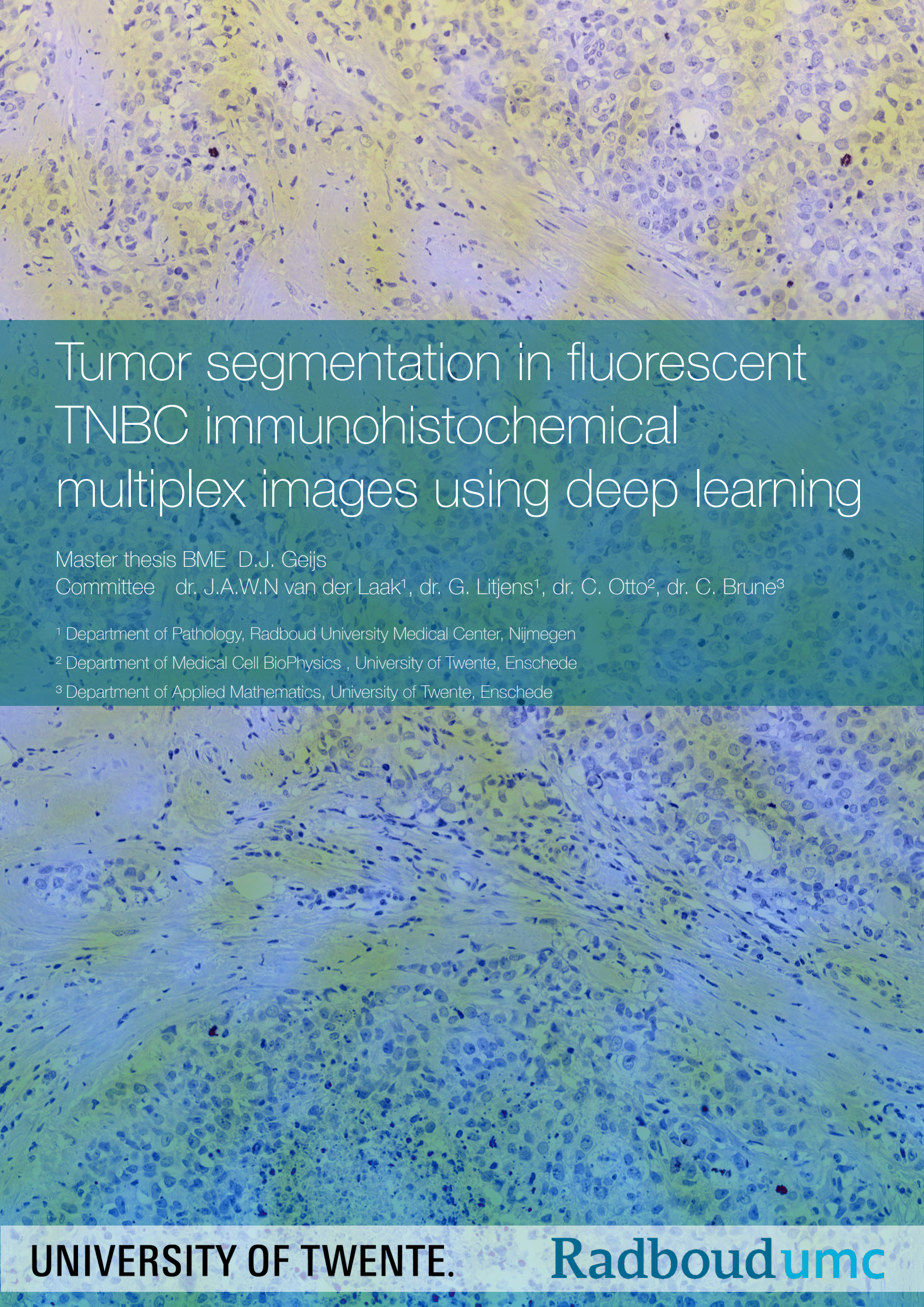




Tumor segmentation in fluorescent TNBC immunohistochemical multiplex images using deep learning

Master thesis BME D.J. Geijs

Committee dr. J.A.W.N van der Laak¹, dr. G. Litjens¹, dr. C. Otto², dr. C. Brune³

¹ Department of Pathology, Radboud University Medical Center, Nijmegen

² Department of Medical Cell BioPhysics, University of Twente, Enschede

³ Department of Applied Mathematics, University of Twente, Enschede

Tumor segmentation in fluorescent TNBC immunohistochemical multiplex images using deep learning

MSc Thesis (*Afstudeerscriptie*)

written by

by Daan Jan Geijs

(born in)

under the supervision of **Geert Litjens** and **Cees Otto**, and submitted to the Board of Examiners in partial fulfillment of the requirements for the degree of

MSc in Biomedical Engineering

at the *University of Twente*.

Date of the public defense: **Members of the Thesis Committee:**
July 12, 2019

Jeroen van der Laak
Christoph Brune

Abstract

Breast cancer is among females the most frequently diagnosed cancer and the leading cause of cancer death. A subtype of breast cancer called triple negative breast cancer (TNBC) is known to be more aggressive, generally occur at younger age and even very small ($<1\text{cm}$) node-negative TNBC show recurrence within 5 years in 15% of cases if left untreated. For TNBC, several studies showed that the number of tumor-infiltrating lymphocytes (TIL) in hematoxylin and eosin (H&E) stained sections strongly correlates with disease free survival. Subtyping of lymphocytes could strongly help finding more powerful prognostic markers. However, standard H&E stained sections do not permit specific subtyping of lymphocytes and immunohistochemistry (IHC) allows very limited subtyping. New scanning systems, staining protocols and medical imaging analysis algorithms allows to gather spatial information of lymphocyte subtypes and to determine subtype positioning of lymphocytes in peri- or intertumoral regions. A first step towards this goal is to detect tumor regions to determine whether a lymphocyte is positioned peri- or intertumoral.

Therefore, the aim of this thesis was to investigate the performance of convolutional networks to segment tumor regions in TNBC whole-slide multiplex IHC slides. Multiple experiments were conducted to investigate and maximize the performance. The data used for training was investigated and it was concluded that training a FCNN (fully convolutional neural network) using the DAPI and CK8-18 data channels together with a resolution of $0.96\text{ }\mu\text{m}/\text{pix}$ and patch size of 128×128 resulted in the highest segmentation performance. Enriching the dataset with hard mining had no positive effects on the performance. Using the different architecture U-net resulted in similar results compared to that of a FCNN. A 'model averaging ensemble' resulted in the highest segmentation performance with a F1 score of 0.83. It can be concluded that fully convolutional networks were able to segment tumor regions in triple negative and holds true for both FCNN and U-net architectures and can be used for the overarching aim of this research, namely extracting powerful prognostic information from intra- and peritumoral lymphocytes.

Contents

1	Introduction	3
2	Background	6
2.1	Breast cancer in pathology	6
2.2	Tumor-infiltrating lymphocytes	7
2.3	Multiplex immunohistochemistry	9
2.3.1	Tyramide signal amplification	9
2.3.2	Spectral unmixing	10
2.4	Deep learning	11
2.4.1	Neural Networks	11
2.4.2	Convolutional Neural Networks	14
2.4.3	Semantic segmentation	15
3	Methods	19
3.1	Acquisition and pre-processing	19
3.1.1	Material	19
3.1.2	Preparation and staining	19
3.1.3	Scanning instrument	20
3.1.4	Scanning and unmixing	20
3.1.5	Conversion to multichannel whole-slide images	20
3.1.6	Multichannel whole-slide image to brightfield whole-slide-image	22
3.2	Annotations	24
3.2.1	Set 1	24
3.2.2	Set 2	25
3.3	Training, validation and testing	26
3.3.1	Minibatch preparation	26
3.3.2	Training FCCN	26
3.3.3	Generation cancer likelihood maps	27
3.3.4	Performance	27
3.4	Experiments	29
3.4.1	Channels	30
3.4.2	Context vs spatial resolution	30
3.4.3	Hard mining	31
3.4.4	U-net	32
3.4.5	Ensemble	33

4	Results	35
4.1	Experiment 1: Channels	35
4.2	Experiment 2: Context vs spatial resolution	37
4.3	Experiment 3: Hard mining	37
4.4	Experiment 4: U-net	38
4.5	Experiment 5: Ensemble	39
5	Discussion	41
6	Conclusion	45

1. Introduction

Among females, breast cancer is the most commonly diagnosed cancer and the leading cause of cancer death. In 2018, worldwide 2.1 million newly diagnosed female breast cancer cases were diagnosed, accounting for almost 1 in 4 cancer cases among women (Bray et al., 2018). Diagnosis is done by using the most widely used staging system among clinicians, the TNM system. This system codes the extent of the primary tumor (T), regional lymph nodes (N), and distant metastases (M) and provides a “stage grouping” based on T, N, and M (Edge and Compton, 2010). The TNM staging system correlates all the T, N and M tumor characteristics with survival data to help estimate outcomes. It is based upon a retrospective analysis of survival in diverse samples of patients representing all stages of disease. It reflects the clinical evaluation methods and treatments that are applied to the particular study population. While an individual patient’s clinical course and outcome cannot be predicted with certainty, available survival data can help direct treatment decisions and provide an estimate of the likely prognosis (Hayes, 2010).

In case of breast cancer, apart TNM staging the tumor is also graded according to the modified Bloom-Richardson method. This method, specifically for breast cancer, is used to score an invasive breast tumor into three grades ranging 1 to 3 and consists of three parts of tumor morphology: the rate of tubular differentiation, nuclear pleomorphism and the mitotic activity defined as number of mitosis per mm². Each of these three categories is labeled with a score from 1 to 3 and the total score can be calculated by summing the labels of the three categories. This can give a minimal score of 1 and a maximum score of 9. The histological grade is I for the scores 3-5, II for 6-7 and III for 8-9 (Singletary et al., 2002).

Besides histological grading and TNM-staging, more recently immunohistochemical (IHC) grading of the expression of oestrogen receptor (ER), progesterone receptor (PR) and human epidermal growth factor receptor 2 (HER2) is performed. Different combinations of the expression of these receptors lead to different treatments and therefore classifying patients using these markers add to better individual treatment decisions. Approximately 20% of all breast cancers are ER-/PR-/HER2- and are therefore referred to as ‘triple negative breast cancer’ (TNBC) (Brouckaert et al., 2012). TNBC tumors are known to be more aggressive, generally occur at younger age and reduce therapy options because of lacking active receptor sites. In 15% of cases very small (<1cm) node-negative (cancer that has not spread to the lymph nodes) TNBC show recurrence within 5 years if left untreated. In contrast, hormone receptor positive (HR+; i.e. ER+ and/or PR+) breast cancer recurrence is only 6% (Berry et al., 2006; Dent et al., 2007). Therefore, in practice consideration of adjuvant chemotherapy is recommended for TNBC areas exceeding 0.5 cm. However, chemotherapy after resection of limited stage TNBC results in significant over-treatment. Over-treatment is associated with high morbidity, mortality and cost for society. Reversed, abstaining

from adjuvant chemotherapy, results in under-treatment of a large number of women (Katz and Morrow, 2013). Unfortunately, no biomarkers currently exist for assessing the risk of TNBC recurrence after resection. Availability of such a biomarker would enable selection of patients at increased risk, requiring adjuvant chemotherapy.

Recently it was found that for different types of malignancies powerful prognostic information can be derived from the immune infiltrate in histological sections. Increased numbers of tumor-infiltrating lymphocytes (TILs) are associated with better prognosis, probably because of a stronger and more effective host antitumor immune response. Specifically for TNBC, several studies showed that the number of intra- and peritumoral lymphocytes in hematoxylin and eosin (H&E) stained sections strongly correlates with disease free survival (DFS) and distant metastasis-free survival. So far, no results were published on quantification of specific subsets of lymphocytes for TNBC (Mahmoud et al., 2011).

However promising these biomarkers and techniques may be, a number of challenges still have to be addressed before they can be introduced in clinical practice. First of all, standard H&E stained sections do not permit specific subtyping of lymphocytes (Mahmoud et al., 2011). With the use of immunohistochemistry (IHC), subtyping is possible by targeting specific antibodies. A standard brightfield IHC allows to localize and identify one subtype of lymphocytes. Therefore, subtyping different lymphocytes in a tumor requires large amount of tissue sections. Increasing the amount of stains on a single IHC is problematic due to the crosstalk between the different targeted antibodies.

Secondly, human visual assessment was shown to lack reproducibility, having kappa values as low as 0.45 (Adams et al., 2014). In general, visual counting of IHC positive cells is too time-consuming and is replaced by semi-quantitative 'eyeballing' which is far less accurate. To date, no study attempted to identify which methodology yields the strongest, most reproducible prognostic information from TIL.

Thanks to the advent of whole-slide scanning systems, pathologists can perform their diagnoses on a computer screen instead of using a microscope. Recent commercial scanning systems allows visualization of multiple targeted antibodies on a single slide. Therefore, IHC slides are stained using tyramide signal amplification (TSA) allowing to stain tissue up to seven unique fluorophores. Next to the increasing variety of whole-slide scanning systems, the digital systems allowed the development of computer-aided diagnostic (CAD) tools (Litjens et al., 2017). Mainly the wide variations in pathology and the potential fatigue of human experts led to researchers and doctors beginning to benefit from computer-assisted interventions (Shen et al., 2017).

Although at first the rate of progress in computational medical image analysis was not as rapid as that in medical imaging technologies but improved with the introduction of all kinds of machine learning techniques. Conventionally, features were designed mostly by human experts on the basis of their knowledge about the subject. For nonexperts it was therefore challenging to apply machine learning techniques for their own studies. Deep learning has overcome this obstacle by incorporating the feature engineering step into a learning step, enabling hierarchical feature representations to be learned from data. Deep learning has achieved record-breaking performance in a variety of artificial intelligence applications and grand challenges (Shen et al., 2017).

Current state-of-art are fully-convolutional neural networks (FCNN) first applied by Long et al. (2015) and all kind of variants such as U-net (Ronneberger et al., 2015) are a popular choice for analyzing medical images (Litjens et al., 2017). Moeskops et al. (2016) trained a single fCNN to segment brain MRI images, arteries in CT angiography and muscle tissue in breast MRI. Next to tissue segmentation, Cireşan et al. (2013)

used CNN to detect single mitotic cells in histologic breast cancer images. However, no studies report the use of fully-convolutional networks to segment whole-slide multiplex IHC slides yet.

The new scanning systems, staining protocols and current medical imaging analysis algorithms allows to overcome the challenges that withhold gathering reproducible prognostic information from TIL. Current deep learning architectures achieve fully segmentation of medical images at high accuracy's. The possibility of full segmentation allows to objectively extract spatial information about the TILs with respect to the tumor tissue. Segmenting tumor is the first step working towards counting the number of lymphocytes in intratumoral and peritumoral regions for reproducible prognostic information from TIL. Therefore, the aim of this thesis is to use and to investigate convolutional networks to segment tumor regions in triple negative breast cancer whole-slide multiplex IHC slides.

2. Background

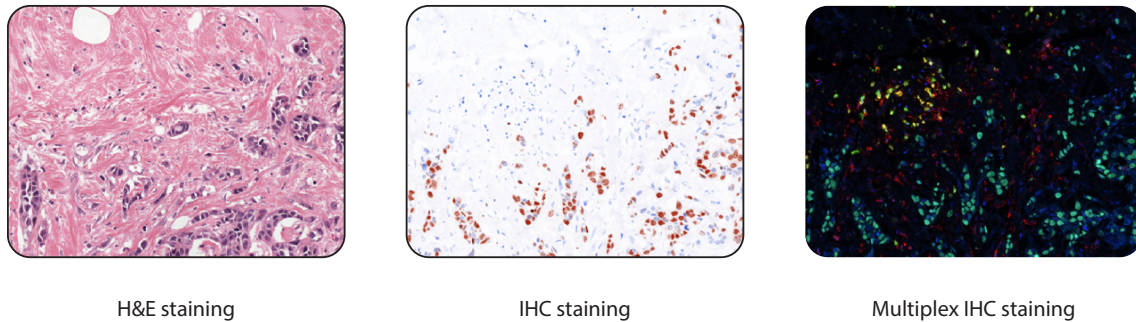


Figure 2.1: Example of a metastatic region in triple negative breast tissue in (A) a histochemistry hematoxylin and eosin staining (H&E), (B) immunohistochemistry Ki67 staining and (C) multispectral immunohistochemistry staining.

The goal of this thesis is to segment tumor in whole-slide multiplex IHC slides. To reach the goal of this thesis a whole cascade of steps was involved to achieve results. First of all this thesis used non-conventional scanning systems and staining protocols for histopathology and to detect tumor regions in the acquired data deep learning was used. Therefore, background information about these topics will be given in this chapter starting with a global overview of the normal workflow in histopathology. Because the staining and acquisition deviates from the normal workflow, the second topic will explain the background information about the staining procedure and the acquisition of the images more carefully. Since this thesis uses data that contains TILs the third part contains information about relevant lymphocyte markers used in this project. The last part will give a concise introduction to neural networks focussing on the techniques used in this thesis, namely convolutional neural networks and semantic segmentation.

2.1 Breast cancer in pathology

Breast cancer evolves through clinical and pathological stages starting with normal epithelial proliferation to invasive carcinoma and carcinoma in-situ. Treatment depends on the stage of a tumor and therefore careful examination of breast tissue sections is performed under a microscope by a pathologist. The procedure for preparing tissue slides consists of multiple steps before it is examined by a pathologist. (i) Collecting the tissue removed from the body for diagnosis. (ii) Applying a fixative to degenerate and preserve the tissue. (iii) Embed the tissue in paraffin. (iv) Cutting a thin (3-5 μm) slide from the paraffin-embedded tissue section and mounting it on glass slides. (v) Treating

the tissue with multiple contrasting stains to highlight different tissue structures and cellular features.

This last step is one of the most important steps in the tissue preparation cascade. Cells are colorless and transparent and therefore histological sections have to be stained to highlight features of importance. Microscopical evaluation can only be done if a tissue slide is stained and provides valuable information to pathologists for diagnosing and characterizing pathological conditions. Hematoxylin and eosin (H&E) staining is the golden standard in histopathology. Cell nuclei are stained by hematoxylin with a blue color and the counter-stain eosin stains cytoplasmic and non-nuclear components in different shades of pink (Figure 2.1.A). Most TNM staging and histopathological grading for breast cancer is done on tissue stained with this technique. However, H&E is a non-specific stain, meaning that it stains most of the cell types in the same way. This makes it sometimes hard for a pathologist to do proper TNM staging or grading. In this case it is common to do another staining called immunohistochemistry (IHC).

IHC makes it possible to visualize particular properties of cell types, such as chemical groupings or molecules within cells or tissue (Figure 2.1.B). This enables pathologists to stain specific types of cells such as subtypes of lymphocytes. The downside of IHC is that the amount of specific components that can be stained are limited. This is due to the crosstalk between antibody's or bad selectivity of chemical agents. For every marker a new slide has to be prepared and stained, being laborious and time-consuming.

For daily practice H&E and IHC provides enough information for a pathologist to do TNM staging and histopathological grading. However, it should be noted that these staging and grading models are based on large retrospective studies that reflects the clinical evaluation methods and treatments that are applied to the particular study population. This means that the current TNM staging method is build on information from IHC and H&E evaluations of pathologist. Therefore, periodically the TNM method is updated with the latest scientific insights (Edge and Compton, 2010).

A relatively new technique called multiplex immunohistochemistry enables to stain up to 7 immunochemical markers using fluorescent dyes (Figure 2.1.C). Although, this technique is neither used on a daily basis in pathological assessments nor part of the TNM staging method, it holds invaluable information about the distribution and localization of specific cell types and multiple expressed antigens and therefore the choice of this work. This scientific information is of huge importance for the further development of the TNM staging system.

Table 2.1: Overview of different T cells and the antibodies they express.

T Cell	CD3	CD4	CD8	CD45RO	FOXP3
Cytotoxic	+	-	+	-	-
Memory	+	+	+	+	-
Regulatory	+	+	-	-	+
Helper	+	+	-	-	-

2.2 Tumor-infiltrating lymphocytes

Histopathological analyses of human tumors have provided evidence that variable numbers of infiltrating immune cells are found in different tumors of the same type and are found in different locations within and around a tumor. One type of infiltrating immune cells are lymphocytes and are located in specific areas. T cells are a subgroup

of lymphocytes and are known to be located in the invasive margin or tumor core, therefore also called tumor-infiltrating lymphocytes (TIL). The fact that populations of T cells are located in different areas of a tumor suggests that different immune cell populations may have different roles in tumor defense. Moreover, the variable density and location of these immune cells between tumors in different individuals with the same cancer type prompted the investigation of whether the immune contexture might affect clinical outcome (Fridman et al., 2012).

T cells can be distinguished by targeting a set of antigens that are available on their surface, called the CD-markers. The combination of CD-markers present characterizes the type of T cell. Table 2.1 gives an overview of T cells and their corresponding CD-markers. First of all CD3+CD4+ or CD3+CD8+ cells can be naive T cells that are able to differentiate into effector T cells, namely CD8+ cytotoxic T cells or CD4+ t helper cells. Therefore these naive T cells have to be stimulated by a combination of antigens, co-stimulatory molecules and cytokines. Important are CD3+CD8+ cytotoxic T cells that contain perforin and granzymes, which are released on interaction with target cells expressing cognate antigen. This leads to the death of target cells by apoptosis (Fridman et al., 2012). A population of the CD3+CD4+ T cells are helper T cells that secrete cytokines which trigger multiple cascades. In this way they can act on cytotoxic T cells, immune cells and epithelial and endothelial cells. Another population of the CD3+CD4+ T cells are regulatory T cells that can inhibit T cells and therefore have a central role in suppressing anti-self immune responses. Memory T cells are CD3+CD4+CD45RO+ or CD3+CD8+CD45RO+ cells that have encountered antigen and that respond faster and with increased intensity on antigenic stimulation compared with naive T cells (Fridman et al., 2012).

A strong lymphomatic infiltration has been reported to be associated with good clinical outcome in many different tumor types including melanoma, head and neck, breast, bladder, urothelial, ovarian, colorectal, renal, prostatic and lung cancer. Therefore, high densities of CD3+ T cells, CD8+ cytotoxic T cells and CD45RO+ memory T cells were clearly associated with a longer disease-free survival and overall survival (Macchia et al., 2013).

In contrast to the effects of cytotoxic T cells and memory T cells, analysis of the effect of CD4+ T cell populations on clinical outcome has resulted in apparent contradictory results. Therefore, their effects have been a matter of debate for the past decade. The case of regulatory T cells is a striking example of conflicting data that lead to difficult interpretation (Fridman et al., 2012). The lack of an unambiguous molecular definition of regulatory T cells has severely hampered efforts to experimentally address the developmental processes that generate these cells and their functional role within the immune system (Fontenot et al., 2005). It was shown that the transcription factor forkhead box protein P3 (FOXP3) is a key regulatory gene for the development of regulatory T cells and made it possible to make a better distinction between CD4+ populations (Fontenot et al., 2005; Hori et al., 2003). This led to more correlating research that the high infiltration of regulatory T cells has been correlated with poor overall survival in breast cancer (Fridman et al., 2012).

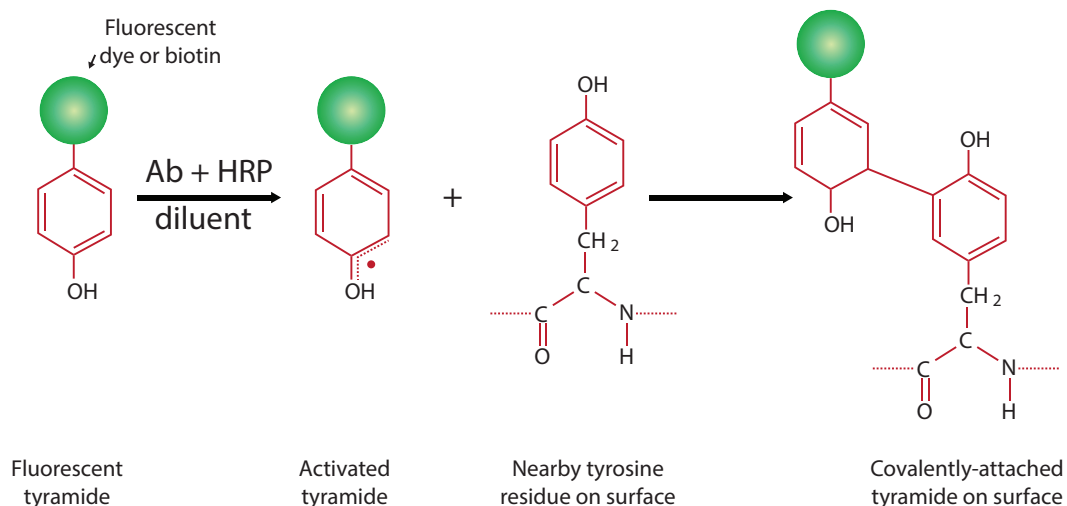


Figure 2.2: Tyramide signal amplification reaction. Horseradish peroxidase catalyses tyramide to form a radical. The tyramide radical reacts with nearby tyrosine residue on the tissue surface and forms a covalent bond.

2.3 Multiplex immunohistochemistry

To make it possible to label different CD-markers for lymphocytes and antigens related to tumor tissue, a staining technique called tyramide signal amplification was used. The background of this staining technique will be explained first and secondly the acquisition technique needed for this staining method will be explained.

2.3.1 Tyramide signal amplification

Tyramide signal amplification (TSA) labels antigens by enzyme-catalyzed deposition of probe-conjugated tyramide molecules. Tyramide can be conjugated to probes such as fluorescent labels or biotin. The enzyme horseradish peroxidase (HRP) is conjugated to an antibody which binds to a specific antigen. The HRP catalyzes the formation of reactive tyramide radicals that covalently bind to tyrosine on the tissue (Figure 2.2). Tyramide radicals are unstable and the half-life time is short, which restricts the binding of tyramide to tyrosine in close proximity of the antibody (Dixon et al., 2015). TSA increases the proportion of total signal derived from specific antibody binding events relative to autofluorescence. This amplification is caused by the fact that tyrosine moieties are more present than target antigen sites (Clutter et al., 2010).

A sequential multiplexing technology developed by Perkin Elmer uses multiple probe-conjugated tyramides to identify up to seven antigens on tissue. Each round of antigen labeling, the primary and secondary antibody complex used for binding the HRP are removed from the tissue. While the antibodies are removed the covalently attached tyramides stay attached to the tyrosine moieties on the tissue. This allows to repeat the labeling process now with another HRP conjugated antibody. This indirect way of antigen labeling also prevents crosstalk between primary antibodies from the same species (Dixon et al., 2015).

2.3.2 Spectral unmixing

Spectral unmixing decomposes one or more images that include contributions from multiple spectral sources into a set of component images (the “unmixed images”) that correspond to contributions from each of the spectral entities within the sample (Levenson et al., 2009). If the sample includes three different fluorescent dyes, each specific to a particular structural entity, then each unmixed image reflects contributions principally from only one of the dyes.

The unmixing procedure requires spectral eigenstates are known beforehand. Pure spectra are measured from which spectral eigenstates are derived. Once eigenstates have been identified, an image can be decomposed by calculating a coefficient matrix, that corresponds to the relative weighting of each of the eigenstates in the overall image. The contributions of each of the individual eigenstates can be separated out to yield the unmixed image set (Levenson et al., 2009).

For example and the sake of simplicity, an sample may contain spectral contributions from two different spectral sources $F(\lambda_k)$ and $G(\lambda_k)$. The net signal at any coordinate and at a particular wavelength is the sum of the two contributions, weighted by the relative abundance of each (Levenson et al., 2009). This can be expressed as

$$I(\lambda_k) = aF(\lambda_k) + bG(\lambda_k) \quad (2.1)$$

where λ_k is used to denote a given wavelength. The functions F and G can be termed the spectral eigenstates for the system, belonging to the pure spectra of the spectral source in the sample (Levenson et al., 2014). The abundance of the spectral contributions from the two sources are described by a and b . If $F(\lambda_k)$ and $G(\lambda_k)$ are known, To solve a and b , Equation (2.1) can be inverted and rewritten as

$$A = E^{-1}I \quad (2.2)$$

where A is a column vector with components a and b , and E is a matrix whose column are the spectral eigenstates, in this case $[FG]$ (Levenson et al., 2014).

Using Equation (2.2), measured spectral images of a sample can be used to calculate contributions to the images arising purely from source F and purely from source G at particular pixel locations, which can be repeated for each pixel location (Levenson et al., 2014). In the above example the number of spectral sources is two ($m = 2$ with F and G), however, the unmixing technique is not restricted to any particular number of sources (Levenson et al., 2014). For example if the number of wavelengths at which data is collected is n , then the matrix E is an $n \times m$ matrix instead of $n \times 2$. The unmixing algorithm can then be employed in the same manner as described above to isolate specific contributions at each pixel location in an image from each of the m spectral eigenstates. However, one factor can limit the ability of the algorithm and that is the correlation between two spectra and their eigenstates. The correlation between two spectra, such as two spectral eigenstates I_1 and I_2 can be described by a spectral angle θ where

$$\theta = \cos^{-1} \left[\frac{I_1 \cdot I_2}{|I_1||I_2|} \right] \quad (2.3)$$

Sets of spectra for which θ is small for two members are not as easily separated into their components. Physically, the reason for this is easily understood: if two spectra are only marginally different, it is harder to determine the relative abundance of each. In most cases manufacturers sell kits that use a optimized set of fluorescent labels having a large θ (Levenson et al., 2009).

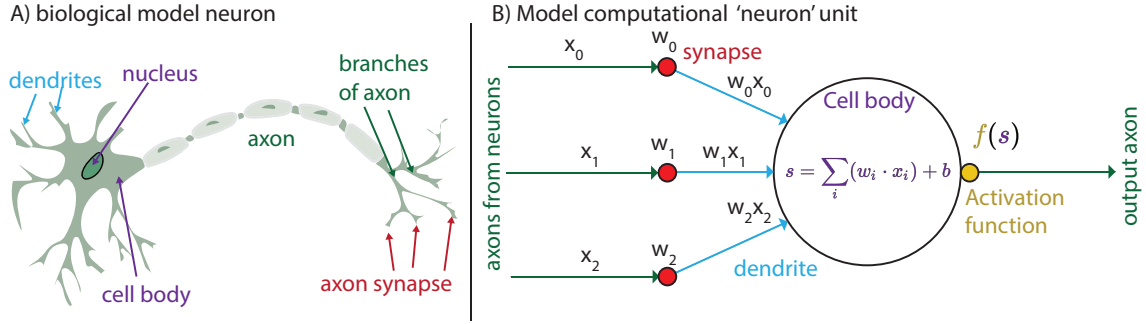


Figure 2.3: A cartoon drawing of a biological neuron (A) and its mathematical model (B). The parts in the model representing parts of the biological neuron are sharing the same color. Image derived from [Karpathy \(2018\)](#).

2.4 Deep learning

2.4.1 Neural Networks

Neural networks are adaptive statistical models based on an analogy with the structure of the brain as can be seen in Figure 2.3. They are adaptive in that they can learn to estimate the parameters of some population using a small number of samples (one or a few) at a time ([Abdi et al., 1999](#)). They are built from simple units, sometimes called neurons by analogy. These units are interlinked by a set of weighted connections. Learning is usually accomplished by changing of the connection weights. Single or multiple units can correspond to features of a pattern that we want to analyze or that we want to use as a predictor. Within neural systems it is useful to distinguish three types of units: input units which receive data from outside the neural network, output units which send data out of the neural network, and hidden units whose input and output signals remain within the neural network. The standard layer type for these hidden layers is the fully connected layer. Here, the outputs of the neurons in one layer are connected pairwise to all neurons in the subsequent layer, thus propagating information through the layers ([Krose and Van der Smagt, 1996](#); [Abdi et al., 1999](#)). An example of such a network can be seen in Figure 2.5.

The output of an artificial neuron is determined by its inputs, which are individually weighted and then summed as the 'nett' input signal s . The output or by analogy called 'firing' of the artificial neuron is regulated by an activation function. This is computed by first calculating s by taking the dot product of its weights w and inputs x combined with an added bias b , after which it applies an activation function $f(s)$, resulting in an output y .

$$y = f \left(\sum_i (w_i \cdot x_i) + b \right) \quad (2.4)$$

Multiple activation functions f can be used in neural networks. An example of an activation function is the sigmoid function. The sigmoid is a non-linear function with the form

$$f(s) = \frac{1}{1 + e^{-s}} \quad (2.5)$$

and is shown in Figure 2.4. The sigmoid functions allows real numbers to be 'converted' into a range of 0 and 1. Where large negative numbers tend to go to zero, and large positive numbers to 1. This can be interpreted as the firing rate of a neuron being

strongly inhibited and therefore not firing at all to fully-saturated firing (Karpathy, 2018). An alternative activation function called the Rectified Linear Unit (ReLU) has become very popular in the last few years. It can be seen in Figure 2.4 and can be expressed as $f(x) = \max(0, x)$ or mathematically as

$$f(s) = \begin{cases} 0 & \text{if } s < 0 \\ s & \text{if } s \geq 0 \end{cases} \quad (2.6)$$

doing a non-linear job by thresholding the activation of a neuron at zero.

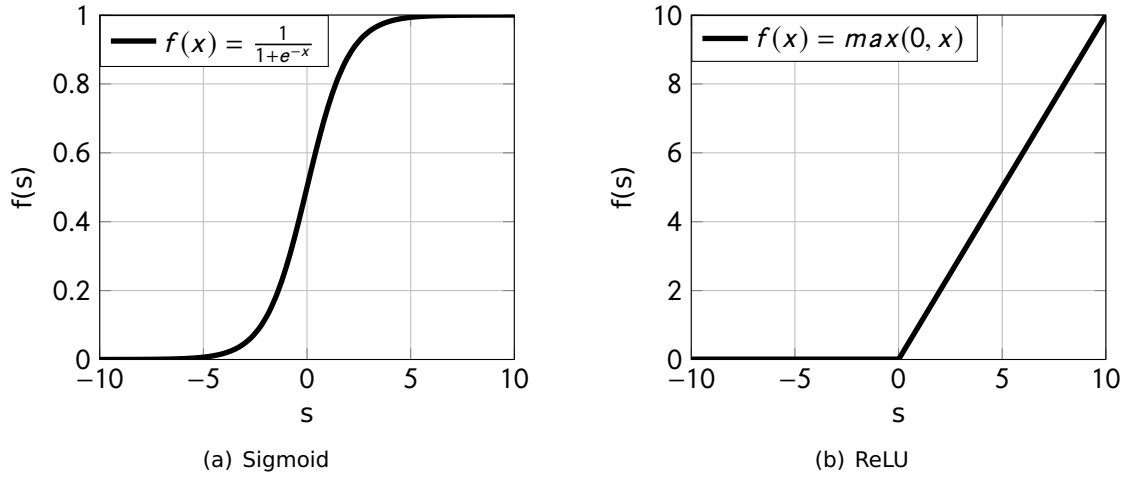


Figure 2.4: (a) Sigmoid non-linearity 'converts' real numbers between $[0,1]$. (b) Rectified Linear Unit (ReLU) activation function, which is zero when $x < 0$ and then linear with slope 1 when $x \geq 0$.

So in summary, each unit performs a relatively simple job: receive input from neighbours or external sources and processes this to an output signal which is propagated to other units. Apart from this processing, a second task is the adjustment of the weights. The weights of the neurons in each layer can be tuned to learn complex interactions between inputs. At the start of the training the weights are randomly chosen and iteratively the weights of the model are adjusted and is called training. In the first step of the training, during forward propagation, the input of the network is processed through the network by computing the activation of each neuron, resulting in an output. The second step of training is based on the output of the network combined with the ground truth or label, which a loss function is computed. The loss function estimates how wrong the model is in terms of its ability to estimate the relationship between the input and the output. It can be thought of a difference or distance between the predicted value and the actual value. An example of a commonly used function is the cross-entropy loss function:

$$H(y, \hat{y}) = - \sum_i (\hat{y}_i \cdot \log(y_i)) \quad (2.7)$$

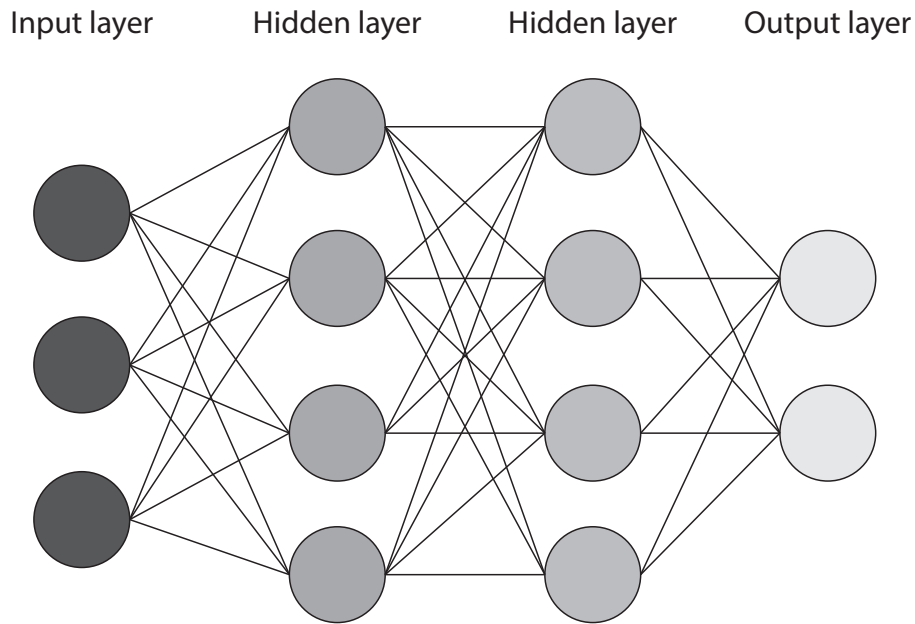


Figure 2.5: A neural network with two hidden layers. Between subsequent layers, all neurons are connected to each other. This network can learn a function with three inputs that produces two outputs.

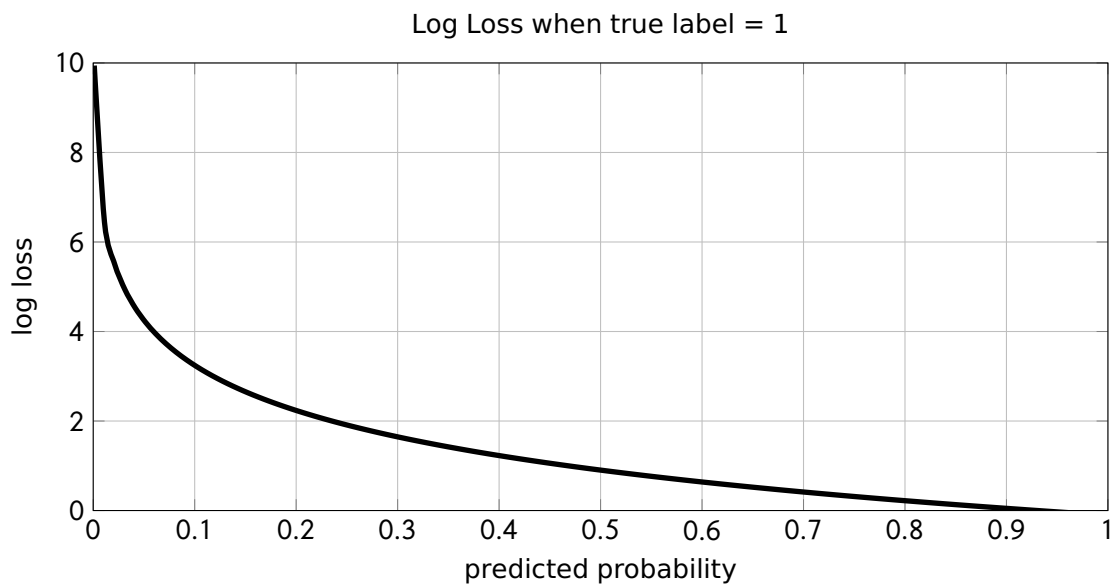


Figure 2.6:
Log Loss graph shows the range of possible loss values given a true observation.

Figure 2.6 shows the range of possible loss values given a true label. When the predicted probability of the network approaches 1 the decrease in loss reduces. However, when the network outputs a low predicted probability the log loss increases rapidly, meaning that the cross-entropy loss penalizes predictions that are confident and wrong.

The last step of the training iteration is to update the weights and is called backpropagation. Because all functions used in the neural network are differentiable, we can calculate the gradient of each weight with respect to the loss. We can thus update each

weight to reduce the loss, which in turn increases the similarity between the network output and the true label, bringing us closer to our goal. The objective of a neural network is to find weights or a structure to minimize the loss and is achieved by showing the network millions of inputs. In this way the network slowly converges and 'learns' the desired function. The way the network converges or updates depends on the method chosen to update the parameters. A popular method is a momentum update, which adds an extra parameters that slows down the learning process the further the network converges. A downside of the fully connected layer is that the amount of weights that need to be learned quickly rises when increasing the size of the network, making it computationally expensive, the next section describes how this problem can be reduced.

2.4.2 Convolutional Neural Networks

In the late nineties deep convolution neural networks (CNNs) showed promising and spectacular advances in deep learning. Despite these successes, CNNs were largely disregarded by the machine-learning community and their widespread breakthrough had to wait a decade longer when a CNN was used to beat state-of-the-art in the Imagenet image classification challenge (Krizhevsky et al., 2012). This success came from the efficient use of GPUs, ReLUs, techniques to generate more training examples by deforming the existing ones (data augmentation) and a new technique called dropout where neurons in the network randomly are disabled to prevent the network from saturating (LeCun et al., 2015; Srivastava et al., 2014).

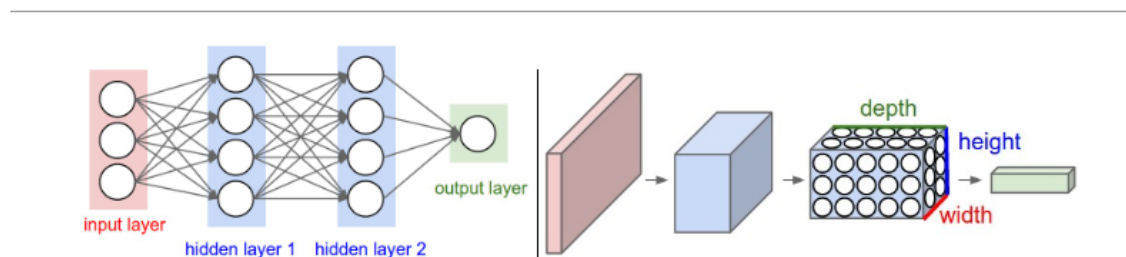


Figure 2.7: A neural network with two hidden layers. Between subsequent layers, all neurons are connected to each other. This network can learn a function with three inputs that produces two outputs. Image from Karpathy (2018).

Figure 2.7 provides an example of a simple CNN layout compared to a neural network. The CNN is still a neural network having multiple layers with an input layer and output layer. However, the way this layers are structured differ from the fully connected neural networks. The convolutional layer is similar to the fully connected layer in that it has learnable weights. Instead of passing a single neuron activation to all neurons downstream a convolutional layer slides kernels across the input. The CNN reduces the amount of weights per layer, by using a so-called kernel. This kernels form the hidden layers and are composed of three-dimensional volumes (Dumoulin and Visin, 2016; Karpathy, 2018). This means that every layer of a CNN transforms the 3D input volume to a 3D output volume of neuron activations. The dot product of the weights of the kernel and the overlapping values of the input image result in a feature map. The output of a layer is called a feature map. In other words, neurons in a convolutional layer are organized in feature maps, within which each neuron is connected to local patches in the feature maps of the previous layer through a set of weights in the kernel, which is often

referred to as a 'filter bank' (LeCun et al., 2015). Back-propagation for a convolutional layer means that the weights of the kernel are updated to produce features that carry sensible information to solve a task.

To get a better understanding how the kernel interacts with a previous layer a step by step explanation is provided in Figure 2.9. To keep the explanation simple a single input feature map is drawn, while in reality this can be a whole stack of maps on which the kernel works. The kernel slides across the feature map and the product between each element of the kernel and the feature map it overlaps is computed. If there are multiple input feature maps, the kernel will have to be 3-dimensional or, equivalently each one of the feature maps will be convolved with a distinct kernel – and the resulting feature maps will be summed up element-wise to produce the output feature map (Dumoulin and Visin, 2016).

The white grid is called the input feature map and the light blue grid the kernel. The normal dot product operation in Figure 2.9.A results in a light green 3x3 output feature map, while the input was a 5x5 feature map. The input feature map size can be increased by adding zero's. This is called zero-padding, and helps preserving the size of the input (Figure 2.9.B). Another method is striding which is a form of subsampling. The kernel skips a value in the feature map when performing the dot product operation (Figure 2.9.C). Both methods can also be combined as can be seen in Figure 2.9.D

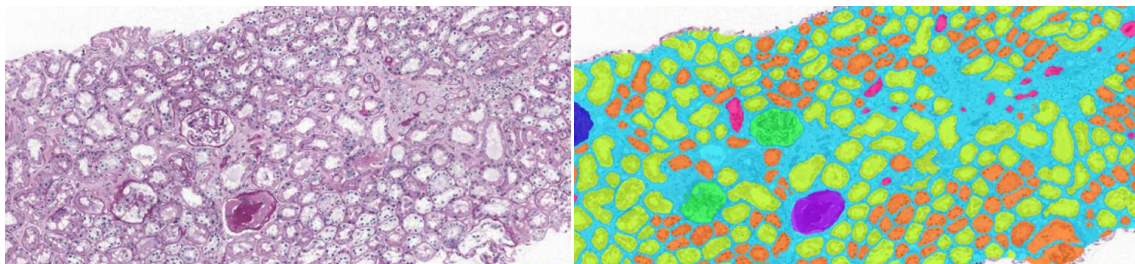


Figure 2.8: Example of classifying an histopathologic PAS stained slide at pixel level. The colors correspond to the following classes: light-blue/background, green/glomeruli, purple/sclerotic glomeruli, yellow/proximal tubuli, orange/distal tubuli, pink/atrophic tubuli and darkblue/ artery. (de Bel et al., 2018)

2.4.3 Semantic segmentation

Semantic segmentation is classifying an image at pixel level, in other words, we want to classify each pixel in the image to a class instead of classifying the patch as a whole with one label. An example can be seen in Figure 2.8. This is technique widely applied in segmenting tissue samples. The first semantic segmentation were based on classifying patches and then handing the label to the center pixel of the patch (Ciresan et al., 2012). Long et al. (2015) proposed a architecture called a fully convolutional neural network (FCNN) which allowed pixel based predictions without any fully connected layer. This permit segmentation maps to be generated for images larger sizes and was much faster than the patch based approach.

One of the big challenges using a FCNN is dealing with the pooling layers. While pooling layers allow for an increase in the receptive field, they cause losing spatial information. However, with semantic segmentation we want an output for each pixel of the original image. A popular way around pooling layers is to use dilated convolutions instead. These are convolutions with padding and striding on the convolution kernel. An example of a dilated convolution is shown in Figure 2.9.D. Dilated convolutions allow

<table><tr><td>0</td><td>0</td><td>1</td><td></td><td></td></tr><tr><td>2</td><td>3</td><td>2</td><td></td><td></td></tr><tr><td>0</td><td>1</td><td>0</td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr></table> <table><tr><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>2</td><td>1</td></tr><tr><td>2</td><td>3</td><td>0</td></tr></table> Kernel Dot product with kernel at given location	0	0	1			2	3	2			0	1	0													1	1	1	1	2	1	2	3	0	$0 \times 1 + 0 \times 1 + 1 \times 1 + 2 \times 1 + 3 \times 2 + 2 \times 1 + 0 \times 2 + 1 \times 3 + 0 \times 0 = 7$ Result is a single output value at the center of the kernel	<table><tr><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>2</td><td>1</td></tr><tr><td>2</td><td>3</td><td>0</td></tr></table> Kernel → ↓ The kernel is swept over the feature map and results in output feature map	1	1	1	1	2	1	2	3	0																																																																																																																																																																																																																																																																																																																																												
0	0	1																																																																																																																																																																																																																																																																																																																																																																																							
2	3	2																																																																																																																																																																																																																																																																																																																																																																																							
0	1	0																																																																																																																																																																																																																																																																																																																																																																																							
1	1	1																																																																																																																																																																																																																																																																																																																																																																																							
1	2	1																																																																																																																																																																																																																																																																																																																																																																																							
2	3	0																																																																																																																																																																																																																																																																																																																																																																																							
1	1	1																																																																																																																																																																																																																																																																																																																																																																																							
1	2	1																																																																																																																																																																																																																																																																																																																																																																																							
2	3	0																																																																																																																																																																																																																																																																																																																																																																																							
A. <table><tr><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td></tr><tr><td>2</td><td>3</td><td>2</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>2</td><td>1</td><td>1</td><td>3</td></tr><tr><td>0</td><td>2</td><td>0</td><td>0</td><td>2</td></tr></table> Normal <table><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td>7</td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr></table> <table><tr><td>7</td><td>10</td><td>12</td></tr><tr><td>9</td><td>13</td><td>15</td></tr><tr><td>6</td><td>11</td><td>14</td></tr></table> 3 x 3 output featuremap	0	0	1	1	1	2	3	2	0	1	0	1	0	0	1	0	2	1	1	3	0	2	0	0	2												7																			7	10	12	9	13	15	6	11	14	B. <table><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>2</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>3</td><td>2</td><td>0</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>2</td><td>1</td><td>1</td><td>3</td><td>0</td></tr><tr><td>0</td><td>0</td><td>2</td><td>0</td><td>0</td><td>2</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table> Padding <table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td>6</td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table> <table><tr><td>6</td><td>14</td><td>15</td><td>8</td><td>6</td></tr><tr><td>7</td><td>14</td><td>11</td><td>6</td><td>7</td></tr><tr><td>6</td><td>15</td><td>13</td><td>9</td><td>14</td></tr><tr><td>3</td><td>12</td><td>10</td><td>7</td><td>14</td></tr><tr><td>4</td><td>7</td><td>6</td><td>7</td><td>8</td></tr></table> 5 x 5 output featuremap	0	0	0	0	0	0	0	0	2	0	1	1	1	0	0	0	3	2	0	1	0	0	0	1	0	0	1	0	0	0	2	1	1	3	0	0	0	2	0	0	2	0	0	0	0	0	0	0	0																								6																																	6	14	15	8	6	7	14	11	6	7	6	15	13	9	14	3	12	10	7	14	4	7	6	7	8	C. <table><tr><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td></tr><tr><td>2</td><td>3</td><td>2</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>2</td><td>1</td><td>1</td><td>3</td></tr><tr><td>0</td><td>2</td><td>0</td><td>0</td><td>2</td></tr></table> Striding <table><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr></table> <table><tr><td>3</td></tr></table> 1 x 1 output featuremap	0	0	1	1	1	2	3	2	0	1	0	1	0	0	1	0	2	1	1	3	0	2	0	0	2																																									3	D. <table><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>2</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>3</td><td>2</td><td>0</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>2</td><td>1</td><td>1</td><td>3</td><td>0</td></tr><tr><td>0</td><td>0</td><td>2</td><td>0</td><td>0</td><td>2</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table> Striding and Padding <table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table> <table><tr><td>12</td><td>10</td><td>10</td></tr><tr><td>9</td><td>3</td><td>6</td></tr><tr><td>8</td><td>10</td><td>7</td></tr></table> 3 x 3 output featuremap	0	0	0	0	0	0	0	0	2	0	1	1	1	0	0	0	3	2	0	1	0	0	0	1	0	0	1	0	0	0	2	1	1	3	0	0	0	2	0	0	2	0	0	0	0	0	0	0	0																																																									12	10	10	9	3	6	8	10	7
0	0	1	1	1																																																																																																																																																																																																																																																																																																																																																																																					
2	3	2	0	1																																																																																																																																																																																																																																																																																																																																																																																					
0	1	0	0	1																																																																																																																																																																																																																																																																																																																																																																																					
0	2	1	1	3																																																																																																																																																																																																																																																																																																																																																																																					
0	2	0	0	2																																																																																																																																																																																																																																																																																																																																																																																					
	7																																																																																																																																																																																																																																																																																																																																																																																								
7	10	12																																																																																																																																																																																																																																																																																																																																																																																							
9	13	15																																																																																																																																																																																																																																																																																																																																																																																							
6	11	14																																																																																																																																																																																																																																																																																																																																																																																							
0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																			
0	2	0	1	1	1	0																																																																																																																																																																																																																																																																																																																																																																																			
0	0	3	2	0	1	0																																																																																																																																																																																																																																																																																																																																																																																			
0	0	1	0	0	1	0																																																																																																																																																																																																																																																																																																																																																																																			
0	0	2	1	1	3	0																																																																																																																																																																																																																																																																																																																																																																																			
0	0	2	0	0	2	0																																																																																																																																																																																																																																																																																																																																																																																			
0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																			
		6																																																																																																																																																																																																																																																																																																																																																																																							
6	14	15	8	6																																																																																																																																																																																																																																																																																																																																																																																					
7	14	11	6	7																																																																																																																																																																																																																																																																																																																																																																																					
6	15	13	9	14																																																																																																																																																																																																																																																																																																																																																																																					
3	12	10	7	14																																																																																																																																																																																																																																																																																																																																																																																					
4	7	6	7	8																																																																																																																																																																																																																																																																																																																																																																																					
0	0	1	1	1																																																																																																																																																																																																																																																																																																																																																																																					
2	3	2	0	1																																																																																																																																																																																																																																																																																																																																																																																					
0	1	0	0	1																																																																																																																																																																																																																																																																																																																																																																																					
0	2	1	1	3																																																																																																																																																																																																																																																																																																																																																																																					
0	2	0	0	2																																																																																																																																																																																																																																																																																																																																																																																					
3																																																																																																																																																																																																																																																																																																																																																																																									
0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																			
0	2	0	1	1	1	0																																																																																																																																																																																																																																																																																																																																																																																			
0	0	3	2	0	1	0																																																																																																																																																																																																																																																																																																																																																																																			
0	0	1	0	0	1	0																																																																																																																																																																																																																																																																																																																																																																																			
0	0	2	1	1	3	0																																																																																																																																																																																																																																																																																																																																																																																			
0	0	2	0	0	2	0																																																																																																																																																																																																																																																																																																																																																																																			
0	0	0	0	0	0	0																																																																																																																																																																																																																																																																																																																																																																																			
12	10	10																																																																																																																																																																																																																																																																																																																																																																																							
9	3	6																																																																																																																																																																																																																																																																																																																																																																																							
8	10	7																																																																																																																																																																																																																																																																																																																																																																																							

Figure 2.9: A step by step explanation of the dot product operation between the kernel and an input feature map. (A) The result of swiping a 3x3 kernel over a 5x5 feature map. (B) The result of swiping a swiping a 3x3 kernel over a 5x5 feature map with a zero-padding of 1. (C) The result of swiping a strided kernel of 3x3 over a 5x5 feature map with a stride of 1. (D) The result of swiping a strided kernel of 3x3 over a zero-padded 5x5 feature map, respectively having a stride and padding of 1.

for an increased receptive field without the need to downsample the input. Another often used solution is encoder-decoder architecture. The encoder network part gradually reduces the spatial dimension to get increasingly abstract features encoding the original image. The decoder part of the architecture recovers detail at the pixel-level of the original image, resulting in a dense pixel-wise segmentation (Badrinarayanan et al., 2015).

This structure was used in a very successful way by Ronneberger et al. (2015) called U-net and won in multiple competitions. The encoder part of the network chains blocks consisting of two convolutional layers with one max-pooling layer. The decoder part of the network uses blocks consisting of a deconvolutional layer chained with two convolutional layers. The deconvolutional layer (also known as fractionally strided layer, transposed convolution or upconvolution) uses convolutions to increase the spatial dimensions of its input until it has recovered the original pixel density (Figure 2.10) (Dumoulin and Visin, 2016). Next to that, the U-net has so-called 'skip-connections', which concatenate the features after each deconvolution with the features of the encoder path at the same spatial density. Figure 2.11 shows the architecture that the original U-net paper uses where the encoder-decoder blocks are respectively on the left and right side and the 'skip-connections' are noted with grey arrows. In this project both FCNN and U-net will be investigated for their use in the semantic segmentation of tumor and non-tumor tissue in multiplex IHC images.

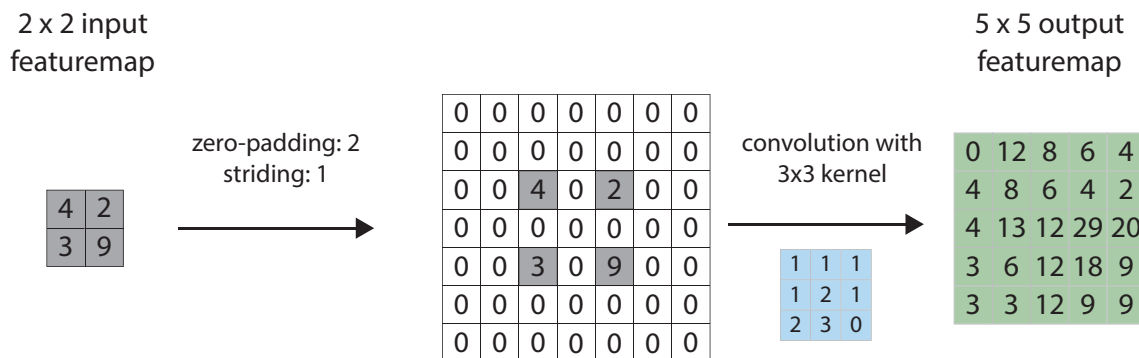


Figure 2.10: A deconvolution of a 2x2 input feature map (grey), resulting in a 5x5 output feature map (green). It is equivalent to a normal convolution of a 3x3 kernel (blue) on a strided 2x2 input with zero-padding of 2.

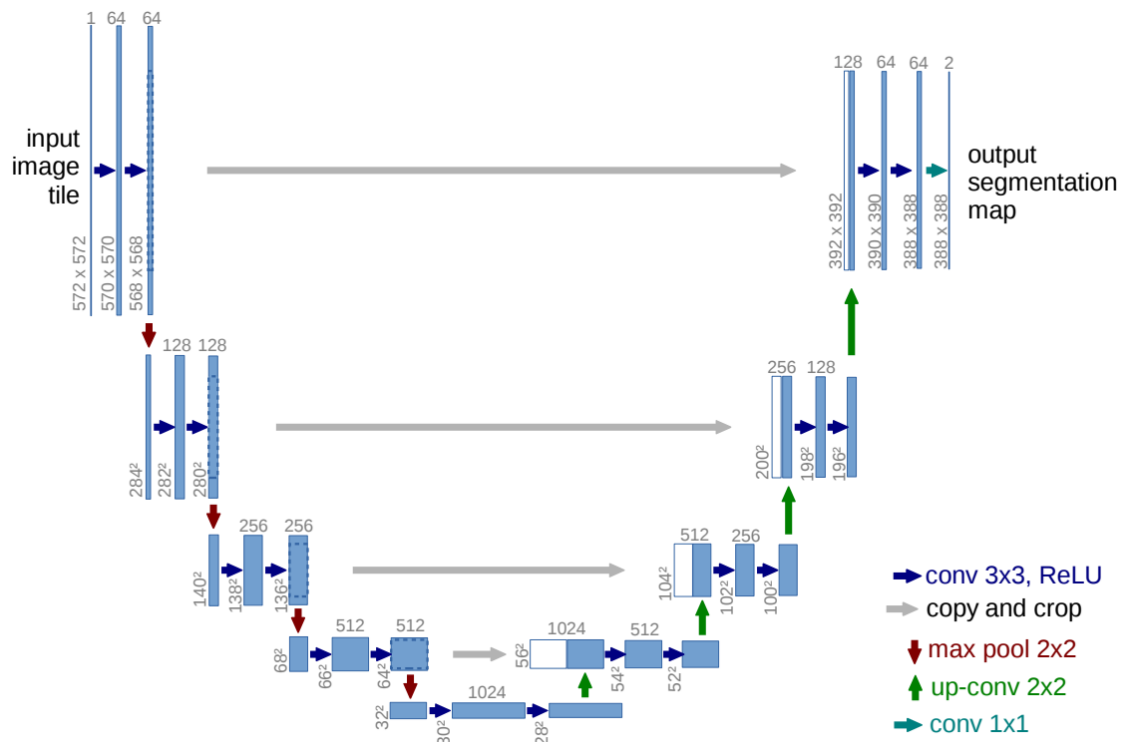


Figure 2.11: The architecture of the original U-net. The blue boxes depict the feature maps, with the number of features above the box. The image dimensions are shown in the left-bottom corner of each feature map. (Ronneberger et al., 2015)

3. Methods

3.1 Acquisition and pre-processing

3.1.1 Material

For this study formalin-fixed paraffin-embedded breast tissue from non-treated triple negative breast cancer patients was used.

3.1.2 Preparation and staining

Before TSA staining 3 μm paraffin tissue sections were prepared on Superfrost Plus slides (Thermo Scientific). Paraffin was removed using xylene and slides were treated with ethanol 99% and distillate water to rehydrate tissue. After rehydration, tissue was washed with wash buffer TBS-T (Thermo Scientific) and placed in a citrate buffer. To improve the antigen-antibody interaction, an antigen retrieval method was performed. Slides were cooked in the microwave for three minutes at maximum power and for 10 minutes at 20% power to break protein cross-links formed by formalin fixation. After the antigen retrieval step, tissue was washed for two minutes with distillate water and TBS-T. To minimize the amount of staining solution, a hydrophobic barrier pen was used to decrease the slide surface area.

For staining, slides were incubated with a TBS-T + 1% BSA blocking buffer for 10 minutes to block non-specific binding of antibodies. Next, the slides were incubated with primary antibody for 60 minutes and washed with TBS-T followed by 30-minute incubation with the secondary antibody. The secondary antibody was conjugated with horseradish peroxidase enzyme(HRP) and washed with TBS-T. After washing, slides were incubated for 10 minutes with TSA molecules, washed with TBS-T and added to a citrate buffer. An antigen retrieval was performed to remove the antibody by cooking slides in microwave. To detect all targets of interest, the staining steps were repeated for all antibody's. Table 2.1 gives an overview of the fluorophores that were used together with their target, antibody and emitting wavelength.

Table 3.1: Used fluorophores and antibody's.

$\lambda(\text{nm})$	Fluorophore	Antibody	Target
570	Opal 570 (PerkinElmer)	CD3 (Dako)	All T-cells
650	Opal 650 (PerkinElmer)	CD8 (Dako)	Cytotoxic T-cell
690	Opal 690 (PerkinElmer)	CD45RO (Dako)	Memory T-cell
488	DAPI	-	DNA A-T rich regions
620	Opal 620 (PerkinElmer)	FOXP3 (Bioscience)	Regulatory T-cell
540	Opal 540 (PerkinElmer)	Ki67 (Thermo scientific)	Ki67 nuclear protein
520	Opal 520 (PerkinElmer)	CK8-18 (Becton Dickson)	Epithelial cells

3.1.3 Scanning instrument

For scanning the stained slides a Vectra® 3.0 System (Perkin Elmer) was used. All slides were scanned with a 20x setup and the spectral cubes: DAPI, FITC, Cy3™, Texas Red™ and Cy5™ from Olympus Corporation.

3.1.4 Scanning and unmixing

The scanning instrument allowed to set up regions for acquisition. To save storage space a region of interest was selected for each stained slide before scanning. The region was divided into tiles and scanned at a magnification of 20x (Figure 3.1.1). All tiles were stored as an spectral file. To unmix a single tile containing multiple acquired spectral signals, software InForm® (Perkin Elmer) was used. To correct the image for autofluorescence effects, autofluorescence regions were selected on three non-stained breast tissue slides of the same cohort. This resulted in a single unmixed tile in a multi-layer tif format with dimension 1000 × 1350 pixels and each layer corresponding to an unique fluorophore intensity (Figure 3.1.2).

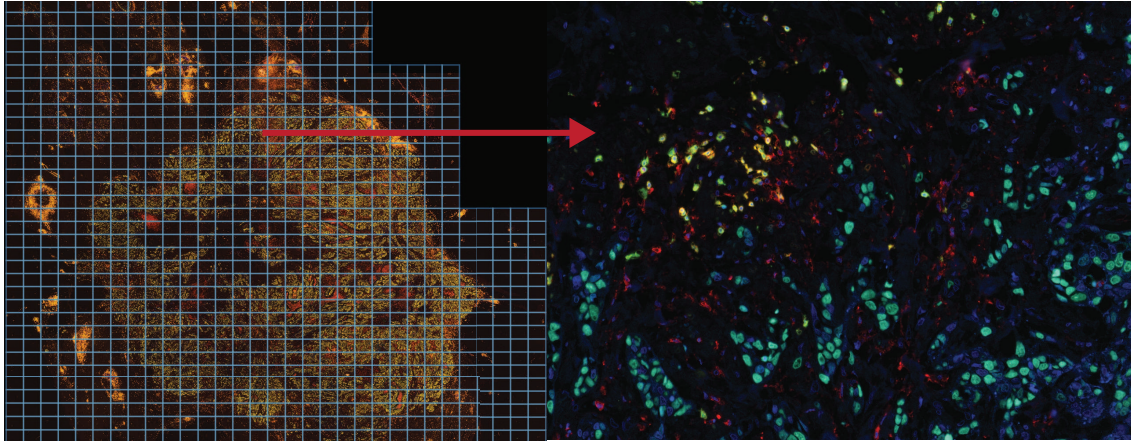
3.1.5 Conversion to multichannel whole-slide images

To convert multiple multi-spectral tiles to one whole-slide image a method had to be designed, since the used multi-spectral setup did not allow scanning slides as a whole-slide image. Therefore, a single tile was unmixed manually and afterwards the performed actions (inForm® nomenclature) were saved. The saved steps were used in the batch converter option of the inForm® software to convert a whole queue of images. To reduce the amount of manual labor involved in this step a script was created to manage batch conversion (Algorithm 1). The script generated shortcuts for the multi-spectral files that needed to be converted to multi-layer tif. The folder with the created shortcuts acted as a file cue and was managed by the script. Files converted to a multi-layer tif file were renamed and moved to a folder of their corresponding slide.

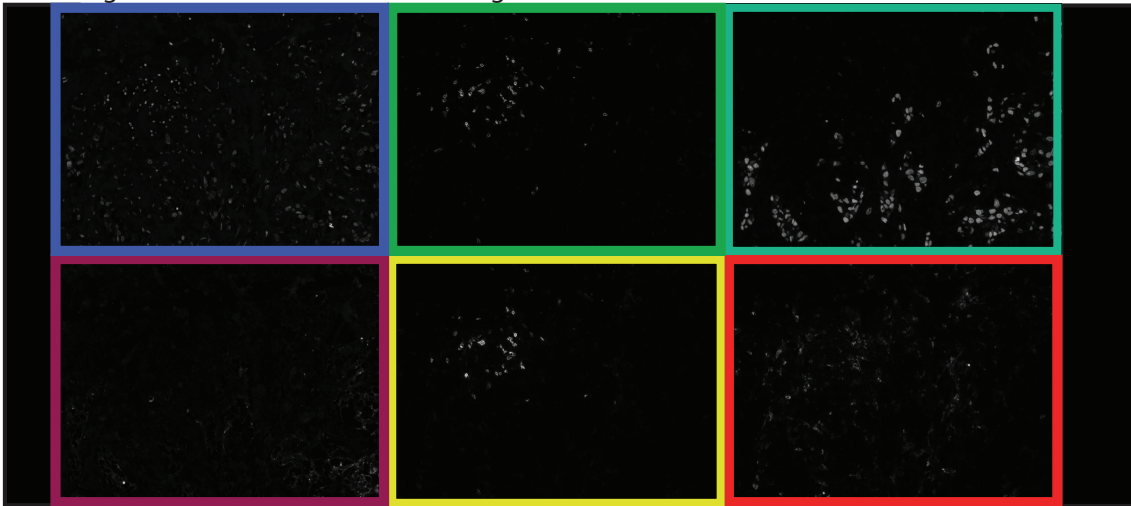
```
Data: Input/Output folders
Result: Move and rename files
check if new tif files;
if new tif files then
    for each new file do
        create a slide directory if not available;
        rename file to uniform standard;
        remove conversion cue shortcut;
    end
end
check if tif files still missing;
if missing tif files then
    for each missing file do
        create conversion cue shortcut;
    end
end
```

Algorithm 1: Script that manages batch conversion of multi-spectral images.

1. A multiplex IHC stained slide was split into a grid and scanned tile by tile in high resolution.



2. All fluorescent signals were separated by applying spectral color unmixing and a multi channel tif tile was generated.



3. All generated multichannel tif tiles were stitched together to form a multi channel whole slide image.

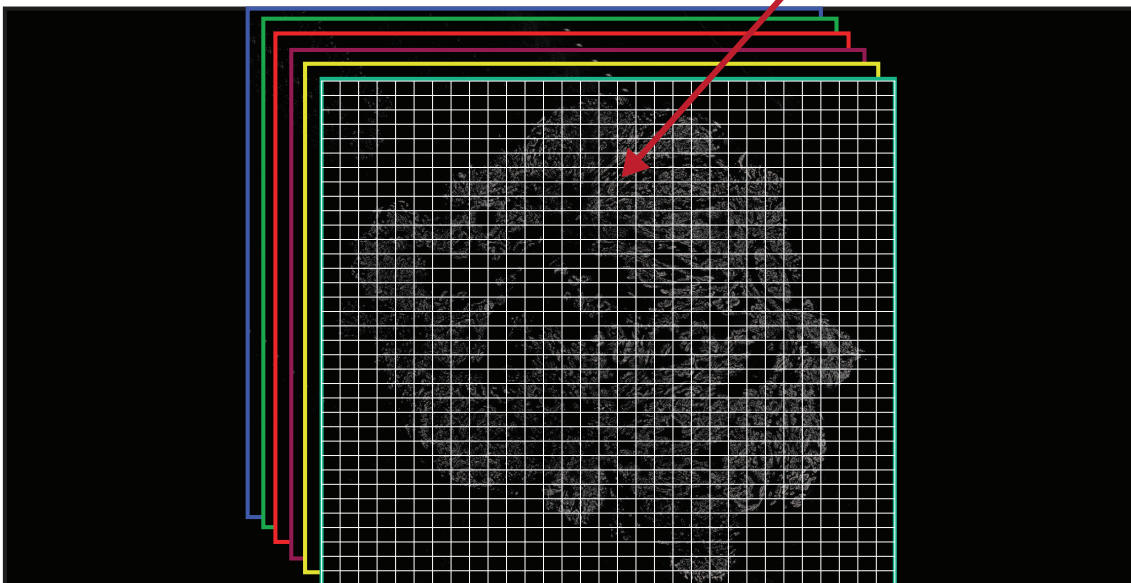


Figure 3.1: Flow of the conversion to multichannel whole-slide images.

After all spectral tiles were unmixed the whole-slides were assembled (Figure 3.1.3). This was done by collecting all converted tiles for one slide (Algorithm 2). First, to reduce the file size of a whole-slide, the 12bit spectral values were converted to 8 bit. Therefore, before assembling, a histogram was made containing all the minimum and maximum values of all tiles. The 95th percentile of the histogram was calculated and used to normalize a tile if it contained pixels with a value bigger than 255. Second, the total size of the WSI was predicted by extracting the coordinates from the filenames (added by the scanning instrument). These coordinates were also used to stitch the tiles in the right spatial order. After normalizing and assembling the whole-slide images were stored as multi-resolution multi-layer tif files.

Data: Single folder with all tif converted tiles

Result: A single WSI tif file

for *Each tile in folder* **do**

 Collect min and max value;

end

Create histogram collected min and max;

Set max: Take 95th percentile of histogram;

Set min: Collect smallest value ≥ 0 ;

Extract coordinates from filenames;

Calculate total size whole-slide array;

Sort files by coordinates;

for *Each tile in folder* **do**

if *max > 255 or min $\neq$ 0* **then**

 Normalize tile;

end

 Write tile to correct coordinates whole-slide array ;

end

Write output file to output folder;

Algorithm 2: Script that assembles multi-channel tiles to a whole-slide image

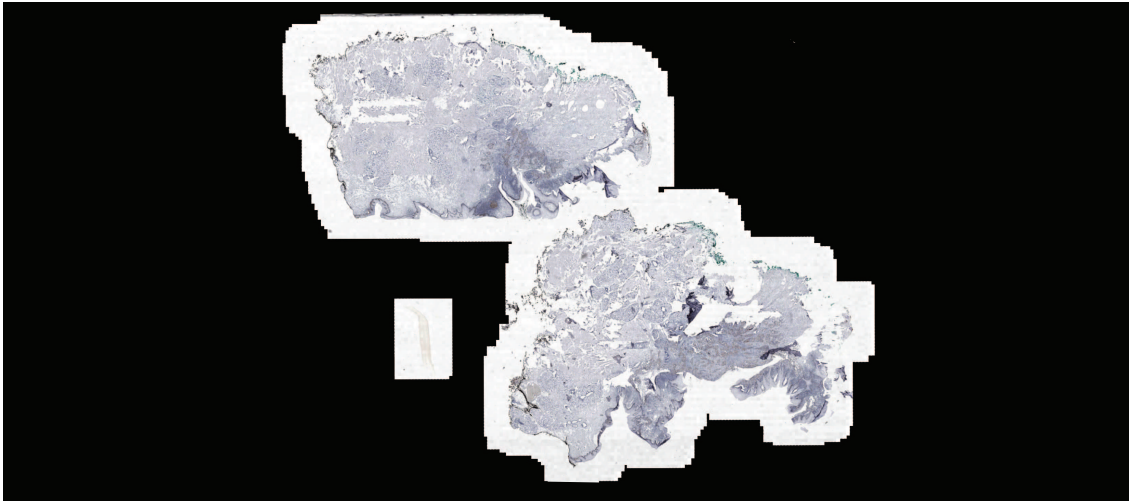
3.1.6 Multichannel whole-slide image to brightfield whole-slide-image

Annotating tumor regions is mostly done by experts on H&E images or IHC images. Fluorescent images with multiple colors are hard to interpreted as a human, therefore a method was designed to convert a fluorescent image to a brightfield IHC image. To create an whole-slide that allowed experts to accurately annotate tumor regions two fluorescent markers were chosen. For the nuclei the DAPI channel was selected and combined with the tumor region marker CK8-18. The color representation is an important feature that contribute to the decision making of experts. Hence, an expert was asked to select an IHC slide that contained, in his opinion, a good color contrast. This slide was used as an target slide and acted as an example to which the fluorescent data was converted (Figure 3.2.1).

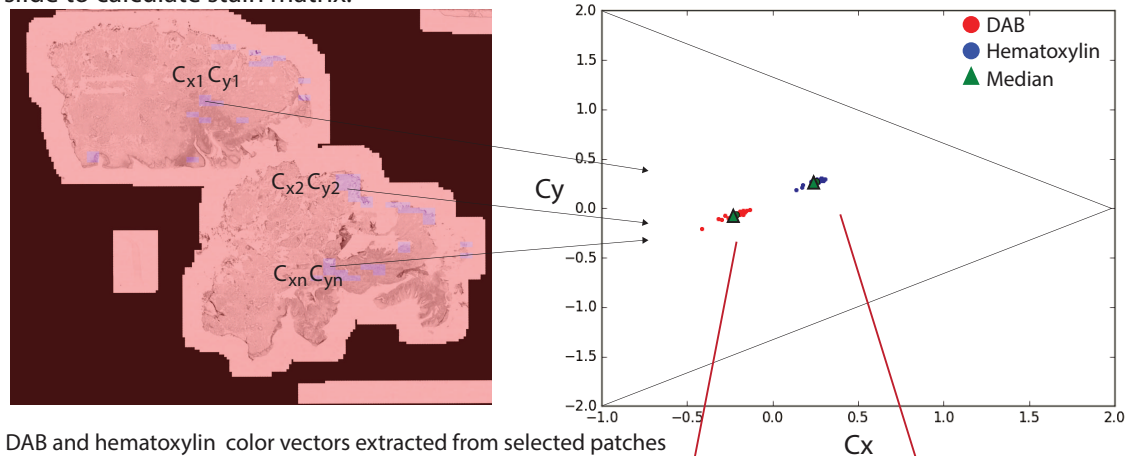
Previous research about automatic color unmixing was used to acquire the stain vectors of the target IHC slide (Geijs et al., 2018). From the two stain vectors, one for DAB and one for hematoxylin, a stain matrix was calculated (Figure 3.2.2). This resulted in the following stain matrix:

$$\begin{bmatrix} 0.71 & 0.63 & 0.32 \\ 0.33 & 0.54 & 0.78 \\ 0.00 & 0.00 & 0.00 \end{bmatrix} \quad (3.1)$$

1. A target IHC slide is selected and used to extract color vectors



2. Automatic color unmixing algorithm is applied and used to extract color vectors from target slide to calculate stain matrix.



DAB and hematoxylin color vectors extracted from selected patches

3. Reversed color deconvolution was applied by inverting the stain matrix and multiplying this with the fluorescent ununmixed channels. This resulted in an artificial IHC slide that mimicked the colors of the original target slide.

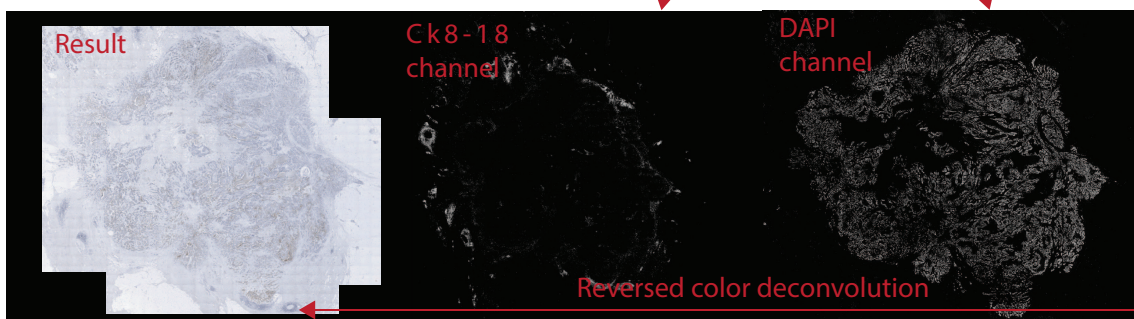


Figure 3.2: Flow of the conversion to artificial brightfield whole-slide images.

By multiplying the two fluorescent channels with the inverse of this stain matrix a brightfield image was generated (Figure 3.2.3). The DAPI channel, containing the cell nuclei, was multiplied with hematoxylin stain vector and the CK8-18 channel with the DAB stain vector. The assumption was made that the fluorescent signal was equal to the optical density (OD), since the intensity of this fluorescent signal was determined by the concentration of fluorophores at the tissue site.

3.2 Annotations

Two sets of annotations were created for training, validation and testing. Set 1 was used for training and validating the network and set 2 for testing the trained network. Annotations were made by a non-expert with a biomedical background and revised afterwards by a pathologist-resident. The open-source software ASAP 1.7 was used for annotating every slide. To help with visualization of the tumor artificial brightfield images were generated with the DAPI marker in a hematoxylin color and the tumor marker CK8-18 with a DAB color.

3.2.1 Set 1

For each individual slide annotations were divided in four different groups. These were 'tumor easy', 'tumor hard', 'non-tumor easy' and 'non-tumor hard'. Subclasses were for balancing purposes during training (which will be explained later) and were merged when validating into their overarching groups 'tumor' and 'non-tumor'. Figure 3.3 shows multiple examples for each class. Annotations belonging to the 'easy' groups were mainly large clearly defined regions. In case of the non-tumor this were simple stain artefacts or sparse cell areas as for the tumor regions these were large dense tumor cell areas with high morphological changes. The 'hard' groups existed out of more widely different annotations. Small crevices existing out of lymphocytes within tumor rich area's were added to the non-tumor group together with healthy epithelium. Very small isolated clumps of tumor cells together with unclear tumor cells or areas were added to the tumor group.

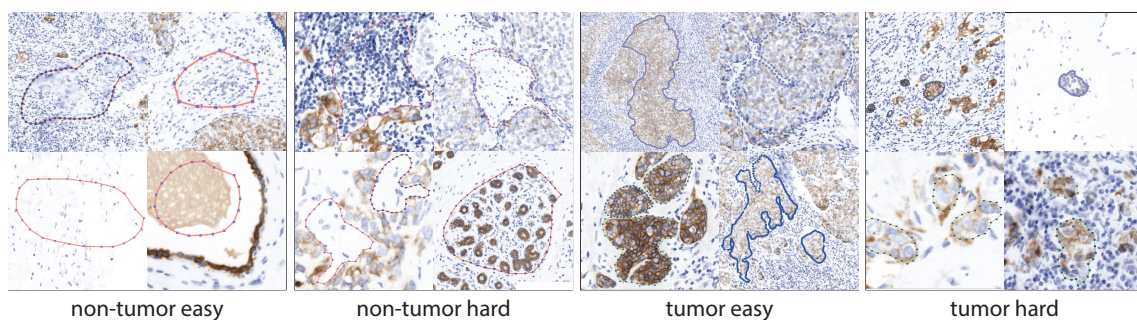


Figure 3.3: Annotated examples belonging to the classes: non- tumor easy, non-tumor hard, tumor easy, tumor hard.

A set of 41 slides were annotated consisting of 31 triple-negative slides and 10 normal healthy slides. Table 3.2 shows the total amount of annotations that were sampled and available. Randomly slides were distributed to a training and validation group. The training group contained a total of 2380 non-tumor and 3192 tumor annotations spread over 31 slides. The other 10 slides were part of the validation set and from this set 623

non-tumor and 927 tumor annotations were available. In this way the total amount of annotations available for each set had a ratio of 2:1. In addition to the 2380 manual non-tumor annotations in the training set, false positives from preliminary results were added. Therefore, a simple mask was drawn that excluded all the possible tumor tissue. All the positives outside this mask were considered as false positives. The false positives were added to the 'non-tumor hard' class. These false positives mainly consisted out of healthy epithelium and massively enriched the amount of healthy epithelium samples in the training data set.

Table 3.2: Overview of all training and validation annotations sampled from 41 slides.

	Non-tumor		Tumor	
Training	1757		2265	
Validation	623		927	
Total	2380		3192	
	easy	hard	easy	hard
Training	720	1037	1299	966
Validation	263	360	587	340
Total	983	1397	1886	1306

3.2.2 Set 2

To evaluate the segmentation performance of the trained networks regions were selected and fully annotated. An evaluation set containing 20 unique slides were annotated that were neither part of the training set nor the validation set. To balance the evaluation set, regions (hotspots) were manually selected for every slide and categorized into one of three groups: Tumor, healthy epithelium or background. The network has to classify a pixel to be tumor yes or no.

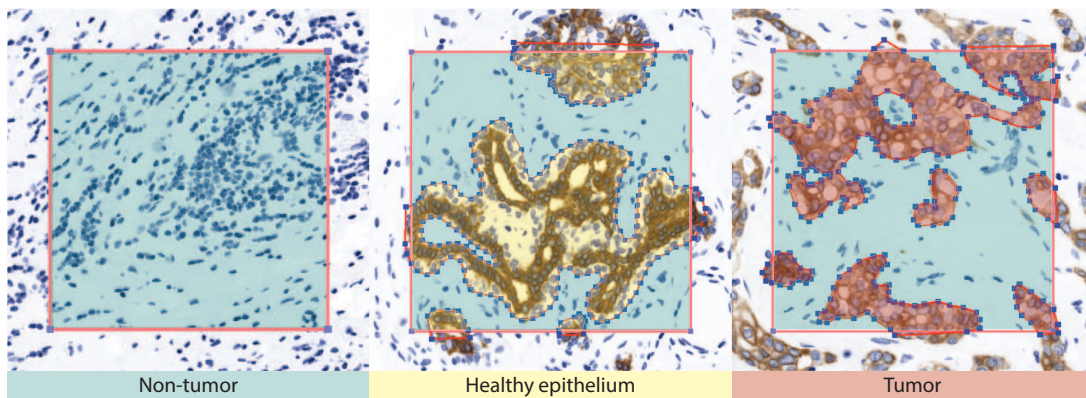


Figure 3.4: Example of fully annotated hotspot boxes

Healthy epithelium and tumor look very similar, therefore a third class was added called 'healthy epithelium'. Adding this group gives more insight in the ability of the network to discriminate between healthy epithelium and tumorous tissue. If possible, for each group two regions were selected per slide. The hotspots were chosen to represent the overall slide as good as possible. Around the center of the hotspots 512x512 pixel boundary boxes were generated. Inside the boxes the tissue was fully

annotated meaning that every pixel was part of one of the three previously described groups. Figure 3.4 shows for a group an example of a fully annotated hotspot.

3.3 Training, validation and testing

For conducting an experiment a network had to be trained, validated and tested. This was performed in a single workflow and the methods used for these multiple steps will be elaborated in the same order. Figure 3.5 gives an overview of the workflow and the methods used per step. The experiments performed all went through the same workflow with the same annotations.

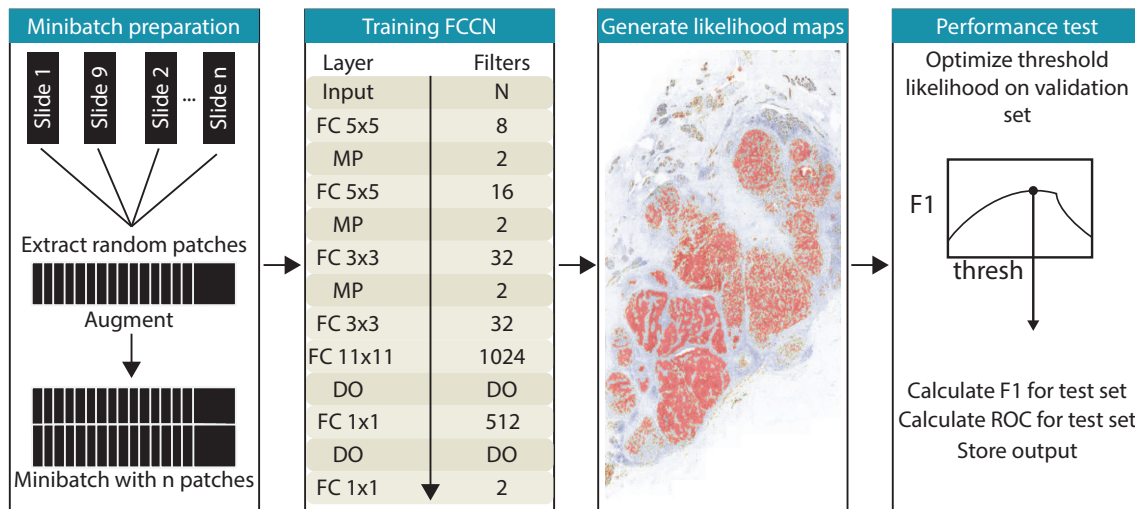


Figure 3.5: Workflow for training and validating a network.

3.3.1 Minibatch preparation

Batches filled with randomly chosen patches were propagated through the network when training or validating. When not balancing patches, all four classes were divided equally over each batch. Recruiting a patch for class consisted of picking a randomly labeled pixel from a pool of k slides. Due to memory limitations not all slides could be part of this pool. After i epochs the pool of slides was refreshed. The randomly selected pixel formed the center pixel of a $n \times n$ patch and that was extracted from the WSI. Four copies of the patch were added to the batch consisting of one original and three on which data augmentation was applied. Patches were flipped, rotated (0, 90, 180 and 270 degrees) blurred (random σ from $\{\sigma \in \mathbb{R} | 0.1 < \sigma < 0.5\}$) to make sure the network would not learn rotation-dependent and out-of-focus features. Patch balancing was applied to increase the amount of difficult patches in a batch. In this way the network spent most of the time training and validating on hard and complex patches.

3.3.2 Training FCCN

After patch extraction, a fCNN was trained where the performance of the network was monitored by assessing the accuracy on the validation set patches. Training was stopped when validation accuracy did not improve for ten epochs. To build and train the convolution neural network the open-source library Theano 1.0.0 together with Python 3.5 was used. All networks were trained using a GeForce GTX970 (Nvidia Corporation)

graphical processing unit. Preliminary, parameters (e.g. learning rate, momentum) were determined empirically to obtain maximum converging of the network loss in the first five epochs. The weight initialization sampled from the normal distribution was used and the initial learning rate was set to 0.000025 (He et al., 2015). The learning rate decay was set to 0.75 and applied if there was no improvement in validation accuracy after 4 iterations. An epoch always consisted of 100 training iterations with a mini-batch size of 32 patches and a validation epoch out of 100 iterations with a mini-batch of 128 patches.

One network structure was used and not modified (e.g. number of layers, filters per layer, number of nodes in fully connected layers) during the experiments. This FCNN was used for all experiments involving a FCNN, were only the training parameters changed per experiment. The fully FCNN consisted of 7 convolutional layers with filter sizes 5 x 5 in the first two convolutional layers, 3 x 3 in the third and fourth layers, 11 x 11 in the fifth layer and 1 x 1 in the last two layers. The number of filters were 8, 16, 32, 32, 1024, 512 and 2 respectively. Max pooling with 2 x 2 pooling size and stride of 2 was inserted after each of the first three convolutional layers to reduce the memory requirements of the network. After every convolutional layer batch normalisation was applied (Bánci et al., 2017). A L2 regularized cross-entropy cost function ($\lambda = 0.00005$) was applied after applying the softmax function on the output layer.

3.3.3 Generation cancer likelihood maps

A cancer likelihood map (CLM) was generated for all validation and test slides. A successfully trained network was applied on a slide in a tile-based fashion. The output of the FCNN indicating cancer likelihood per pixel was then reassembled to match the slides dimensions. The prediction of the network was normalized to [0,1], with 0 being 0% confident of tumor and 1 being 100% confident of tumor.

3.3.4 Performance

To assess the performance of the trained network a set of CLM were compared with the ground truth. Figure 3.6 shows an example of the method that was used. A complete slide was scored by selecting an annotation and turning it into a binary mask by converting the coordinates to a polygon shape together with the corresponding prediction masks. The latter was converted to a binary mask by using a cut-off value. Pixels with a value above the cut-off value were labelled as tumor. Both masks were then compared according to table 3.3, which resulted in a comparison mask.

Every pixel in the comparison mask could be true negative(TN), false negative(FN), true positive(TP) or false positive(FP). Iterating through all annotations on per slide resulted in the total amount of TN, FN, TP and FP pixels per slide.

With this distribution known the Dice similarity coefficient (DSC) also known as F1-score was calculated for each slide. The F1 is defined as

$$F1 = \frac{2TP}{2TP + FP + FN} \quad (3.2)$$

where TP is the sum of *true positive* pixels, FP the sum of *false positive* pixels and FN the sum of *false negative* pixels.

Since the false/positive ratio depends on the cut-off value of the CLM, the cut-off value was optimized before applying a network on the test set. This was done by calculating the mean F1-score for the complete validation set with different cut-off

values. The cut-off value resulting in the highest F1-score was selected for testing the network. Figure 3.7 illustrates the effect of the cut-off value on the F1-score.

Table 3.3: Overview of true false labeling

Class	Ground truth			Network	
	Tumor	Healthy epithelium	Non-tumor	Tumor	Non-tumor
FN	0				0
TN		0	0		0
FP		0	0	0	
TP	0			0	

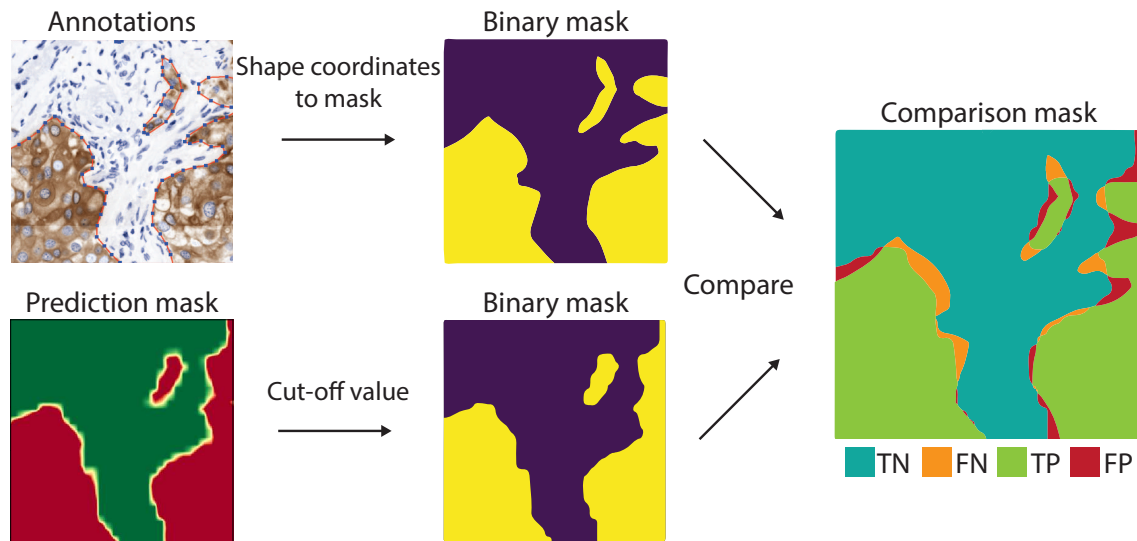


Figure 3.6: Example of comparing the annotated ground truth with the predicted outcome of the network. First both sources are converted to a binary mask after which they are compared. This results in an comparison mask where every pixel is labelled as true negative(TN), false negative(FN), true positive(TP) or false positive(FP).

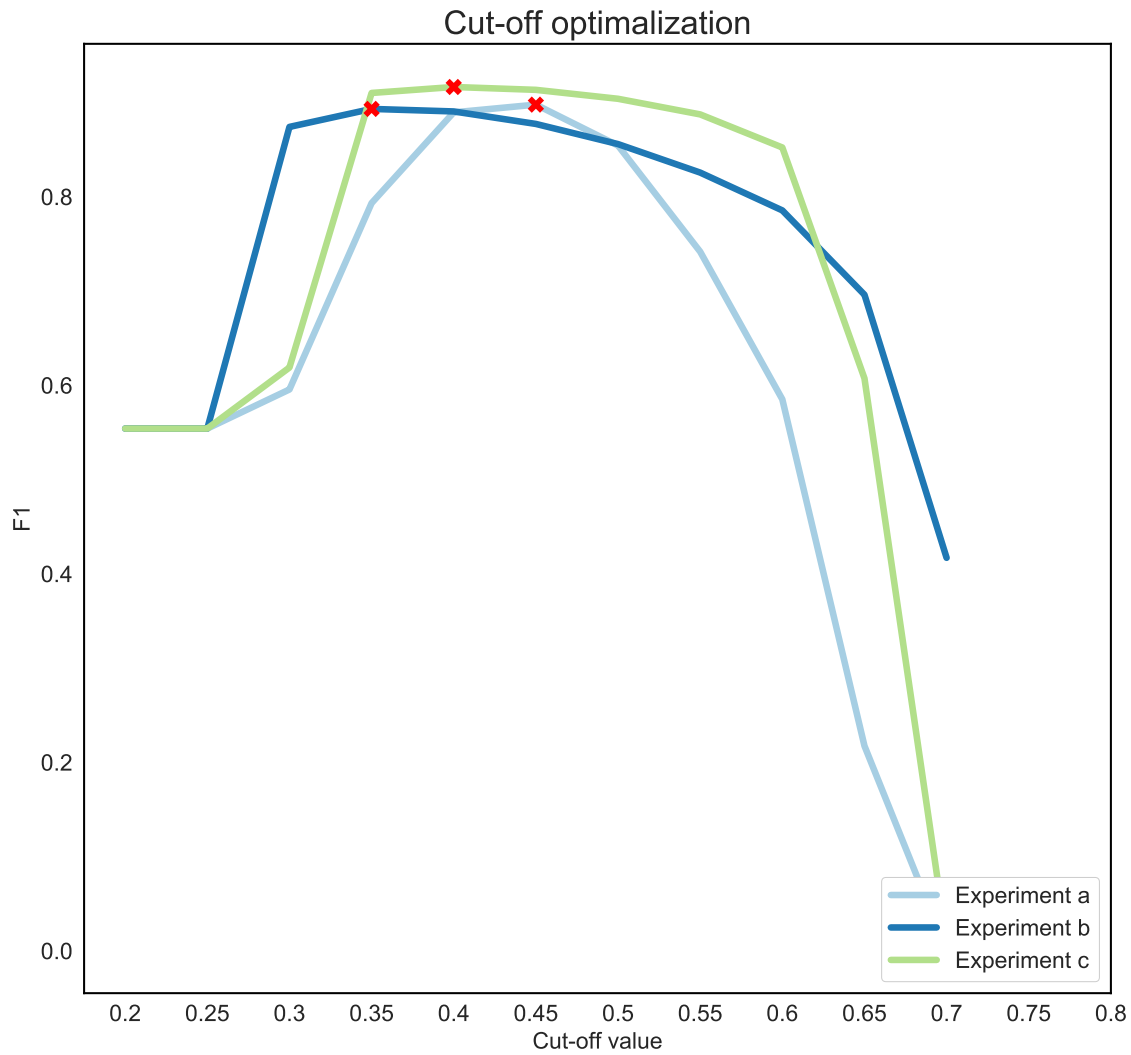


Figure 3.7: Graph with F1-score for a range of cut-off values done for multiple experiments. An optimal plateau can be seen, with a maximum F1-score (red cross). The cut-off resulting in the highest F1-score was selected for testing the network.

3.4 Experiments

The experiments that were conducted (according to the workflow described in Figure 3.5) can be divided into two groups. One group of experiments focused on the way the data was presented to the network. This involved evaluating the influence of context and resolution together with the influence of number of fluorescent data channels available for the network. The other experiments focused on prediction optimization. Hard mining was performed and multiple ensembles were developed to optimize the performance. To access the performance of the FCCN it was compared to another semantic segmentation network. A detailed overview of all experimental methods can be found in the sections below.

3.4.1 Channels

The data used to train the network differs from normal RGB data. Patches were sampled from spectral multi-resolution whole-slide images and therefore consisted out of a different type of data. Each channel contains spectral intensities that provide spacial information about the labelled immunomarker. Multiple networks were trained on datasets consisting out of patches with different channels, to validate the importance of using multiple channels for training besides DAPI. An overview of the used channels is given in Table 3.4. The first network was trained solely on the DAPI data channel. Tumor causes morphologic changes in the cells nuclei which can be learned by the network. In this way the network was kept independent of other markers. The second network included the CD3 marker during training, which gave the network spacial information about all types of lymphocytes. The third network was trained by using both of the tumor markers Ki67 and CK8-18. The fourth network was a combination of both a tumor marker and a lymphocyte marker. Ck8-18 and CD3 were chosen since they were both the most abundant marker in respectively tumor and non-tumor regions. The last two networks were trained on the whole set of data channels with and without the DAPI channel. These network were trained to test if adding more increased the performance of tumor detection and the importance of the DAPI channel. The performance (F1-score) of all six networks were compared together with their sensitivity and specificity.

Table 3.4: Overview of networks trained including different type of data channels

	CD3	CD8	CD45	DAPI	FoxP3	Ki67	CK8-18	Autofluorescence
Network 1.1				X				
Network 1.2	X			X				
Network 1.3				X		X	X	
Network 1.4	X			X			X	
Network 1.5	X	X	X	X	X	X	X	X
Network 1.6	X	X	X		X	X	X	X

3.4.2 Context vs spatial resolution

To allow the network to learn the appearance of cancer, it is important that the small patches contain enough information to allow discrimination between patches with and without cancer. In contrast, when one selects patches which are too large, it is harder for the network to identify the relevant discriminatory features (Litjens et al., 2016). Fixing the patch size means that when the spatial resolution is doubled the amount of context is halved. An example of this trade-off is shown in Figure 3.8 by showing the same patch for different spatial resolutions and patch sizes. As such, the optimal resolution and patch size at which to extract information was investigated and whether it was of huge importance. The channel configuration network 1.4 was used for this experiment. The maximum size of a patch is limited due to memory limitations of the GPU and therefore a maximum patch size of 256x256 was used. Table 3.5 gives an overview of the size and spatial resolution of the patches used to train the network. For each network the performance was calculated and compared.

Table 3.5: Overview of networks trained including different type of data channels

	0.48 $\mu\text{m}/\text{pix}$	0.96 $\mu\text{m}/\text{pix}$	1.92 $\mu\text{m}/\text{pix}$
64x64	Network 2.1	Network 2.2	Network 2.3
128x128	Network 2.4	Network 2.5	Network 2.6
256x256	Network 2.7	Network 2.8	Network 2.9

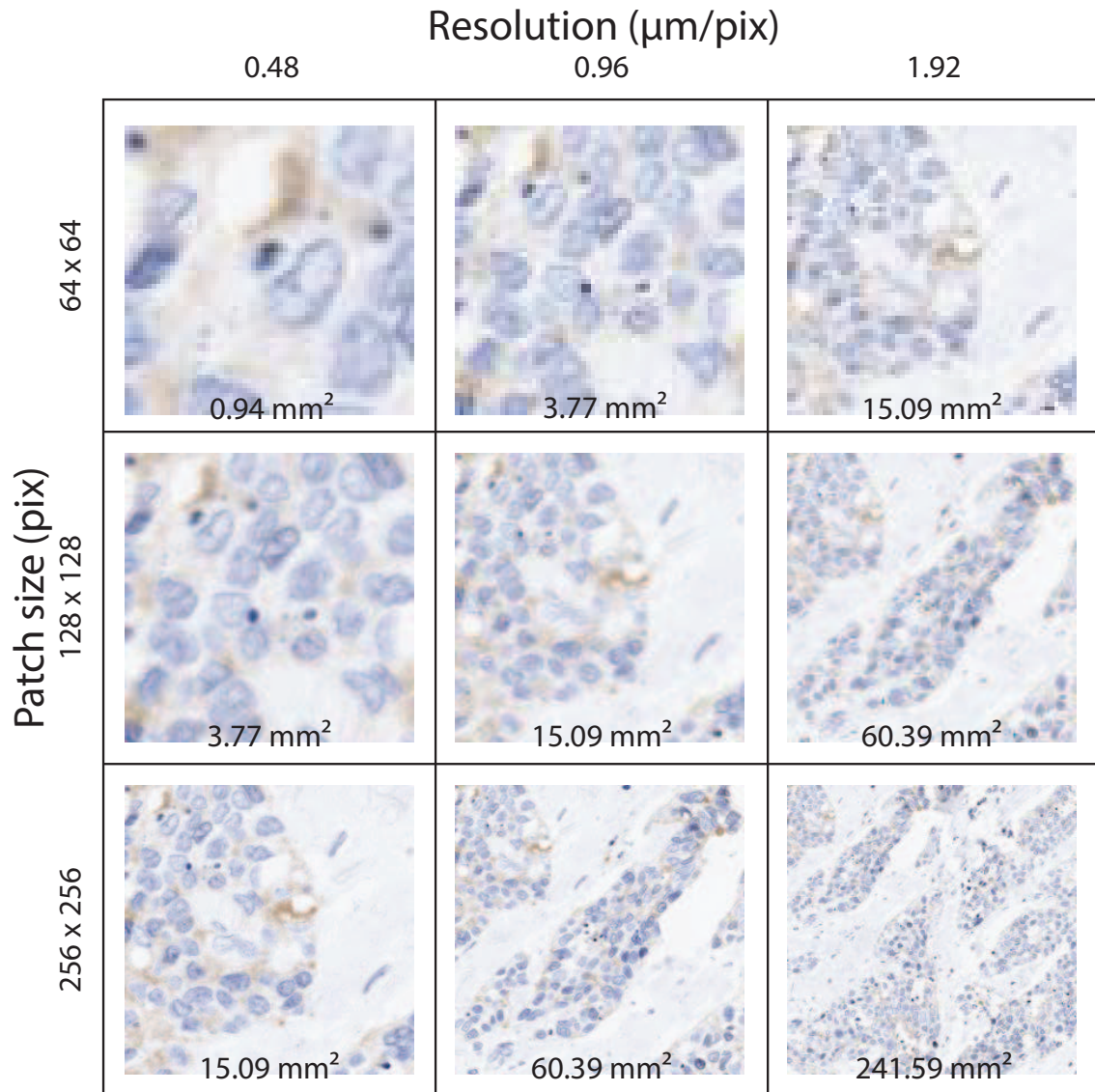


Figure 3.8: Example of the information available when patch size and the spatial resolution were changed.

3.4.3 Hard mining

Due to the nature of the triple negative breast tumor slides, the morphology of the tissue varies widely between the slides. Certain normal, mainly ducts, regions which look similar to cancer (e.g. Figure 3.4) are not available in every slide and underrepresented in the training data. This means that the network is not capable of correctly identifying these areas as normal. By balancing the patches based on their type and occurrence

the network can be trained on more difficult patches. More particular, we optimized our training batch in such a way that relatively rare challenging non-tumor and tumor patches were well represented, whereas patches obviously belonging to class non-tumor or tumor (which predominate both classes) were added rarely to the training batch. This allows us to spend most of the training time in learning important differences among tumor and non-tumor regions. To tackle this problem a boosting approach similar to the one used by [Cireřan et al. et al. \(2013\)](#) was used. Figure 3.9 shows how the original annotated masks were compared with the calculated CLM of network 2.5. All the pixels that were wrongly classified were added to their corresponding 'tumor hard' or 'non-tumor hard' class depending on their ground truth. This process resulted in additional training data which contained more difficult samples. Subsequently four new networks were trained with the new updated masks. The networks were trained with different balanced batches. Next, to the normal performance tests an additional confusion matrix was generated to investigate the misclassification of the four classes before and after performing hard mining.

Table 3.6: Overview of networks trained with different balanced batches. For each network the ratios of the composed batch are shown.

	Easy tumor	Hard tumor	Easy non-tumor	Hard non-tumor
Network 3.1	1	9	4	6
Network 3.2	3	7	6	4
Network 3.3	5	5	5	5
Network 3.4	2.5	7.5	5	5

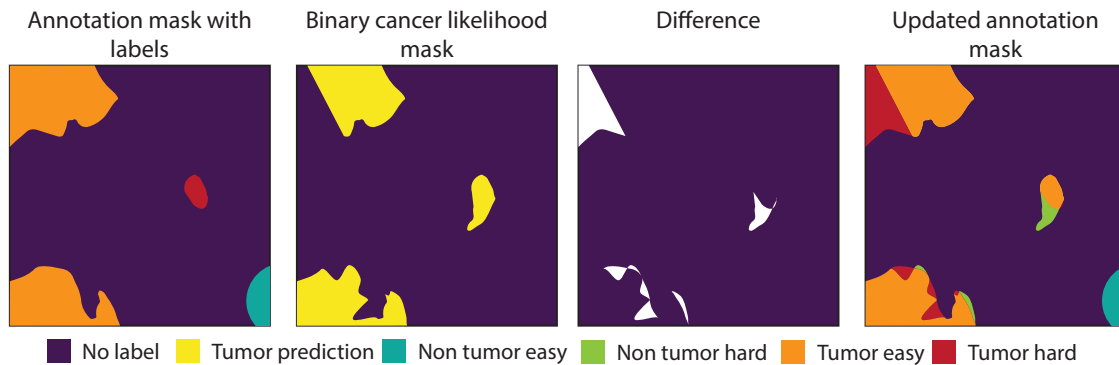


Figure 3.9: Example of the method used for updating the annotation masks. The network samples patches from these labelled locations.

3.4.4 U-net

With its encoder-decoder structure, the U-net produces segmentation maps at the pixel density of the input image. This makes it particularly well suited for our case, as small and dense tumor regions need accurate segmentations to be separated from non-tumor regions. The U-net architecture was used and build according to Figure 2.11. The 'encoder' section of the U consisted of 5 blocks. Every block consisted out of two convolutional layers with filter size 3 and both were normalized the block ended with a maxpool layer of 2x2. Five blocks were used with the number of filters for both convolutional layer in the block increasing per block. The number of filters used from

top to down were: 32, 64, 128, 256, 512. Before max pooling of the second last layer, a dropout channel layer was added. The 'decoder' side of the U consisted out of similar blocks. With the difference that a upconvolutional was performed on the previous output before the block was applied. The upconvolution had filter size 2, stride 2 filters according to the 'depth'. Then the 'encoder' layer of the same depth was concatenated with the upconvolution output and formed the input of a block. Three networks were trained with different set of parameters which can be seen in Table 3.7 . With knowledge of prior experiments of channels, patch size as resolution and patch class distribution setting were adjusted. Also the depth of the U-net structure was changed in network 4.3. A training minibatch consisted out of 2 patches, due to GPU memory limitations and a training epoch consisted out of 300 minibatch passes. The validation epoch consisted of 400 minibatch passes of 4 patches per minibatch. All patches that were used for all experiments had a size of 512 x 512

The performance (F1-score) of all networks were compared together with their sensitivity and specificity. Also, the performance of the performing FCNN was compared with that of the U-net networks.

Table 3.7: Overview of networks trained with different balanced batches. For each network the ratios of the composed batch are shown.

	Channels	Resolution	Class ratio	Depth
Network 4.1	3, 6	0.96 $\mu\text{m}/\text{pix}$	Et: 5 Ht:5 Ent:5 Hnt:5	5
Network 4.2	3, 6	1.92 $\mu\text{m}/\text{pix}$	Et: 2.5 Ht:7.5 Ent:5 Hnt:5	5
Network 4.3	3	0.96 $\mu\text{m}/\text{pix}$	Et: 5 Ht:5 Ent:5 Hnt:5	6

3.4.5 Ensemble

Ensemble methods, work on a higher level to improve the performance of "unstable" predictors and reduces the variance of predictions (Qiu et al., 2014). A simple way to combine the predictions is to calculate the average of the predictions from different network outputs, and is called a 'model averaging ensemble'. When having an odd number of ensemble members it is possible to apply majority voting on the predicted labels to create an 'majority-vote ensemble'. A more advanced ensemble is the approach called model stacking or stacked generalization. Here, a new model is used to learn how to best combine the predictions from each ensemble member (Wolpert, 1992).

The methods 'model averaging ensemble', 'majority-vote ensemble' and 'model stacking' were investigated if ensemble methods enhance the performance and reduce the variance of predictions. The 'model averaging ensemble' was assembled by taking the mean of the CLM of three different models. Table 3.8 gives an overview of the models used. The mean was calculated and a new CLM was generated. New optimal cut-off value were determined and eventually new F1-scores calculated.

A similar approach was used for the 'majority vote ensemble' method, with the difference that first the CLM maps were converted to binary maps by applying their optimal cut-off value. The labels were assigned

$$label = \begin{cases} 0 & \text{if } \mu < mean(N) \\ 1 & \text{if } \mu > mean(N) \end{cases} \quad (3.3)$$

where μ is the population mean and N the number of binary maps. A 1 was labeled as tumor (positive) and 0 as non-tumor (negative).

Table 3.8: Overview of CLM outputted by different networks used for the ensemble 'model averaging' and 'majority vote'.

	Type	Net 1	Net 2	Net 3
Model 6.1	Average	1.4	3.3	4.1
Model 6.2	Majority	1.4	3.3	4.1

Model stacking was performed by creating a modified dataset with added the predictions of multiple selected models and their 'model averaging ensemble' output as data channels. The modified files were then used to train a new network. This was done with the same parameters as previous experiments. Different combinations of data channels were selected from the modified dataset for training. Table 3.9 gives an overview of the data channels that were used. The performance (F1-score) of all ensembles were compared together with their sensitivity and specificity.

Table 3.9: Overview of model stacks trained on different data channels and CLM.

	CD3	CD8	CD45	DAPI	FoxP3	Ki67	CK8-18	Mod 6.1	Net 1.4	Net 3.3	Net 4.1
Network 6.3	X			X			X		X	X	X
Network 6.4									X	X	X
Network 6.5				X			X	X			
Network 6.6				X				X			

4. Results

In total five experiments were conducted and in this section the results are presented per experiment. The sectioning is in the same order as the experiments described in the methods. Due to the high amount of changing variables per experiment the results are presented as readable possible without the need of referencing the methods too much. However, when the results do not provide enough clarity it is advised to reference the corresponding section in the methods.

4.1 Experiment 1: Channels

All the networks were successfully trained with the different channels available in the patches. The results for the F1-score, specificity and sensitivity are shown in Figure 4.1.

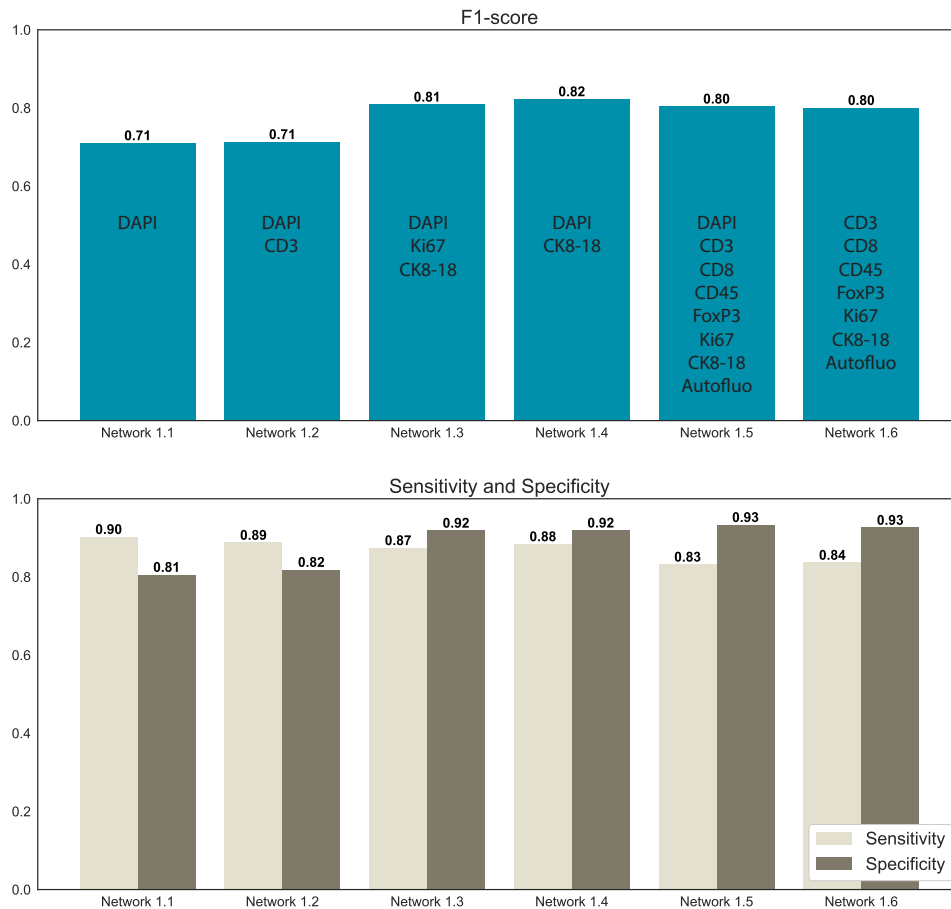


Figure 4.1: Results of networks trained on different type of data channels

The network (experiment 1.4) with the highest F1-score was trained on patches with DAPI, CD3 and CK8-18 channels available. The second highest F1-score was achieved by the network (experiment 1.3) trained on the DAPI channels and only the tumor markers channels, Ki67 and CK8-18.

Adding more data channels to a patch increased the specificity of the network. The networks trained in experiment 1.3, 1.4, 1.5 and 1.6 achieved higher specificities than the networks trained in experiment 1.1 and 1.2. The specificity difference between these experiments was roughly 10% . Experiment 1.5 and 1.6 achieved the highest specificity of 0.94 and experiment 1 the lowest of 0.83. However, increasing the amount of data channels per patch resulted in a even decrease of sensitivity. Experiment 1.1 scored the highest sensitivity of 0.86 and experiment 1.6 the lowest of 0.81.

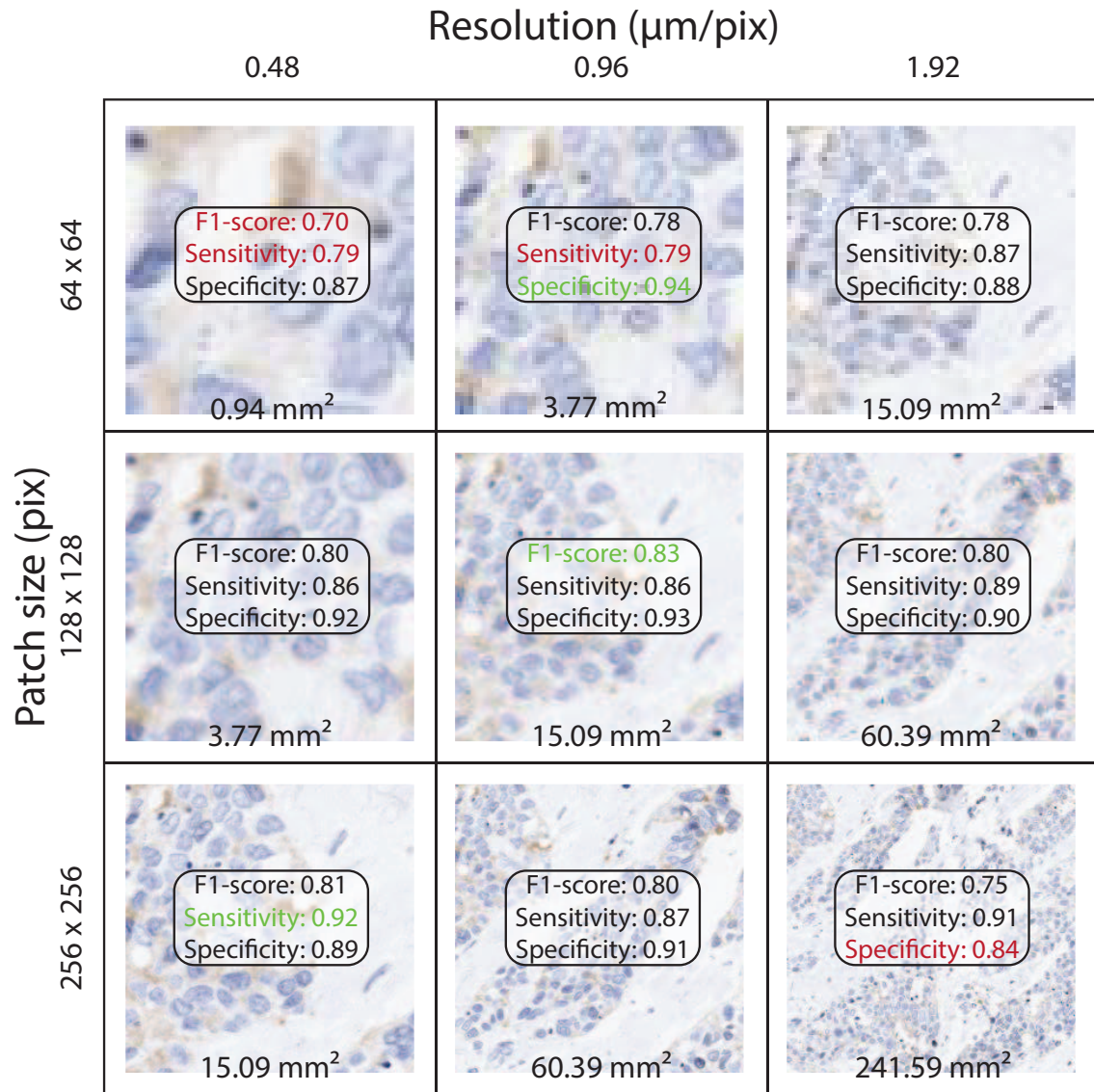


Figure 4.2: Results of networks trained on different resolutions and patch sizes.

4.2 Experiment 2: Context vs spatial resolution

All nine networks were successfully trained and tested. The results are shown in Figure 4.2. The highest F1-score, sensitivity and specificity are marked with a green color, the lowest scores with a red color. The network trained with 128x128 patches on 0.96 $\mu m/pix$ achieved the highest F1-score and the network with 64x64 patches on 0.48 $\mu m/pix$ the lowest resulting in a score of respectively 0.83 and 0.70. The highest sensitivity was 0.92 and achieved by a network trained on 256x256 patches with a resolution of 0.48 $\mu m/pix$. Two networks with a small patch size and low resolution (64x64@0.48 and 64x64@0.96) scored with 0.79 the lowest sensitivity. Despite having the lowest sensitivity, the network trained with patches of 64x64@0.96 scored the highest specificity of 0.94. The network trained on patches with 256x256@1.92 scored the lowest specificity of 0.84.

4.3 Experiment 3: Hard mining

Four networks were trained and tested and resulted in Figure 4.3 with network 2.5 added as an comparison. Network 3.3 and 3.4 both resulted in a higher F1-score compared to network 3.1 and 3.2. The addition of hard mining showed no effects on the performance when comparing network 3.3 with 2.5. However, when looking at the percentage of false classified pixels for the annotated classes 'healthy epithelium' and 'non-tumor' in Table 4.1 network 3.3 shows a higher percentage of false positives in both classes. In particular the 'healthy epithelium' class is causing more false positives and causes the specificity to be lower than network 2.5. In general, changing the ratio mainly results in more or less false positives in the healthy epithelium class.

Table 4.1: The percentage of false classified pixels for the annotated classes 'Healthy Epithelium' (HE) and 'Non-tumor' (NT). Pixels of both classes formed the total amount of false negatives(FN).

Experiment	Healthy Epithelium (% false of total HE)	Non-tumor (% false of total NT)
Network 2.5	0.05	0.08
Network 3.1	0.10	0.11
Network 3.2	0.12	0.12
Network 3.3	0.18	0.10
Network 3.4	0.14	0.10

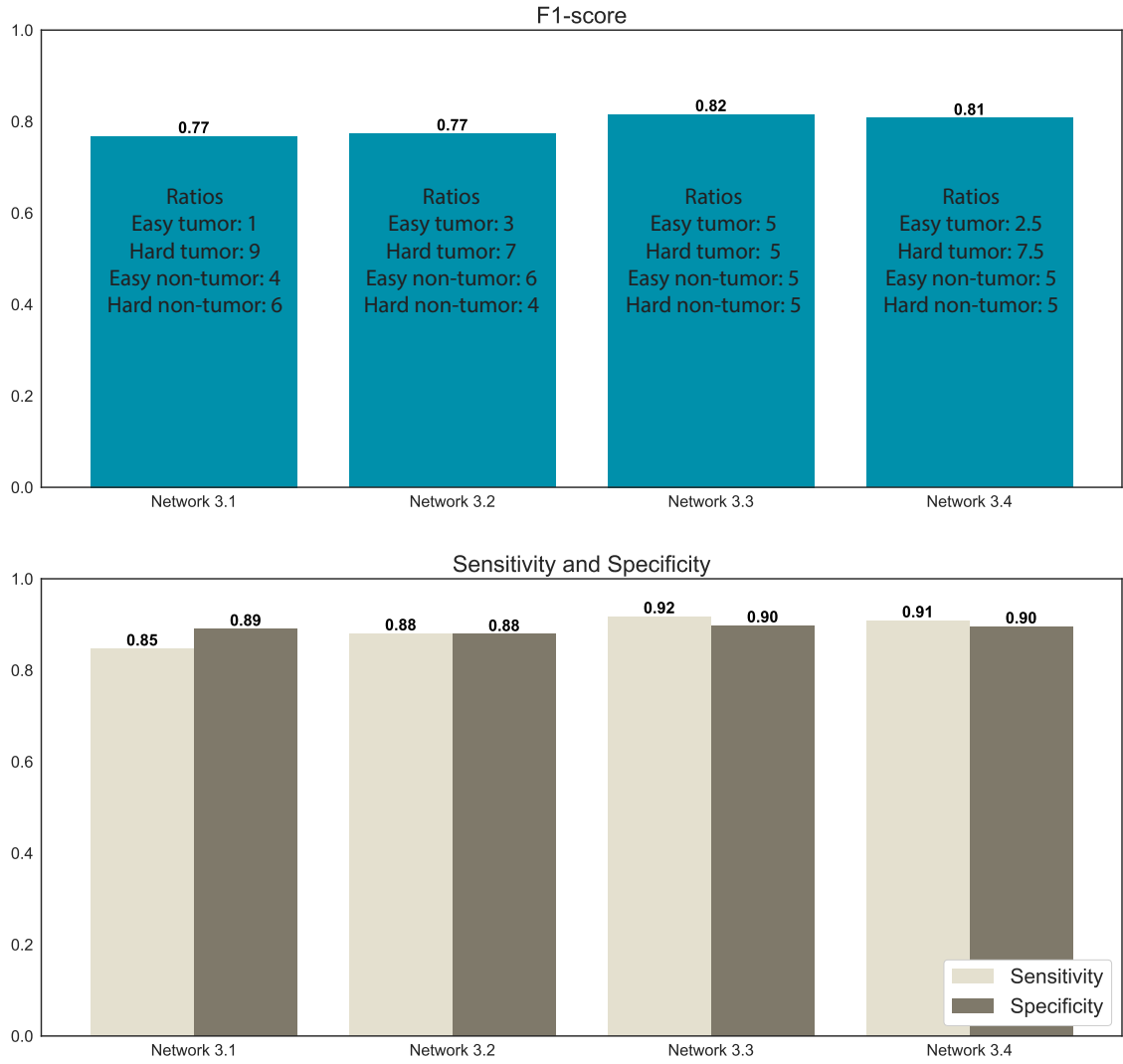


Figure 4.3: Results of networks trained on differently assembled minibatches.

4.4 Experiment 4: U-net

Three networks were successfully trained and were compared with the performance of network 2.5. The results in Figure 4.4 show that the U-net with the highest F1-score ($F1 = 0.81$) has comparable performance as network 2.5. However, the U-net architecture in network 4.2 did not enhance the tumor segmenting performance compared to the best performing FCNN. the two other trained U-nets show far less performance with an F1-score of respectively 0.74 and 0.72. Also, the balance between the sensitivity and the specificity is more out of balance than network 2.5 and network 4.2. Both network 4.3 and 4.4 have a high sensitivity of 0.99 and 0.97, while both their specificity is 0.79.

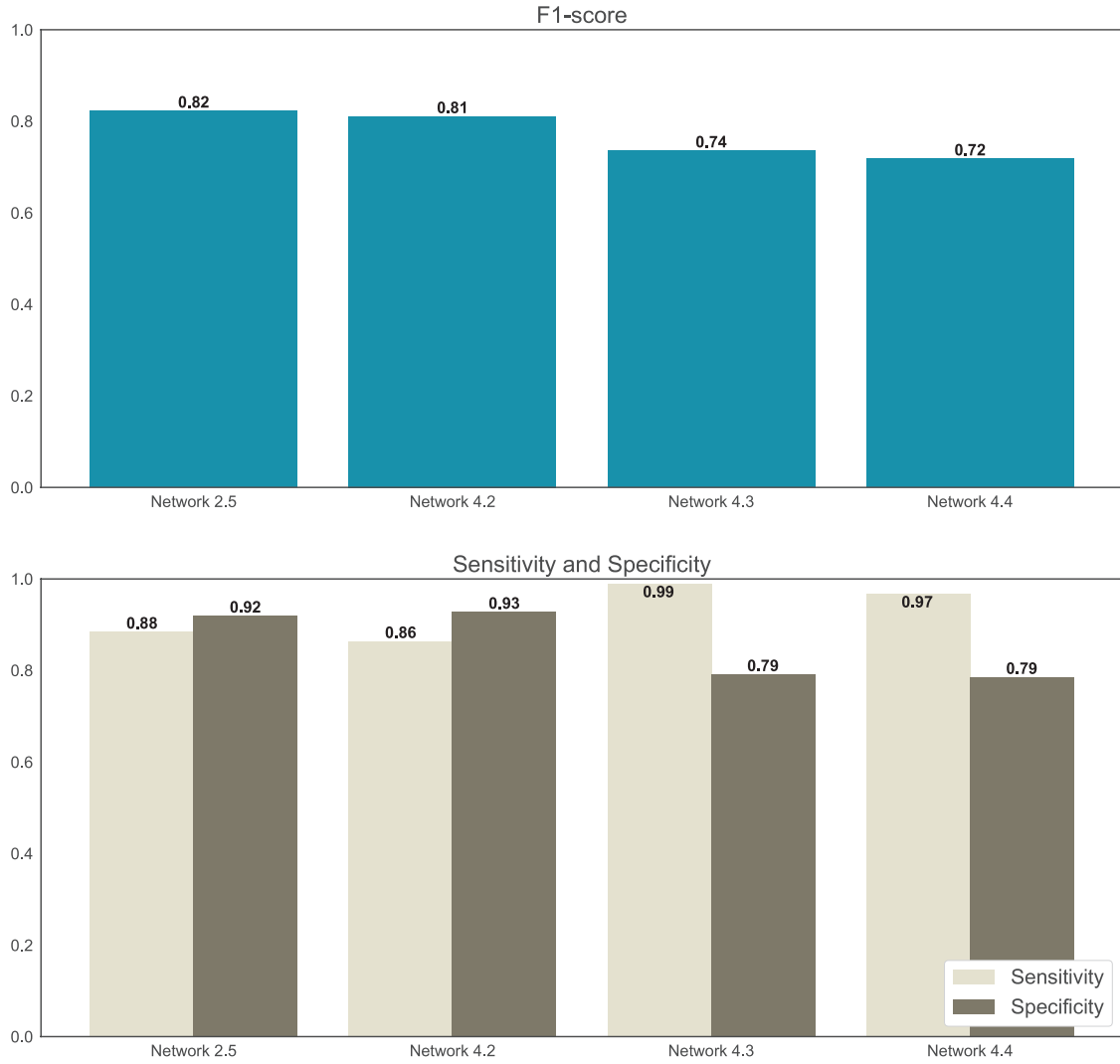


Figure 4.4: Results of networks trained with a U-net architecture

4.5 Experiment 5: Ensemble

Six networks were trained with different ensemble techniques. The best performing model was the 'model averaging ensemble' with a F1-score of 0.83. Together with the 'majority-vote ensemble' (F1 = 0.82) these models performed better than the 'stacked model ensembles'. The stacked models that were trained on different sets of data show near equal results with similar F1-score, sensitivity and specificity performance. The biggest discrepancy between the four 'stacked models' and the two other ensemble techniques was the sensitivity. The 'model averaging' and 'majority-vote' ensemble scored a higher sensitivity of respectively 0.94 and 0.92 compared to the four stacked ensembles with 0.81, 0.83, 0.82 and 0.81.

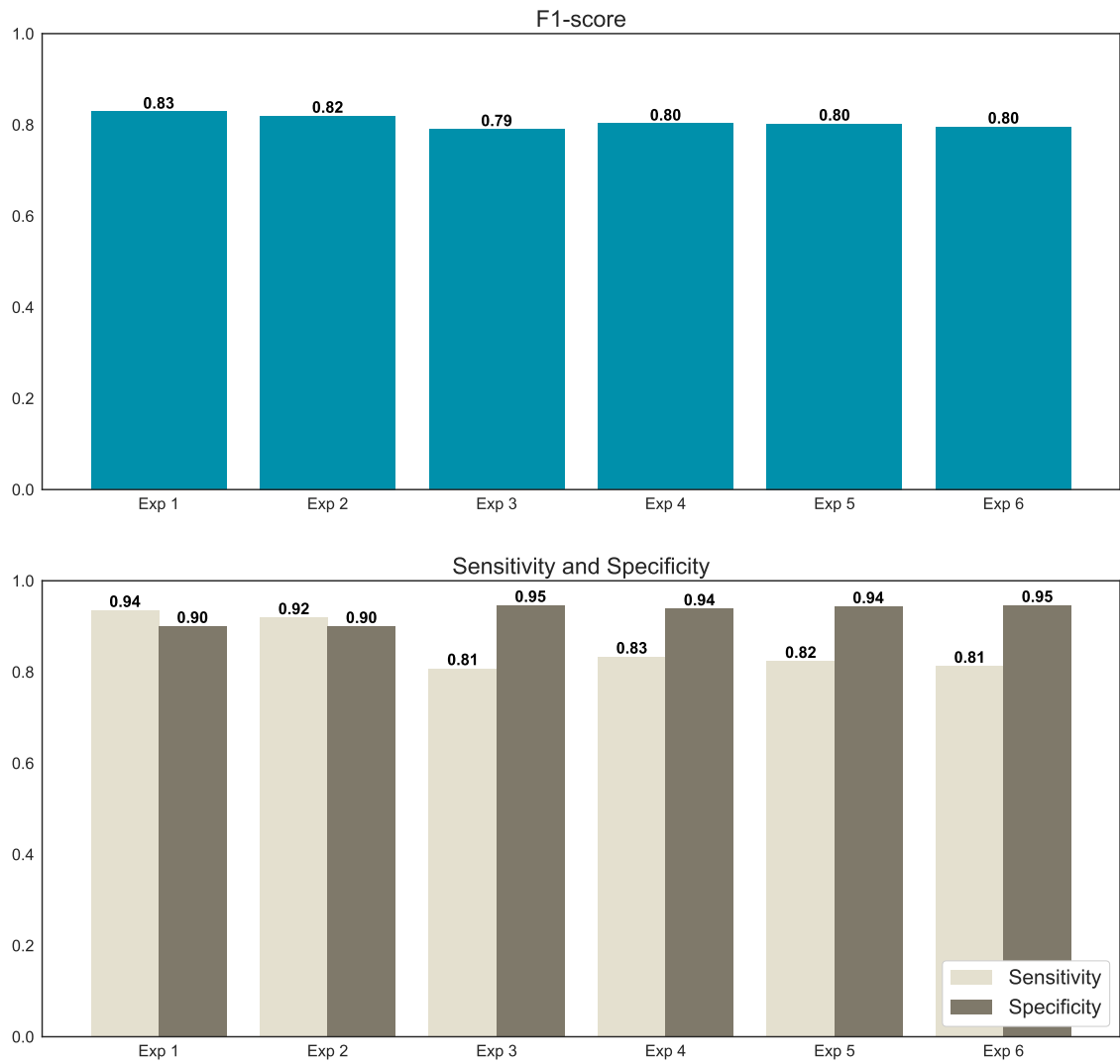


Figure 4.5: Results of different type of ensemble methods. Experiment 1 is a 'model averaging ensemble', experiment a 'majority-vote ensemble' and experiment 3 to 6 were 'stacked models' that were trained on different sets of data.

5. Discussion

We set out to segment tumor in TNBC, we managed to get promising results by doing series of experiments. The first and second experiment showed that training a FCNN on patches with the DAPI and CK8-18 data channels together with a resolution of $0.96 \mu m/pix$ and patch size of 128×128 resulted in the highest performance, however the addition of hard mining did not led to an increase in performance. A different DL architecture U-net showed similar results compared to that of the FCNN, despite its multiple successes over FCNN in the literature. Creating a 'model averaging ensemble' led to a slight increase in performance by combining the predictions of the best performing FCNN, U-net and FCNN with hard mining. This led to an overall F1-score performance of 0.83 for segmenting tumor in TNBC immunohistochemical multiplex images. Next the results will be discussed in the same order as they were presented in the resulted, followed by three overarching limitations of the research.

The first experiment showed that it is possible to train networks with different sets of data channels. It was for a network beneficial to train on tumor markers. When these data channels were added to the patches F1-scores and specificity significantly increased. Obviously these tumor markers add spatial information about the tumor location and therefore enhance the specificity and the F1-score. However, the CK8-18 markers lacks specificity and therefore also stains healthy epithelium. This led to an increase of false positives of the network, because the networks were not able to discriminate between healthy and tumor tissue (Figure 5.1.B). Adding more data channels did not help to overcome this problem. In fact, when adding too much data channels a decrease in sensitivity was observed. This is caused by the fact that adding more data channels introduces more noise which could lead to overfitting.

The results from the second experiment show that selecting patches with a patch size of 64×64 is not beneficial to train on. For all the resolutions the F1 performance is lower than 0.80 with the lowest score being 0.70. The same holds for the resolution $1.92 \mu m/pix$ where for all patch sizes the performance is low compared to other trained networks. The best performing settings were resolutions of 0.48 or $0.96 \mu m/pix$ and patch sizes of at least 128×128 .

The hard mining showed no improvement in the performance of the FCNN. The addition of FP regions did not allow to learn more complex features. This could be explained by the fact that too few were added to the dataset, which caused the network to see examples repeatedly. Also by only enriching the training set samples could be included that were not available in the validation set. A solution could be to apply transfer learning after training a network. This next learning stage can be conducted by only using FN and FP as dataset in order to fine-tune the weights. However, the ability to learn tasks in a sequential fashion could lead to catastrophic forgetting.

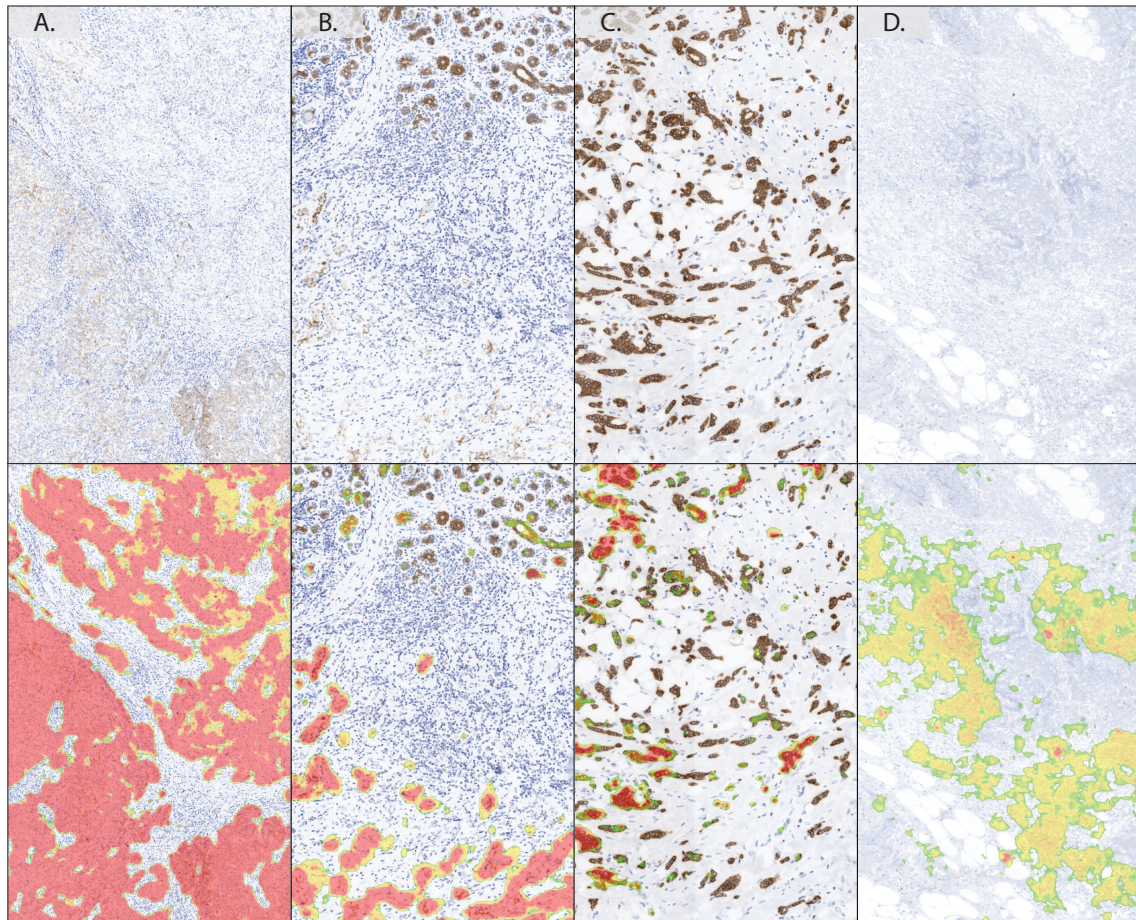


Figure 5.1: Predictions of network 6.1 the 'model averaging ensemble'. The tumor prediction is color coded with green being a small likelihood and red a high likelihood. A.) Sample of tissue where the prediction was highly accurate in segmenting the tumor, even when lacking the CK8-18 marker as seen in the top of the tissue. However, the absence of the marker causes the prediction to be less certain in some regions. B.) In the bottom a very sparse tumor area that still is segmented highly accurate. Even very small regions are noted by the network. The top showing healthy tissue were some regions being labelled as tumor regions with a low likelihood. The difference between healthy epithelium and tumor tissue is hard, due to the high variety of both. C.) Type of tumor tissue that was not in the training set. The very small tumor regions are not detected. All networks trained on this slide scored bad. D.) A piece of tissue with no tumor tissue. The DAPI channel contains more data than only the nuclei. Autofluorescence or other marker data 'leaking' into the DAPI channel could lead to a 'ghost prediction' of the network.

U-net showed similar results compared to that of the FCNN, both network 2.5 and the best performing U-net (network 4.2). Next to that, the U-net trained on only the DAPI datachannel achieved performance similar to the FCNN network (network 1.1). This could indicate that the dataset is limiting both architectures learning new features. Since the encoder part of the U-net is comparable with a normal FCNN, is it the decoder part what makes the U-net stand out, however the training data lacked fully annotated patches, which is beneficial for U-nets. This limited the network from learning higher level spatial features between non-tumor and tumor borders. In addition to that, the

minibatches consisted , due to memory limitations, out of 1 patch limiting the network to only learn from a small fraction from the patch, since it was sparse annotated. A downside for using U-net for this tissue type, is that fully annotating sparse tumor regions is extremely laborious and therefore infeasible (such as the type of tissue in Figure 5.1.C).

The minor performance increase of the 'model averaging ensemble' indicate that the combined networks did not differ widely. This can be explained by the fact that since it was two class segmentation problem, the prediction masks of the FCNN did not hold a wide range of likelihoods. The softmax output was either almost 1 for tumor, and 0 for non-tumor. Therefore, taking the average of three similar performing models with high tumor likelihoods, will result in a nearly similar model. In this way it will also act the same as a majority vote model, which showed nearly identical performance results. Introducing a third class for the 'healthy epithelium' will reduce this effect, since tumor and healthy epithelium look very similar. Also, by adding a third class allows to train a network mainly on tumor and healthy epithelium, which both can be added to an ensemble, introducing a wider range of likelihoods.

Finally three overarching limitations will be discussed. The first was that during the acquisition and pre-processing multiple problems occurred. Visual assessment pointed out that the CD45 staining failed in most of the slides. Instead of staining lymphocytes, a fluorescent signal was observed between all the cells. Also the tumor marker CK8-18, which was included in most of the trained networks did not adhere to all tumor tissue. In hugely differentiated slides a lack of CK8-18 signal was observed. This can be explained by the fact that types of tumor tend to lose there antigenicity, causing CK8-18 antibodies to be unable to bind (Figure 5.1.A). Therefore it is recommended to investigate the possibilities of a suitable substitute for CK8-18, such as CK19 (Dawson et al., 2009).

The second is that apart from problems that occurred during the staining procedure, the stain unmixing also led to data impurities. It was observed that certain fluorescent signals were leaked into the wrong datachannel (Figure 5.1.D). For example, Ki67 stained cells were observed in the CK8-18 channel. This is caused by the fact that the emitting wavelength of both fluorophores only differs 20nm. Therefore there fluorescent 'fingerprint' used for unmixing looks similar and can cause unmixing problems. Next to that, only three autofluorescence measurement were used to form the autofluorescence signal for unmixing thousands of multi-spectral tiles. Data leaking into the wrong channels can seriously hamper the performance of the networks, especially when tumor markers leak into lymphocyte marker channels. The problems that occurred during the stain unmixing could be avoided by doing the tumor segmentation on the raw spectral data. Similar strategies are already applied in the geospatial field (Yue et al., 2015).

The third limitation was that most of the networks achieved F1-scores of around 0.80, indicating that it was not possible for the networks to learn more complex features. Especially, rare and challenging annotations were sparse and due to the TNBC the tissue showed huge morphologically variations (see variety Figure 5.1.A-D). After inspection of the training and validation set, wrong annotations were found and types of tumor and non-tumor were not available in both sets. Therefore the validation set was not representative for some types of tissues. It is hard to compare the performance of the test set, because no other datasets exist and no reports of similar approaches in literature. It is therefore recommended that the human performance should be measured, before enlarging the dataset. This could give more insights whether the networks are approaching human performance and are getting the most out of the given

dataset.

However, the ensemble was the best performing network and was able to segment tissue successfully with a F1-score of 0.83. A next step can now be made to combine lymphocyte detection and tumor segmentation in search of powerful prognostic markers for TNBC. Preliminary results showed that the current trained network can be applied on other TSA stained tissues such as lymph nodes. Next to that, the network (network 1.1) trained on only the DAPI channel showed promising results for future work. By only training a network on the DAPI channel could result in tumor segmentation on any breast tissue stained with DAPI, since DAPI is a routine counter-stain for nuclei when using fluorescence microscopy. The data was acquired using a multi-spectral camera setup, future networks could be trained using hyperspectral image segmentation, allowing to bypass any unmixing steps that were involved in this work. In this way the network is also able to learn not only spatial but also spectral features. Further, the method of creating artificial IHC whole-slide images based on a target IHC slide shows promising results for multiple applications such as enriching lymphocyte stained breast cancer datasets.

6. Conclusion

The aim of this thesis was to investigate the performance of convolutional networks to segment tumor regions in triple negative breast cancer whole-slide multiplex IHC slides. Multiple experiments were conducted to investigate and maximize the performance. The data used for training was investigated and it can be concluded that using the DAPI and CK8-18 data channels together with a resolution of $0.96 \mu m/pix$ and patch size of 128×128 resulted in the highest performance.

Enriching the dataset with hard mining had no positive effects on the performance. Using a different DL architecture U-net resulted in similar results compared to that of a FCNN. A 'model averaging ensemble' resulted in the highest segmentation performance with a F1 score of 0.83.

It can be concluded that fully convolutional networks are able to segment tumor regions in triple negative and holds true for both FCNN and U-net architectures. We discussed the improvements that can be made to enhance the segmentation performance. Next to this we discussed future applications and it can be concluded that the current top performing network is applicable for a step closer to the overarching aim of this research, namely extracting powerful prognostic information from intra- and peritumoral lymphocytes.

Bibliography

- Abdi, H., Valentin, D., and Edelman, B. (1999). *Neural networks*, chapter Neural networks. Quantitative applications in the social sciences ; no. 07-0124. Sage Publications, Thousand Oaks, Calif.
- Adams, S., Gray, R. J., Demaria, S., Goldstein, L., Perez, E. A., Shulman, L. N., Martino, S., Wang, M., Jones, V. E., Saphner, T. J., et al. (2014). Prognostic value of tumor-infiltrating lymphocytes in triple-negative breast cancers from two phase iii randomized adjuvant breast cancer trials: Ecog 2197 and ecog 1199. *Journal of clinical oncology*, 32(27):2959–2966.
- Badrinarayanan, V., Kendall, A., and Cipolla, R. (2015). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561*.
- Bándi, P., van de Loo, R., Intezar, M., Geijs, D., Ciompi, F., van Ginneken, B., van der Laak, J., and Litjens, G. (2017). Comparison of different methods for tissue segmentation in histopathological whole-slide images. In *Biomedical Imaging (ISBI 2017), 2017 IEEE 14th International Symposium on*, pages 591–595. IEEE.
- Berry, D. A., Cirrincione, C., Henderson, I. C., Citron, M. L., Budman, D. R., Goldstein, L. J., Martino, S., Perez, E. A., Muss, H. B., Norton, L., et al. (2006). Estrogen-receptor status and outcomes of modern chemotherapy for patients with node-positive breast cancer. *Jama*, 295(14):1658–1667.
- Bray, F., Ferlay, J., Soerjomataram, I., Siegel, R. L., Torre, L. A., and Jemal, A. (2018). Global cancer statistics 2018: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA: a cancer journal for clinicians*, 68(6):394–424.
- Brouckaert, O., Wildiers, H., Floris, G., and Neven, P. (2012). Update on triple-negative breast cancer: prognosis and management strategies. *International journal of women’s health*, 4:511.
- Ciresan, D., Giusti, A., Gambardella, L. M., and Schmidhuber, J. (2012). Deep neural networks segment neuronal membranes in electron microscopy images. In *Advances in neural information processing systems*, pages 2843–2851.
- Cireşan, D. C., Giusti, A., Gambardella, L. M., and Schmidhuber, J. (2013). Mitosis detection in breast cancer histology images with deep neural networks. In *International Conference on Medical Image Computing and Computer-assisted Intervention*, pages 411–418. Springer.

- Clutter, M. R., Heffner, G. C., Krutzik, P. O., Sachen, K. L., and Nolan, G. P. (2010). Tyramide signal amplification for analysis of kinase activity by intracellular flow cytometry. *Cytometry Part A*, 77(11):1020–1031.
- Dawson, S., Provenzano, E., and Caldas, C. (2009). Triple negative breast cancers: clinical and prognostic implications. *European journal of cancer*, 45:27–40.
- de Bel, T., Hermesen, M., Smeets, B., Hilbrands, L., van der Laak, J., and Litjens, G. (2018). Automatic segmentation of histopathological slides of renal tissue using deep learning. In *Medical Imaging 2018: Digital Pathology*, volume 10581, page 1058112. International Society for Optics and Photonics.
- Dent, R., Trudeau, M., Pritchard, K. I., Hanna, W. M., Kahn, H. K., Sawka, C. A., Lickley, L. A., Rawlinson, E., Sun, P., and Narod, S. A. (2007). Triple-negative breast cancer: clinical features and patterns of recurrence. *Clinical cancer research*, 13(15):4429–4434.
- Dixon, A. R., Bathany, C., Tsuei, M., White, J., Barald, K. F., and Takayama, S. (2015). Recent developments in multiplexing techniques for immunohistochemistry. *Expert review of molecular diagnostics*, 15(9):1171–1186.
- Dumoulin, V. and Visin, F. (2016). A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*.
- Edge, S. B. and Compton, C. C. (2010). The american joint committee on cancer: the 7th edition of the ajcc cancer staging manual and the future of tnm. *Annals of surgical oncology*, 17(6):1471–1474.
- Fontenot, J. D., Rasmussen, J. P., Williams, L. M., Dooley, J. L., Farr, A. G., and Rudensky, A. Y. (2005). Regulatory t cell lineage specification by the forkhead transcription factor foxp3. *Immunity*, 22(3):329–341.
- Fridman, W. H., Pages, F., Sautes-Fridman, C., and Galon, J. (2012). The immune contexture in human tumours: impact on clinical outcome. *Nature reviews. Cancer*, 12(4):298.
- Geijs, D. J., Intezar, M., van der Laak, J. A., and Litjens, G. J. (2018). Automatic color unmixing of ihc stained whole slide images. In *Medical Imaging 2018: Digital Pathology*, volume 10581, page 105810L. International Society for Optics and Photonics.
- Hayes, D. F. (2010). Tumor node metastasis (tnm) staging classification for breast cancer.
- He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034.
- Hori, S., Nomura, T., and Sakaguchi, S. (2003). Control of regulatory t cell development by the transcription factor foxp3. *Science*, 299(5609):1057–1061.
- Karpathy, A. (2018). Cs231n convolutional neural networks for visual recognition.
- Katz, S. J. and Morrow, M. (2013). Addressing overtreatment in breast cancer: The doctors’ dilemma. *Cancer*, 119(20):3584–3588.

- Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.
- Krose, B. and Van der Smagt, P. (1996). An introduction to neural networks. *University of Amsterdam*.
- LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436.
- Levenson, R., Hoyt, C., and Gossage, K. (2009). Classifying image features. US Patent 7,555,155.
- Levenson, R., Hoyt, C. C., and Gossage, K. W. (2014). Classifying image features. US Patent 8,639,043.
- Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., van der Laak, J. A., van Ginneken, B., and Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. *arXiv preprint arXiv:1702.05747*.
- Litjens, G., Sánchez, C. I., Timofeeva, N., Hermesen, M., Nagtegaal, I., Kovacs, I., Hulsbergen-Van De Kaa, C., Bult, P., Van Ginneken, B., and Van Der Laak, J. (2016). Deep learning as a tool for increased accuracy and efficiency of histopathological diagnosis. *Scientific reports*, 6:26286.
- Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440.
- Macchia, I., Urbani, F., and Proietti, E. (2013). Immune monitoring in cancer vaccine clinical trials: critical issues of functional flow cytometry-based assays. *BioMed research international*, 2013.
- Mahmoud, S. M., Paish, E. C., Powe, D. G., Macmillan, R. D., Grainge, M. J., Lee, A. H., Ellis, I. O., and Green, A. R. (2011). Tumor-infiltrating cd8+ lymphocytes predict clinical outcome in breast cancer. *Journal of clinical oncology*, 29(15):1949–1955.
- Moeskops, P., Wolterink, J. M., van der Velden, B. H., Gilhuijs, K. G., Leiner, T., Viergever, M. A., and Išgum, I. (2016). Deep learning for multi-task medical image segmentation in multiple modalities. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 478–486. Springer.
- Qiu, X., Zhang, L., Ren, Y., Suganthan, P. N., and Amaratunga, G. (2014). Ensemble deep learning for regression and time series forecasting. In *Computational Intelligence in Ensemble Learning (CIEL), 2014 IEEE Symposium on*, pages 1–6. IEEE.
- Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer.
- Shen, D., Wu, G., and Suk, H.-I. (2017). Deep learning in medical image analysis. *Annual review of biomedical engineering*, 19:221–248.
- Singletary, S. E., Allred, C., Ashley, P., Bassett, L. W., Berry, D., Bland, K. I., Borgen, P. I., Clark, G., Edge, S. B., Hayes, D. F., et al. (2002). Revision of the american joint committee on cancer staging system for breast cancer. *Journal of clinical oncology*, 20(17):3628–3636.

- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.
- Wolpert, D. H. (1992). Stacked generalization. *Neural networks*, 5(2):241–259.
- Yue, J., Zhao, W., Mao, S., and Liu, H. (2015). Spectral–spatial classification of hyper-spectral images using deep convolutional neural networks. *Remote Sensing Letters*, 6(6):468–477.

