M.Sc. Thesis
Master of Science in Engineering

DTU Compute
Department of Applied Mathematics and Computer Science

Deep learning based detection of breast cancer in hematoxylin and eosin stained histopathology images

Jeppe Thagaard (s123456)

Kongens Lyngby 2017

DTU Compute
Department of Applied Mathematics and Computer Science
Technical University of Denmark

Matematiktorvet
Building 303B
2800 Kongens Lyngby, Denmark
Phone +45 4525 3031
compute@compute.dtu.dk
www.compute.dtu.dk

Abstract

The last few years of advances in deep learning for visual recognition have started to gain interest from healthcare applications. The field of histopathology is no exception after the rapid growth of digital pathology and availability of digital giga-pixel histopathology images. This enables new possibilities of assisting the assessment of breast cancer with deep learning-based image analysis.

The general focus of this thesis is to use deep learning to detect breast cancer in hematoxylin and eosin (H&E) stained histopathology images. However, if digital pathology wants to take advantage of the large amount of histopathology datasets with deep learning, there is a need for new methods. In this project, a novel approach to automatically generate labelled H&E datasets using image registration of differently stained serial tissue sections is presented. By using this method, a deep learning-based algorithm is developed to recognize tumor regions that can be used for precise quantification of immunohistochemical markers. The method produce results similar to an industry-leading IHC-approach from Visiopharm A/S on a small test set but requires further validation.

Determining lymph node involvement for breast cancer patients is another aspect of breast cancer that constitutes an integral part of the treatment decision. In clinical practice, this task is usually performed by expert pathologists using H&E stained tissue sections, but looking through entire giga-pixel histopathology images can be very difficult and time consuming task. As part of the IEEE International Symposium on Biomedical Imaging (ISBI) 2017 Grand Challenge CAMELYON17, the thesis presents a deep learning-based algorithm to automatically detect and classify breast cancer metastases, providing crucial information for the prognosis and treatment decision. The method qualified for the challenge workshop in Melbourne, Australia with a weighted kappa value of 0.8172 on the challenge test set (100 patients), showing very good agreement between the algorithm and the human pathologist. It was ranked 5th out of 23 qualifying teams of international research groups and commercial teams, with a marginal score difference to the winner of the CAMELYON17 competition.

Lastly, a H&E stain specific data augmentation scheme is developed that improve the accuracy of deep learning cancer detection with 3-6 percentage point by forcing models to disregard color variability which is irrelevant to the classification.

Resumé (Danish)

De sidste års fremskridt i deep learning indenfor visuel genkendelse er begyndt at få interesse fra sundhedssektoren. Histopatologi er ingen undtagelse efter den stigende brug af digital patologi og tilgængeligheden af giga-pixel histopathologiske billeder af vævsprøver. Denne tilgængelighed åbner op for nye muligheder for at assisterere undersøgelserne af brystkræft med deep learning baseret billedeanalyse.

Dette speciales generelle fokus er at bruge deep learning til at detektere brystkræft i hæmatoxylin og eosin (H&E) farvede histopatologiske billeder. Men der er behov for nye metoder, hvis digital patologi skal kunne udnytte mængden af data i histopatologiske billeder. I dette projekt præsenteres der en ny metode til automatisk annoterering af H&E farvede datasæt ved at bruge billedregistrering af forskellig farvede serielle vævssnit. Ved at bruge denne metode, udvikles der en deep learning baseret algoritme til genkendelse af tumorregioner, der kan bruges til præcis kvantificering af immunhistokemiske (IHC) biomarkører. Denne metode opnår resultater tæt på industri-ledende IHC metode fra Visiopharm A/S på et lille test datasæt, men den kræver yderligere validering.

Afklaring vedrørende mulig kræftspredning til lymfeknuder er et andet vigtigt aspekt ved brystkræft, der har stor betydning for behandlingsforløbet. I klinisk praksis bliver denne opgave udført af patologer i H&E farvede vævsprøver, men denne undersøgelse kan være meget svær at udføre, ligesom den er meget tidskrævende. Som en del af IEEE International Symposium on Biomedical Imaging (ISBI) 2017 Grand Challenge CAMELYON17 præsenterer dette speciale en deep learning baseret algortime til automatisk detektion og klassificering af brystkræft metastaser, der tilbyder vigtig infomation til prognose og behandlingsforløbet. Metoden kvalificerede sig til konkurrencens workshop i Melbourne, Australien med en vægtet kappa score på 0.8172 på konkurrencens test data (100 patienter), hvilket viser en rigtig god overens stemmelse mellem algoritmen og patologens vurdering. Metoden var rangeret nummer 5 i et felt med 23 kvalificerede hold bestående af internationale forskningsgrupper og kommercielle virksomheder med en marginal score-forskel ifht. vinderen af CAMELYON17 konkurrencen.

Som en del af specialet er der udviklet en data augmenteringsmetode, der forbedrer nøjagtigheden af deep learning detektion af kræft med 3-5 procentpoint ved at tvinge modellerne til at se bort fra den farve-variation, der er irrelevant for klassifikationen.

Preface

This master thesis was prepared at the department of Applied Mathematics and Computer Science at the Technical University of Denmark in fulfillment of the requirements for acquiring a Master of Science degree in Engineering (Medicine & Technology).

The thesis deals with deep learning for computational pathology to improve breast cancer diagnostics where we investigate a new approach to create labelled hematoxylin and eosin stained datasets. Furthermore, we present a deep learning-based algorithm for detecting and staging lymph node metastases in breast cancer.

The project was carried out in close collaboration with Visiopharm A/S, Hørsholm, Denmark and supervised by Anders Bjorholm Dahl, Associate Professor at Section for Image Analysis and Computer Graphics (IACG), Søren Hauberg, Associate Professor at Section for Cognitive Systems (CogSys) and Thomas Ebstrup, Software Development Manager at Visiopharm A/S. Furthermore, the project was assisted by Dr. Eva Balslev, MD, Chief Pathologist at Department of Pathology, Herlev University Hospital, Denmark.

Kongens Lyngby, June 30, 2017

Jeppe Thagaard (s123456)

Acknowledgements

I would like to acknowledge everyone who has supported the work of this thesis with special thanks to Anders Dahl for his supervision, feedback, and trust in my abilities to independently work on my ideas. I also want to thank Søren Hauberg for his indispensable discussions and his willingness to listen to and consult about technical issues throughout this thesis.

Additionally, I would like to thank the entire team at Visiopharm, especially Thomas Ebstrup for his supervision and amazing technical insights into image processing for histopathology images. Furthermore, I sincerely appreciate the trust and support from Johan Doré and Michael Grunkin when I proposed the ideas of this thesis. I also want acknowledge the rest of my amazing colleagues for their interest and discussions.

Further more, I would like to acknowledge Dr. Eva Balslev for providing histopathology images and her work on manual tumor estimations.

Lastly, I thank my family, especially Cecilia for her indispensable support during my entire education the last 5 years. Without you, this thesis would never have been possible.

Abbrevations

Adam ..... Adam adaptive moment estimation
APP ..... Image analysis application in VIS
AUC ..... Area under the curve
BCa ..... Breast cancer
CE IVD ..... European Union (EU) validation for in vitro diagnostic use
CM ..... Classical Momentum
CNN ..... Convolutional neural network
DCIS ..... Ductal carcinoma in situ
DL ..... Deep learning
DNN ..... Deep neural network
DSC ..... Dice Similarity Coefficient
E ..... Eosin
ER ..... Estrogen receptor
FC ..... Fully-connected layer
FCN ..... Fully-convolutional network
FOV ..... Field-of-view
GPU ..... Grapichal processing unit
H ..... Hematoxylin
H&E ..... Hematoxylin and eosin
HER2 ..... Human epidermal growth factor
HR ..... Hormone receptor
IA ..... Image analysis
IDC ..... Invasive ductal carcinoma
IEEE ..... Institute of Electrical and Electronics Engineers
IHC ..... Immunohistochemistry
IHS ..... Intensity-Hue-Saturation
ILSVRC ..... ImageNet Large Scale Visual Recognition Challenge
ISBI ..... International Symposium on Biomedical Imaging
ITC ..... Isolated tumor cell

LCIS ..... Lobular carcinoa in situ
ML ..... Machine Learning
MLP ..... Multi-layer perceptron
NAG ..... Nesterov's Accelerated Gradient
OD ..... Optical density
PCK ..... Pan cytokreatin
PI ..... Proliferation Index
PR ..... Progesterone receptor
RF ..... Random forest
RGB ..... Red-Green-Blue
RMSProp ..... Root mean square propagation
ROC ..... Receiver operating characteristics
ROI ..... Region-of-interest
SGD ..... Stochastic gradient decent
URS ..... Uniform random sampling
VDS ..... Virtual double staining
VIS ..... Visiopharm Integrator System
WSI ..... Whole slide image

Contents

CHAPTER 1

Introduction

Breast cancer (BCa) is the most common cancer disease for women with 1.7 million new cases worldwide in 2012 [1]. Denmark had the second highest incidence of BCa per capita compared to other western countries in 2012 [1] with 4700 estimated new cases each year [2]. The disease arises when cells change and grow out of control in the lobules or ducts in the breast tissue. Regardless of how BCa is detected; from self-palpation to national screening programs, all suspicious cases must go through extensive microscopic analysis of tissue biopsies before patients are given a definitive medical diagnosis. This histopathological evaluation is not only instrumental for diagnosis but also for treatment planning and prognosis of each single patient. Each of these evaluations require tedious microscopic assessment of multiple tissue and/or cell characteristics based on an increasing number of histopathology tests. These tests range from percentage tumor evaluation in tissue samples to molecular profiling of BCa. With high incidence rates and an increasing number of highly important tasks, there is a need for new methods that can assist pathologists and decrease their workload. In this thesis, we take aim at multiple areas to improve automated hematoxylin and eosin (H&E)-based BCa diagnostics.

For BCa, pathologists traditionally perform manual assessment of tumor extent and annotations of tumor regions in H&E-stained samples which can then be used for further analysis. Whilst these tasks are considered easy for an experienced pathologist, they are tedious to perform and studies have shown that subjective assessment introduce potential high inter-observer variability [3]. These observations have several implications. Most obviously, the tissue-based analytics performed after H&E assessment are potentially corrupted by e.g. flawed annotations. Another implication of this relates to recent advances using deep learning algorithms on histopathology images, in which models are usually developed using supervised training on manually annotated images. Clearly, the before-mentioned implication also apply here but another aspect is that deep learning algorithms usually require large amount of labelled training data. Obtaining large training data sets require the very costly expert annotations that may be subjective with high inter-observer variability. These observations motivated the original aim in this thesis:

Investigate a novel approach to train deep learning algorithms in digital pathology using image registration of differently stained serial tissue sections.

We aim to develop our approach as proof-of-concept and use it to train deep learning-

based algorithm for automatic identification of cancer tissue[1]. We strive to validate our approach by comparing to manual expert evaluations and existing CE-IVD[2] immunohistochemistry (IHC)-based methods.

Another crucial aspect of BCa diagnosis is characterization of metastatic involvement of lymph nodes because it is one of the most important prognostic factors. Therefore, it is vital to quickly determine which lymph node(s) if any, the cancer has spread to. Pathologists usually examine metastases in the lymph nodes by measuring the size of the tumor and/or counting the number of tumor cells. Thus, they manually categorize the type of metastases which they use in a prognostic score. This task has a high clinical relevance but requires large amounts of reading time for pathologists. Therefore, an automated solution would hold great promise to reduce the workload of the pathologists and at the same time reduce the subjectivity in diagnosis. This leads to the extended aim in this thesis:

Develop a deep learning approach to automatically detect, classify, and stage BCa metastases in lymph nodes.

As part of this aim, we participated in the IEEE International Symposium on Biomedical Imaging (ISBI) 2017 Grand Challenge CAMELYON17 [4] and submitted our method for external review. We qualified for the ISBI conference workshop in Melbourne, Australia where we presented our first and only submission so far ranking 5th out of 23 qualifying teams.

1.1 Outline of thesis

The thesis is structured as follows. Chapter 2 introduces the histopathology background of the thesis together with aspects of digital pathology that is necessary to understand the concepts presented later. Chapter 3 is dedicated to deep learning as we give a thorough description of the theory behind convolutional neural networks together with related methodology. In Chapter 4 and Chapter 5, we present two independent studies covering the original and extended aim respectively. In Chapter 4, we propose a novel approach to train deep learning algorithms in digital pathology using image registration of differently stained serial tissue sections. All methods and results relating to this part are included in this chapter. In Chapter 5, we present a deep learning approach to automatically detect and classify BCa metastases to stage the disease across multiple images from the same patient. Here, we also show results from our submission to IEEE ISBI 2017 Grand Challenge CAMELYON17. In Chapter 6, we discuss topics relevant for both previous chapters and describe future directions of this thesis. In Chapter 7, we draw the conclusions of the thesis.

[1] In this thesis, the word *we* refers to the single author of this thesis, and not to multiple authors.
[2] CE-IVD refers to European Union (EU) validation for in vitro diagnostic use.

CHAPTER 2

Background

In this chapter, we describe some of the background topics behind the research and methods of this thesis. We briefly introduce BCa from a histopathology perspective and its lymphatic metastization before we review the aspects of digital pathology relevant to the thesis.

2.1 Breast cancer

The disease arises when cells change and grow out of control, typically in milk-producing glands, lobules, or in the connections between lobules and the nipple called ducts, see figure 2.1. BCa is not a single type of tumor but covers a whole range of neoplasms in the breast tissue where the histopathology of BCa refers to how the tissue and cells differs from normal tissue and within different breast cancers. When a BCa patient undergoes histopathology evaluation, the pathologists evaluate many different factors such as malignancy, invasive or noninvasive (in situ), cell origin, molecular sub-type etc. Most suspected cases are benign neoplasms, i.e. non-cancerous tumors that are non-life threatening but for malignant cases, tumors are divided into in situ and invasive [5].

The two major types of noninvasive breast cancers are ductal carcinoma in situ (DCIS) and lobular carcinoma in situ (LCIS) that originate from epithelial cells in the ducts and lobules, respectively [5]. All cancers originating from epithelial cells are termed carcinomas. These noninvasive cancers arise when abnormal cells replace the normal epithelial cells, but have not grown beyond the layer of cells where they originated.

Invasive breast cancers are also categorized based on their origin. Invasive ductal carcinoma (IDC) is the most common, accounting for 75% of all malignant BCa [6]. It arises when abnormal cells break through the wall of the duct and continue to infiltrate the surrounding tissue. There exist up to 21 distinct histological sub-types of invasive BCa with at least four different molecular sub-types, see table 2.1. These types vary in their response to treatment due to differences in their gene expressions [5]. Therefore, it is very important to quickly and precisely determine the molecular sub-type. Commonly, this is performed by evaluating specific immunohistochemical (IHC)-biomarkers. IHC is described later in section 2.2.2.2 but IHC is the most commonly used method in BCa to measure the presence or absence of

specific hormone receptors (HR), e.g. estrogen (ER) or progesterone (PR), excess levels of growth-promoting protein (HER2) or a large proportion of actively dividing cells e.g. measured by Ki67 nuclei-expression [5]. In this thesis, we only consider the biomarker Ki67 further described in section 2.2.2.2 and 2.2.4. When evaluating these biomarkers, it is crucial that pathologists only perform the assessment inside tumor regions and disregard surrounding connective tissue called stroma. Currently, there are clinically approved image analysis (IA) which can perform the before-mentioned quantitative measurements by using multiple IHC-biomarkers as described later in section 2.2.4. In this thesis, we investigate if it possible to measure the IHC expression using a cheaper alternative to current available methods without losing any precision.

Invasive BCa is a severe disease with 5- and 10-year survival rates of 80% and 60%, respectively [6]. The prognosis of breast cancer patients is strongly influenced by the stage of the disease. Medical professionals use the TNM classification system to describe the extent of the disease when it is first diagnosed. The system encapsulates the tumor size and spread to adjacent tissue (T), the spread to closely located lymph nodes (N) and the presence of distant metastases (M). These three factors determine a stage between 0 and IV, with in situ as stage 0, early stage invasive cancer as stage I and the most advanced disease stage at stage IV [5]. As part of this thesis, we focus on the histological assessment of lymph node metastases as it is an essential part of TNM classification.

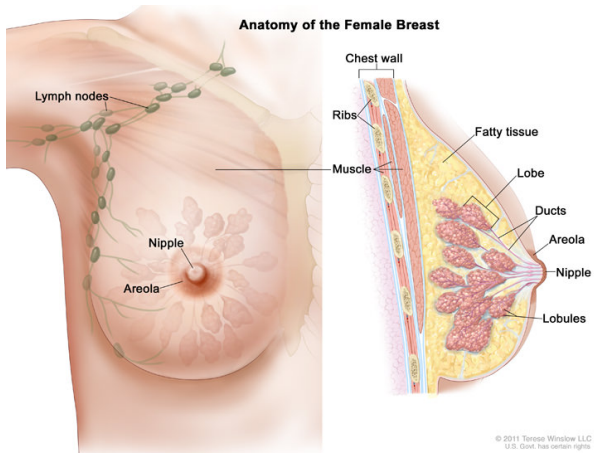


Figure 2.1: Female breast anatomy. BCa most commonly originates from the lobules or ducts (right figure). The lymphatic system drains and transport fluids called lymph, in the body through a network of vessels and lymph nodes (green structure left figure). It drains fluids from the breast tissue, hence metastases are most likely to appear in the lymph nodes of the axilla. Figure from [7].

Name	Molecular expression	Short description
Luminal A	HR+/HER2-	Most common, ER+ and/or PR+, but HER2-, slow growing, good prognosis due responsiveness to hormone treatment.
Triple negative	HR-/HER2-	Do not express ER, PR or HER2, poorer short-term prognosis due no responsive to hormone treatment.
Luminal B	HR+/HER2+	Most common, ER+ and/or PR+ and highly proliferating (Ki67+) or HER2+, more aggressive than luminal A, but respond to hormone treatment.
HER2-enriched	HR-/HER2+	Most uncommon, ER- and/or PR-, but HER2+, more aggressive than other types, but respond to HER2-targeted treatments.

Table 2.1: Molecular sub-types of BCa [5]. This thesis focuses only on Ki67 expression which can be used to discriminate between Luminal A (low proliferating cancer) and Luminal A (high proliferating cancer).

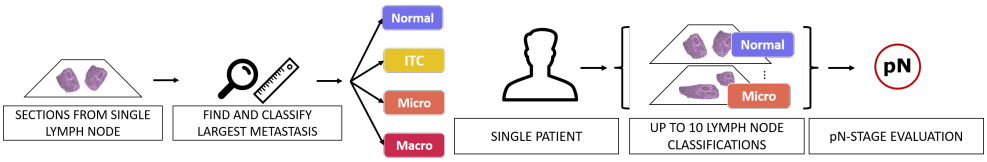
2.1.1 Lymph node metastases

Like many cancers, BCa metastasizes when cancer cells break free from the primary tumor and spread through the lymphatic or blood system to other organs of the body. BCa can spread directly to locally underlying muscular tissue and skin, to closely located lymph nodes in the axilla or hematogenous to lungs, bones, liver and central nervous system. Lymph node metastases in the axilla are common (20-50%) and is one of the most important prognostic factors. Prognosis is poorer when cancer has spread to the lymph nodes which is why lymph nodes are surgically removed and examined microscopically [4].

As introduced earlier, pathologists manually examine metastases in lymph nodes to classify them into macro metastases, micro metastases or isolated tumor cells (ITC), see table 2.2. Another aspect is that there are many normal samples that do not include any tumor cells. Assessment of multiple lymph nodes per patient are then combined into a pathological N-stage (pN-stage) which is part of the TNM-system. The workflow of manual pN-stage evaluation is shown in figure 2.2.

Metastasis	Tumor Cells	Size (Major Axis)
ITC	$\leq$ 200 cells	or $<$ 0.2 mm.
Micro	$>$ 200 cells	or $>$ 0.2 mm but $\leq$ 2.0 mm.
Macro	-	$>$ 2.0 mm.

Table 2.2: Official definition of lymph node metastases: These definitions are frequently up for discussion in the pathology society, but are the current guidelines for metastases categorization. ITC can be single tumor cells or a cluster of tumor cells.



(a) Classification of single section **(b)** pN-stage evaluation

Figure 2.2: Histological assessment of lymph node metastases. In (a), the pathologist manually classify a section based on the largest metastases. In (b), up to 10 sections are combined into a pN-stage. The pN-staging system depends on national guidelines that usually follows the definitions of the American Joint Commitee on Cancer (AJCC).

2.2 Digital pathology

Digital pathology is a relatively new and disruptive technology. It refers to the use of computer technology to view, analyze and manage digitized anatomic pathology tissue samples. This thesis is based in the field of analyzing pathology images using IA often referred to as computational pathology. But an introductory knowledge of key terms in digital pathology is essential for understanding our methods and is therefore presented below.

2.2.1 Preparation of tissue biopsies for histology examination

We shortly describe the preparation process from biopsy extraction to the image data. This process is important to know in order to develop successful IA algorithms for histopathology. The process includes six main steps; **biopsy extraction, fixation, embedding, sectioning, staining and digitization**, see figure 2.3. **Biopsy extraction:** A tissue specimen is removed surgically or by needle biopsy and is send to the pathology laboratory. Here, the specimen is examined macroscopically and suspicious samples are send to further analysis.

Fixation: Performed to preserve the state and structure of cells and tissue in the biological specimen. The most common type is formalin fixation, where the sample is kept in a formaldehyde compound for period of time (>24 h) [8], but other fixation methods exist, e.g. frozen sections etc.

Embedding: After fixation, the specimen undergoes dehydration and the water is replaced by paraffin mixture that solidifies the tissue [8]. The goal of embedding is to harden the sample enough to allow cutting of very thin sections.

Sectioning: The paraffin block is sectioned on a microtome, where a thin knife

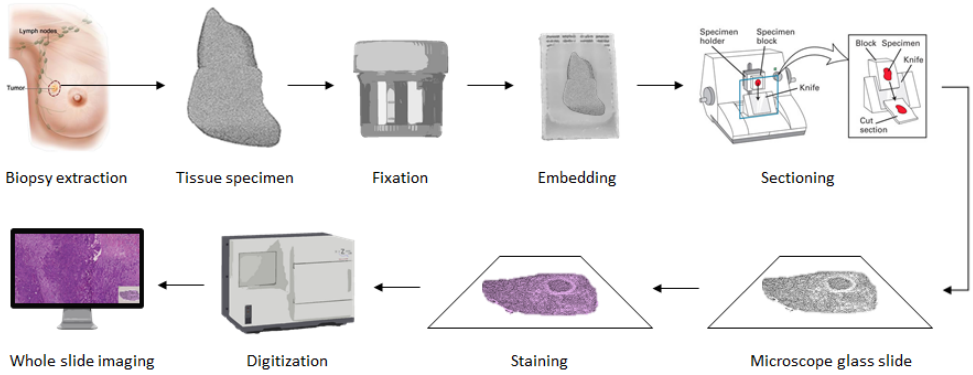


Figure 2.3: Major steps in the preparation. Many of the steps are controlled by or involve human interaction. These steps emphasize how many factors that might affect the final image data; tissue folding or missing tissue from sectioning, insufficient staining, out-of-focus regions from digitization etc.

cuts the specimen into very thin sections, usually between 3-5μm. A bio-analyst controls the microtome manually, and mounts the tissue sections onto a microscope glass slide. At this point in the process, the tissue is colorless and a pathologist would not be able to interpret any information from the sample.

Staining: Refers to the use of chemical compounds to add contrast to otherwise transparent tissue sections [8]. See section 2.2.2 for more details.

Digitization: Traditionally, pathologists use bright-field microscopes to view and interpret the stains of the sections. Recently, research labs and clinical laboratories started to adopt whole slide scanners that capture images of the glass slide at high magnification and stitches these together tile-wise in a digitized format called a whole slide image (WSI), see section 2.2.3 below. WSI scanners enable pathologists to view sections on digital monitors, but also open up for the use of computer-aided detection and diagnosis systems (CADe and CADx, respectively). In contrast to medical imaging in radiology, there is no standardized format so each vendor have their own format. Hence, data originating from different scanners have potential differences in their acquisition settings e.g. γ-correction, saturation etc.

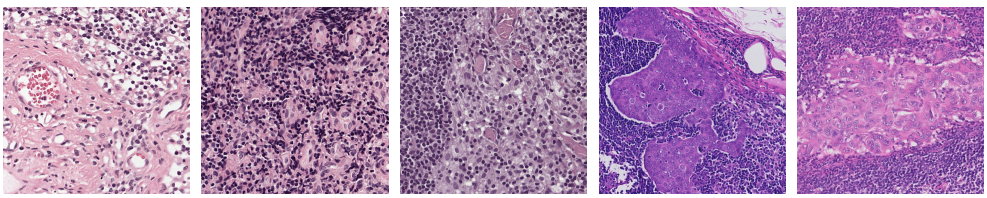
2.2.2 Staining

There are many different staining techniques in histopathology but generally they are either specific or non-specific. Specific staining targets certain molecules in cells or tissue, whereas most cells are similar in colors for non-specific stains. Here, we elaborate on the most common non-specific staining called H&E and the basics of

immunohistochemistry (IHC), a methodology of specific staining. Images from both techniques are used in this thesis.

2.2.2.1 Hematoxylin and Eosin

Hematoxylin and eosin (H&E) staining is the most common staining procedure in histopathology and has been used for at least a century to create contrast between various tissue components [9]. It is comprised of two dyes; hematoxylin and eosin. Hematoxylin targets basophilic cell components and therefore colors cell nuclei chromatins dark blue or purple. Eosin colors acidophilic components red or pink such as positively charged cytoplasmic proteins. H&E is considered the golden standard for many diagnostic examinations performed on conventional microscopy as skilled pathologists can from this interpret morphological characteristics and distribution patterns of cells. However, there is no standardized staining protocol and the procedure consists of multiple steps. The specific protocol is usually designed by the individual pathology laboratory resulting in inter-site variability on color and contrast intensities, see figure 2.4. Moreover, H&E staining is also subject to intra-centre variability as the chemical solutions are influenced by multiple factors such as intensity decay, over-washing etc. Hence, this problem has hindered IA on H&E sections as current methods have difficulties capturing the variation of color intensities. Stain normalization methods [10, 11, 12, 13] have been shown to improve H&E-based IA applications but are also subject to criticism as they may introduce non-biological information, artifacts etc. In this thesis, the primary focus is H&E stained sections as this staining is very cheap compared to the specific IHC staining, which makes it attractive for routine diagnostics.

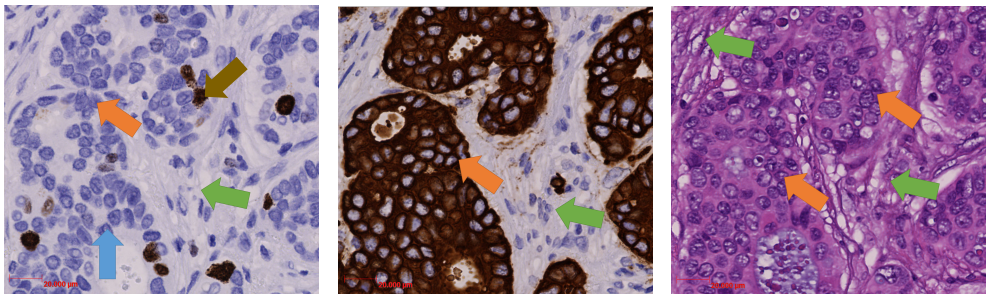


(a) Lab 1 **(b)** Lab 2 **(c)** Lab 3 **(d)** Lab 4 **(e)** Lab 5

Figure 2.4: H&E stain variability. H&E stained lymph node tissue from five different pathology laboratories. The stain colors range from bright red in (a) and (b) to very pink as shown in (d) and (e) but there are also color intensities in between as shown in (c). How to overcome this variability while still considering the information of the color nuances within a section, is one of the challenges in this thesis.

2.2.2.2 Immunohistochemistry

IHC is an advanced histological staining method that utilizes antibody-antigen specificity to selectively image specific cell components or biomarkers, e.g. receptor-proteins. Usually, the antibody is tagged with an enzyme that catalyze specific coloring [2], making it possible to capture the signal using a regular bright-field microscope. This method is used in clinical diagnostics and histology research as more and more IHC markers are developed. For example, pan cytokeratin (PCK) is an IHC-marker used to detect epithelial tissue in carcinomas, creating high contrast between stroma and tumor tissue. Some applications use multiple IHC-markers on the same tissue section to obtain complimentary information from two or more IHC-signals. This process is called physical double staining, but overlapping chromogens/signals make precise assessment difficult for IA [14, 8]. The IHC biomarkers for Ki67, ER and PR are important for BCa diagnosis as described in table 2.1. In this thesis, we work with two IHC stains; Ki67 and PCK. See figure 2.5a and figure 2.5b for more detailed description of these compared to H&E stained tissue.



(a) Ki67 staining **(b)** PCK staining **(c)** H&E staining

Figure 2.5: Difference of stains. The orange arrows shows tumor cells while green arrows show stroma. In (a), Ki67 is expressed in the cell nuclei in all active phases of the cell division [15] and is used as biomarker for cell proliferation. The IHC staining colors proliferating cells dark brown (brown arrow in (a)), while non-proliferating cells are blue due to the hematoxylin counterstain (blue arrow in (a)). The fraction between Ki67 positive and negative cells is termed the Proliferation Index, and can be used to discriminate between the molecular subtypes luminal A and B [16]. This staining cannot easily distinguish between tumor and stroma regions, hence we need to combine it with another staining. In (b), PCK is a diagnostic IHC staining that colors cytoplasmic keratin-filaments (cytokeratins) in epithelial carcinomas brown (orange arrow in (b)). It is well suited to distinguish epithelial tissue from stroma (green arrow in (b)) but is not useful to discriminate between invasive and in situ carcinomas as both types reacts with the IHC. Similar to Ki67, hematoxylin is used as counterstain, hence the blue-ish color of nuclei or non-malignant tissue. In (c), the same tissue is H&E stained where we can discriminate between tumor regions and stroma by the cell morphology and not the color intensity like in the PCK staining.

2.2.3 Whole slide images

Whole slide images (WSI) is the specific file structure of digital histopathology images. This type of image data is different from regular static images as the scanners digitize glass slide at a microscopic level (up to 160 nm per pixels) so the image resolution is comparable to an optical microscope. Therefore, a WSI is a collection of tiled-images, stitched together generating large data files. The pathologists expect rapid image-access as a microscope (zooming, panning etc.), so WSIs are typically stored in a multi-resolution pyramid structure, see figure 2.6. A typical WSI is scanned at 40× (0.25 μm/pixel) or 20× (0.50 μm/pixel) and is approximately 200000 x 100000 pixels on the highest magnification level with 3 byte RGB pixel format. This gives rise to huge data files ranging from 0.5 GB to 5 GB per WSI depending on the vendor format. Therefore, image processing and IA on WSIs require different approaches than other fields of medical imaging, e.g. tile-by-tile processing etc.

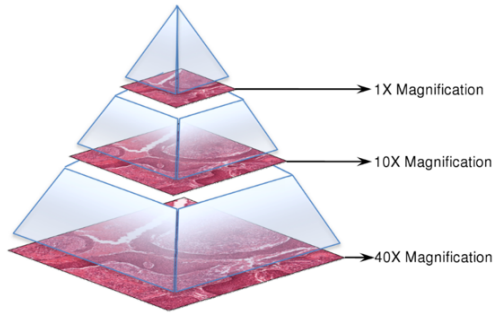


Figure 2.6: WSI resolution pyramid. Each level of the pyramid is referred to by its magnification level similar to the lenses of an optical microscope. Figure from [4].

2.2.4 Virtual Double Staining (VDS) and CE IVD APPs

Virtual Double Staining (VDS) is a computational technique [17] that provides an alternative to physical double staining. Instead of using two IHC-markers on a single tissue section, each IHC-marker is applied independently to adjacent tissue sections, i.e. the chromogens cannot physically overlap. By registering two WSIs using a feature-based registration algorithm [18], we obtain the same level of information as physical double staining and at the same time enable IA to be applied more precisely. The VDS-concept is shown in figure 2.7 and has been used in digital assessment of IHC quantification of nuclei biomarkers (Ki67 and ER) in BCa [16, 19].

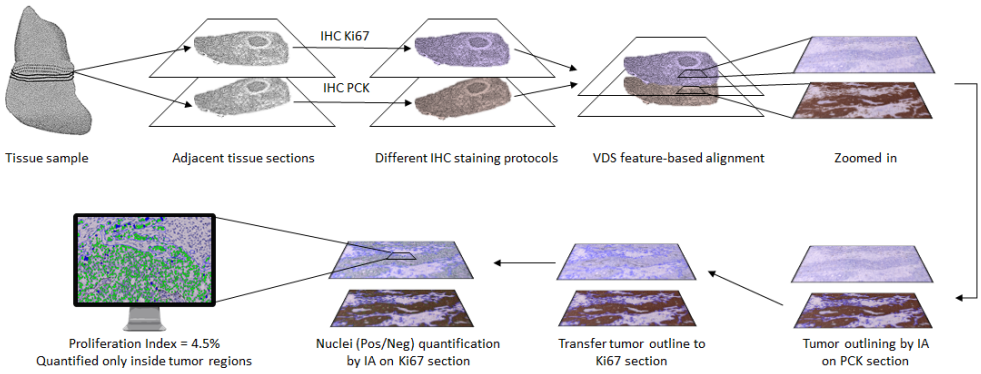


Figure 2.7: Major steps in the VDS workflow for Ki67 quantification. Here, two tissue sections are first aligned where epithelial cells are segmented in the PCK section using a PCK VDS APP (section 2.2.4.2). The results are transfered to the Ki67 section to ensure that the nuclei-expression is only quantified inside tumor regions by the Ki67 APP (section 2.2.4.1).

The registration algorithm is developed by Visiopharm A/S and we will sometimes refer to it as *alignment* in this thesis. The VDS alignment performs image registration [18] of adjacent tissue sections using multi-scale point-matching and local deformations via thin-plate splines [20]. It is implemented in the Tissuealign^{TM} workflow where a user performs either fully-automatic, semi-automatic or manual registration of entire WSIs. Semi-automatic registration includes initial manual registration via 3-4 manually placed landmarks followed by the automatic alignment. This is only necessary when fully-automatic alignment fails, e.g. for serial sections that are rotated or flipped very differently.

We use this alignment technique for our experiments in chapter 4 together with two existing CE-IVD approved IA algorithms described below. The existing algorithms are executed in Visiopharm Integrator System (VIS) (Visiopharm A/S, Hørsholm, Denmark), which is professional software for digital pathology.

2.2.4.1 Ki67 APP

We use this existing algorithm [21] to validate our results in chapter 4. The Ki67 APP uses pixel threshold classification to discriminate between Ki67 positive and negative nuclei, i.e. segmentation and classification is based on the stain intensities. It calculates the Proliferation Index (PI), i.e. the fraction of Ki67 positive and the total number of cells across an entire tissue section. The PI is used to recommend if a patient should receive chemotherapy or not based on the grading system in table 2.3. The APP is CE-IVD approved for diagnostic use when it is combined with the PCK APP.

	Benign	Borderline	Malignant
Ki67 PI	<20%	20-50%	>50%
Chemotherapy	No	Yes	Yes

Table 2.3: Grading of cancer using PI cut-off: The Ki67 Proliferation Index (PI) is used to recommend chemotherapy for BCa patients with fast growing tumors (PI>20%). These definitions are recommended by Danish Breast Cancer Cooperative Group (DBCG) for BCa pathology [22]. We use these cut-off values for validation purposes in this thesis.

2.2.4.2 PCK APP

This existing algorithm [23] is used extensively in chapter 4. The PCK APP uses a Bayesian pixel classifier to segment epithelial regions from stroma regions, i.e. it depends on the stain intensities of the individual pixels. The regions are typically used in combination with other nuclei detection algorithms such as the Ki67 APP. When used on tissue sections with BCa, the epithelial regions are assumed to be the malignant tumor regions but the APP cannot discriminate between invasive and non-invasive tumor regions.

2.3 Related work

Several promising studies have applied deep learning to histopathology. Most of the related studies introduced here are published before the start of this thesis. It should be noted that there has been published a large amount of related studies during the period of the thesis, e.g. [24, 25, 26, 27, 28, 29].

The earliest successful application of deep learning to histopathology images was Ciresan et al.[30] that use convolutional neural networks to recognize and count mitotic figures for BCa grading. A similar approach by Wang et al.[31] combined deep learning and feature-engineering to improve the results on this task. For primary BCa, Cruz et al. [32] presented one of the first deep learning-based models for automatic detection and visual analysis of IDC tissue regions in WSIs. Litjens et al. [33] showed how relatively simple convolutional neural networks could successfully discriminate between benign and malign prostate cancer tissue, where they also showed promise on detection of lymph node metastases. In last year's CAMELYON16 Challenge on detection of lymph node metastases, all top-performing teams used convolutional neural networks [34]. The winning team Wang et al. [35] showed how deep learning-based solutions have potential to outperform pathologists on screening for lymph node metastases in H&E stained sections. Regarding the use of IHC information when applying deep learning to histopathology, Turkki et al. [36] used IHC stained images side-by-side with H&E stained images to guide the manual annotations and showed how deep learning can be applied to quantify immune cell infiltration in H&E stained images in BCa samples. To the best of our knowledge, there are no published

papers on automatically using IHC information to label H&E or other stained serial sections as basis for training deep learning algorithms.

CHAPTER 3

Theory and Methods

In this chapter, we introduce feed-forward networks and other theoretical concepts of supervised deep learning before we describe convolutional neural networks; the type of models, we use in this thesis to learn histological difference between cancer and normal tissue. These sections are aimed at readers with only minor knowledge on deep learning and convolutional neural networks. After the theoretical sections, we propose a new data augmentation scheme specific to histopathology applications.

3.1 Deep learning

Deep learning (DL) has quickly become a very popular choice for analyzing images. Evolving from previous work on artificial neural networks, it now covers an entire sub-field of machine learning (ML) and spans multiple domains. Generally, DL can be defined as hierarchical feature learning that aims to learn complicated concepts by building them out of simpler ones [37]. Most importantly, it differs from traditional feature engineering and ML as we allow the computer to not only learn the mapping from features to the output but also the features that fit the problem at hand.

3.1.1 Feed-forward neural networks

Originating from early work on biological learning [38, 39] and the original perceptron [40], feed-forward neural networks or multilayer perceptrons (MLPs) are the foundation of many modern DL models. These networks are well-known in the ML community and are typically used to approximate a function that maps an input $\mathbf{x}$ to an output $\mathbf{y}$ by learning the parameters $\boldsymbol{\theta} \in \{\mathbf{w}, \mathbf{b}\}$, where $\mathbf{w}$ and $\mathbf{b}$ are called the weight and bias term. The quintessential expression of MLPs is the activation $\mathbf{a}$ of linear combination of the inputs:

$$\mathbf{a} = \sigma(\mathbf{w}^T\mathbf{x} + \mathbf{b}) \tag{3.1}$$

where $\sigma(\cdot)$ is the element-wise activation function that introduces non-linearities to the operation $\mathbf{z} = \mathbf{w}^T\mathbf{x} + \mathbf{b}$ [41]. In fully-connected layers, each unit in a layer is connected to all units in the previous layer through the parameter weights. Feed-forward networks are composed of multiple functions like equation 3.1 stacked together in

sequential layers l such that:

$$\mathbf{y}(\mathbf{x}, \boldsymbol{\theta}) = \sigma(\mathbf{w}^{l^T}\sigma(\mathbf{w}^{l-1^T}...) + \mathbf{b}^l) \tag{3.2}$$

where each layer transforms the input from the previous layer into more abstract representations. The last output layer is an operator that fits a specific task e.g. a logistic sigmoid function for binary classifications. The optimal parameters of a network can be obtained by minimizing a loss function. Usually, neural networks approximate a probability distribution $\mathbf{y} \sim p(\mathbf{y}|\mathbf{x}, \theta)$, so we can train these models using maximum likelihood [37]. The appropriate loss function must fit the objective of the model but for many problems, we simply use the negative log-likelihood described as the cross-entropy between the training data and the model distribution [37]:

$$E(\boldsymbol{\theta}) = -\mathbb{E}_{\mathbf{x},\mathbf{y}\sim\hat{p}_{\text{train}}} \log\, p_{\text{model}}(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) \tag{3.3}$$

By minimizing the loss function w.r.t. to the parameters, we maximize the likelihood function of the model. Therefore, we must select a loss function to represent the output of the model. Generally, for a training set of independent observations N, we express the loss of a network as:

$$E(\boldsymbol{\theta}) = \sum_{n=1}^{N} E_n(\boldsymbol{\theta}; \hat{\mathbf{y}}_n, \mathbf{y}_n), \qquad \{(\mathbf{x}_n, \hat{\mathbf{y}}_n); n \in [1, ..., N]\} \tag{3.4}$$

where $(\mathbf{x}_n, \hat{\mathbf{y}}_n)$ is the n-th example of the N input-target relations [42], i.e. we use a loss function to compute the error between the output $\mathbf{y}_n$ and the target $\hat{\mathbf{y}}_n$. In this thesis, we only use neural networks on binary classification problems. Therefore, we only have one target $\hat{y}$ such that $\hat{y} = 1$ denotes the positive class and $\hat{y} = 0$ denotes the negative class. As mentioned earlier, our output layer is the logistic sigmoid operator:

$$y(\mathbf{x}, \boldsymbol{\theta}) = \sigma(\mathbf{z}) = \frac{1}{1 + e^{-\mathbf{z}}}, \qquad 0 \leq y(\mathbf{x}, \boldsymbol{\theta}) \leq 1 \tag{3.5}$$

where $y(\mathbf{x}, \boldsymbol{\theta})$ is the conditional probability $p(y = 1|\mathbf{x})$ and $p(\hat{y} = 0|\mathbf{x})$ is equal to $1 - p(\hat{y} = 1|\mathbf{x})$ [42]. The conditional probability of targets given the inputs is then modelled as a Bernoulli distribution:

$$p(\hat{y}|\mathbf{x}, \boldsymbol{\theta}) = y(\mathbf{x}, \boldsymbol{\theta})^{\hat{y}}[1 - y(\mathbf{x}, \boldsymbol{\theta})]^{1-\hat{y}} \tag{3.6}$$

We can now write our loss function as the cross-entropy assuming $y_n = y(\mathbf{x}, \boldsymbol{\theta})$):

$$E(\boldsymbol{\theta}) = -\sum_{n=1}^{N} (\hat{y}_n \ \ln y_n + (1 - \hat{y}_n) \ \ln(1 - y_n)) \tag{3.7}$$

Many loss functions and output layer operators exist for various problems, e.g. linear output layer and sum-of-squares loss function for regression or softmax output layer

with the multiclass cross-entropy for multiclass classification [42]. As mentioned earlier, a neural network consists of many layers with different operations. The number of layers gives the depth of the model, hence the terminology 'deep' neural networks (DNNs). Our general goal is to find the parameters $\boldsymbol{\theta}$ which minimize the selected loss function E. As many other modelling applications, we turn to numerical parameter optimization via gradient information, i.e. finding the weights and biases of each layer in the parameter space such that:

$$\nabla E(\boldsymbol{\theta}) = 0 \tag{3.8}$$

In this thesis, we use gradient information to perform iterative updates of the parameters until we reach a stationary point (local minima or saddle-point). This requires evaluation of loss function derivatives w.r.t. the parameters. To solve this, we turn to the backpropagation algorithm [43] and the modular design of feed-forward neural networks before we describe the actual optimization procedure in section 3.1.3.

3.1.2 Modular design of networks

The natural next step is to present layers in a neural network as modular components. This simplifies the understanding of the mathematics and the explanation of how we can train large networks. A layer is for example the linear transformation of the inputs to an output but any differentiable operation is a candidate for a layer. Now, consider a network comprised of L sequential layers, where the L-th layer is the loss function $E(\boldsymbol{\theta})$ for our network[1]. Generally, we send information forwards and backwards through the network to first evaluate $\frac{dE}{d\boldsymbol{\theta}}$ and then use the derivatives to perform the parameter update [42]. We describe the first stage by considering each layer individually. We represent the output of layer l as:

$$\mathbf{z}^{l+1} = \mathbf{f}^l(\mathbf{z}^l; \boldsymbol{\theta}^l) \tag{3.9}$$

where $\mathbf{z}^l$ and $\mathbf{z}^{l+1}$ are the input and output matrices, respectively and $\mathbf{f}^l$ is the layer operation [44] e.g. equation 3.1. This takes care of the forward propagation of information all the way to the loss function E.

[1] In this section, we write $E(\boldsymbol{\theta})$ as E to simplify the equations.

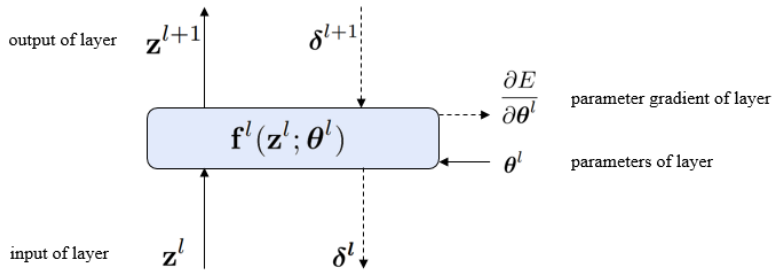


Figure 3.1: Layers as modules. Neural networks propagate information forward and backwards through layers. A layer performs a mathematical operation on the input, e.g. a linear combination. We can define a layer by its forward operation $\mathbf{f}^l(\mathbf{z}^l;\boldsymbol{\theta}^l)$ and the derivatives of the loss w.r.t. input $\mathbf{z}^l$ and parameters $\boldsymbol{\theta}^l$. Figure adapted from [44].

We obtain the backward flow of information (loss/error/cost) by calculating the derivatives of the loss w.r.t. input $\mathbf{z}^l$ and parameters $\boldsymbol{\theta}^l$. We can write the former using the chain rule for partial derivatives and equation 3.9 [44]:

$$\boldsymbol{\delta^l} := \frac{\partial E}{\partial \mathbf{z}^l} = \frac{\partial E}{\partial \mathbf{z}^{l+1}} \frac{\partial \mathbf{z}^{l+1}}{\partial \mathbf{z}^l} = \boldsymbol{\delta}^{l+1} \frac{\partial \mathbf{f}^l(\mathbf{z}^l;\boldsymbol{\theta}^l)}{\partial \mathbf{z}^l} \tag{3.10}$$

From this, we see that we only need to calculate derivative of the operator w.r.t. the input of the current layer and then reuse $\boldsymbol{\delta}^{l+1}$ from the adjacent layer. Moreover, as $\frac{\partial \mathbf{z}^{l+1}}{\partial \mathbf{z}^l}$ is the Jacobian with many inputs and many outputs, we need take the derivative for each output w.r.t. each input in turn [44]. Therefore, we can write equation 3.10 as:

$$\delta_i^l := \sum_j \frac{\partial E}{\partial z_j^{l+1}} \frac{\partial z_j^{l+1}}{\partial z_i^l} = \sum_j \delta_j^{l+1} \frac{\partial f_j^l(z^l;\theta^l)}{\partial z_i^l} \tag{3.11}$$

Similarly, we can compute the derivative of the loss w.r.t. parameters as:

$$\frac{\partial E}{\partial \boldsymbol{\theta}^l} = \frac{\partial E}{\partial \mathbf{z}^{l+1}} \frac{\partial \mathbf{z}^{l+1}}{\partial \boldsymbol{\theta}^l} = \boldsymbol{\delta}^{l+1} \frac{\partial \mathbf{f}^l(\mathbf{z}^l;\boldsymbol{\theta}^l)}{\partial \boldsymbol{\theta}^l} \tag{3.12}$$

also with recursion of information flow from $\boldsymbol{\delta}^{l+1}$ [44]. Again, we can write equation 3.12 on the same form as equation 3.11:

$$\frac{\partial E}{\partial \theta_i^l} = \sum_j \frac{\partial E}{\partial z_j^{l+1}} \frac{\partial z_j^{l+1}}{\partial \theta_i^l} = \sum_j \delta_j^{l+1} \frac{\partial f_j^l(z^l;\theta^l)}{\partial \theta_i^l} \tag{3.13}$$

where we calculate the derivative for each output w.r.t. to each parameter in turn and then sum across all outputs.

Consequently, all we need in order to define a layer are equation 3.9, 3.11 and 3.13.

Here, the essential observation is the recursion of the δ's when layers are stacked sequentially. This tells us that for a particular layer, we simply propagate δ's backwards from layers higher up in the network. This is the foundation of the backpropagation algorithm [43] and based on this, we are currently training DNNs. Luckily, efficient recursive differentiation methods are implemented into deep learning software frameworks, e.g. Theano [45] that enables us to focus on defining layers and the optimization method.

3.1.3 Training deep neural networks

Here, we describe the gradient descent optimization methods that we use in this thesis to perform parameter updates and other optimization considerations.

3.1.3.1 Gradient descent optimizers

Generally, gradient descent works by iterative updating the parameters in small steps in the direction of the negative gradient:

$$\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t - \eta \nabla E(\boldsymbol{\theta}^t) \tag{3.14}$$

where the step size of each update t is controlled by the learning rate η [37]. In practice, we use mini-batch stochastic gradient decent (SGD), i.e. for each parameter update, the loss is summed for random subset of the entire data set:

$$E(\boldsymbol{\theta}) = \sum_{n=1}^{N_{\text{mb}}} E_n(\boldsymbol{\theta}) \tag{3.15}$$

where N_{mb} is the mini-batch size. In contrast to equation 3.4, this result in much faster training but the variance of the parameter update is influenced by the learning rate and mini-batch size.

The global optimization problem of neural networks is highly non-convex where there may exist several local minima and saddle-points. There are many ways to combat this; one of them is modifying the optimizing method e.g. SGD with momentum or adaptive gradient methods. Especially the latter is currently very popular for training deep neural networks as these offer faster convergence but are, at the time of this writing, an active area of ML research [46]. We use **SGD with Nesterov's momentum (Nesterov's Accelerated Gradient (NAG))** [47] that extends SGD and classical momentum (CM) [48]. NAG is a first-order optimization method, which helps increase convergence compared to SGD [49]. We perform the parameter update as:

$$\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t + \mathbf{v}^{t+1} \tag{3.16}$$

$$\mathbf{v}^{t+1} = \mu \mathbf{v}^t - \eta \nabla E(\boldsymbol{\theta}^t + \mu \mathbf{v}^t) \tag{3.17}$$

where μ is the momentum parameter, v^{t+1} is the velocity vector that approximate the position of the next parameters by accumulating previous gradients [47]. Therefore, compared to CM, it calculates the gradient w.r.t. where the next parameters will approximately be and not w.r.t. to the current parameters. This anticipatory update prevents the optimizer from going too fast and results in increased responsiveness to gradient changes.

We also experiment with two popular adaptive gradient methods; RMSProp [50] and Adam [51]. These optimizers work by adapting the learning rate for each of the parameters during training such that larger updates are performed for infrequent parameters and smaller updates for frequent parameters [52]. **RMSProp (Root Mean Square Propagation)** keeps track of the exponentially decaying average of past squared gradients and then divide the learning rate for a parameter by this running average [50]. **Adam (Adaptive Moment Estimation)** is an extension to RMSProp as it keeps track of both the exponentially decaying average of past gradients and the exponentially decaying average of past squared gradients [51]. Hence, Adam can be considered RMSProp with momentum [53]. For the mathematical background of adaptive gradient methods, we refer to [52, 46].

3.1.3.2 Initialization of network parameters

Weight initialization refers to how the parameters are initialized and has been shown to be crucial for training deep networks with SGD [54]. Currently, state-of-the-art is the normalized **He initialization** [55] inspired by normalized **Xavier/Glorot initialization** [56]. He initialization draws random values such that:

$$\boldsymbol{\theta} \sim N(0, \sqrt{2/n}) \tag{3.18}$$

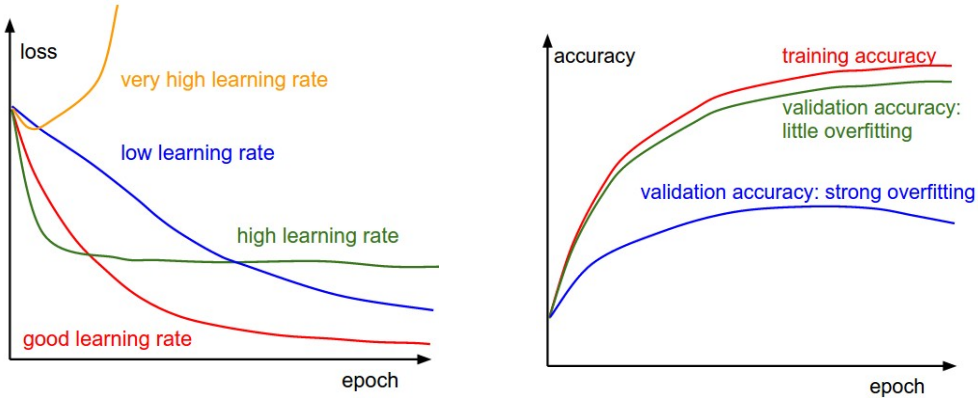
where n is the number of units in the layer. This has been proven to keep the variance of the input equal to output of a layer [56], which solves a problem of exploding variance of gradients for larger networks and modern non-linear activation functions.

During training, we are only sure to converge to a local minimum due to the non-convex nature of the loss function [42]. Therefore, we can retrain the network with a new random weight initialization and compare several local minima to find a sufficient solution. We cannot know if the solution is the global minimum but DNNs can be powerful models regardless because we can monitor the loss and accuracy during the optimization as described next.

3.1.3.3 Monitoring the learning process

An important aspect of successfully training DNNs is monitoring the learning/convergence process, sometimes referred to as babysitting the network. In this thesis, we follow the heuristics recommended by [57] on for example hyperparameter tunning

(e.g. learning rate, minibatch size) and overfitting. Most importantly, we monitor the loss and the accuracy/error rate on a training/validation split, see figure 3.2. There exist a whole range of network indicators that we can monitor e.g. the noise of the loss to tune the minibatch size or the distribution of layer activations to monitor gradient information.



(a) Monitor loss **(b)** Monitor accuracy

Figure 3.2: Babysitting learning convergence. In (a), a toy example of the influence of different learning rates. Very high learning rates will never decrease the loss as the gradient updates are overshot significantly (yellow line). High learning rates will decay the loss faster, but they get stuck and oscillate at worse values of loss (green line). In (b), the gap between the training and validation accuracy indicates the amount of overfitting. The blue line shows overfitting which indicates that the model has to much capacity, i.e. we do not have enough training data to generalize well. Both figures are from [57].

We have introduced the fundamental theory of DNNs and the methods used to train deep learning models. In the next section, we will move on to describe the special variant of feedforward neural networks that we use in this thesis. All previous sections and observations on DNNs are still valid for the next section.

3.2 Convolutional neural networks

Convolutional neural networks (CNNs) are currently the most successful type of DNNs in the field of computer vision. Originally inspired by the visual cortex of the brain [58], CNNs are special variants of MLPs with convolutional layers as the key difference. Modern CNNs are based on the work of [59, 60, 61, 62, 63, 64] and have been the most extensive researched topic in DL for the last 5 years due to ILSVRC competition for object detection and image classification [65]. Essentially, CNNs learn and use spatial convolutional kernels to transform the input into feature representations/maps. Usually, we alternate these convolutional layers with pooling operations for dimensionality reductions before we use e.g. fully-connected layers for classification and are visually represented in figure 3.3. There are different variants of CNNs but the basic operations are the same.

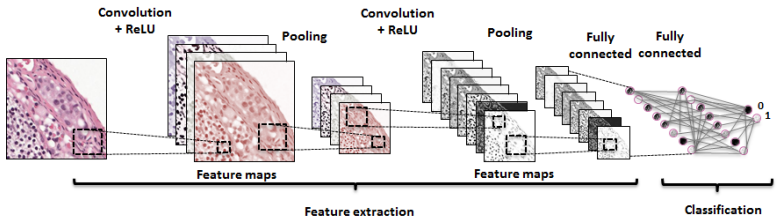


Figure 3.3: Illustration of CNN. We can represent a deep CNN model in many ways but generally, we can divide the network into a feature extraction and classification part depending on the layers (described in section 3.2.1). The network architecture shown here is very similar to LeNet-5 [60], one of the first successful applications of CNN for handwritten digits recognition.

We train CNN models end-to-end in a supervised setting made possible because of the advances of graphical processing units (GPUs) and the significant reduction of model parameters due to weight sharing. This section is structured in such a way that we first describe the layers that we use in our CNN models. Secondly, we elaborate on the network architectures that we experiment with before we cover data augmentation; a key element of training high-dimensional models. Here, we also propose a domain specific augmentation scheme that aims to improve learning of histopathology staining variations.

3.2.1 Layers

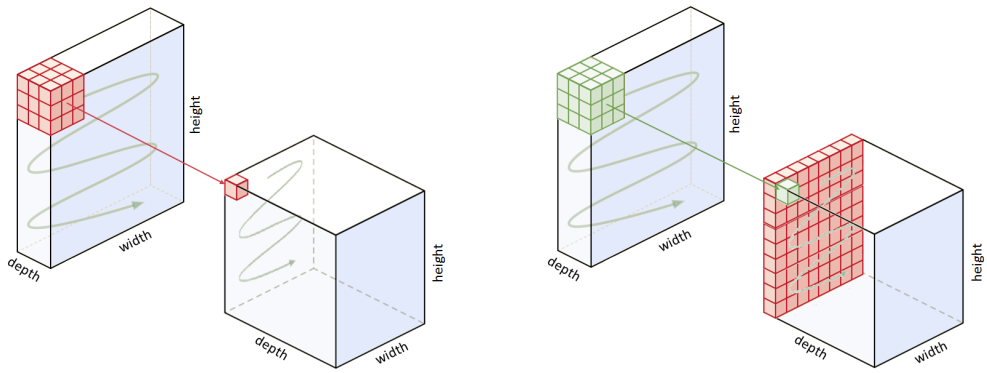
Typically, CNNs are composed of different layer operations which we describe below. We also mention other related operations under each layer type such activation functions under convolutional layers and dropout on fully-connected layers. An important note is that all operations in the CNN must be differentiable to be trained via backpropagation.

3.2.1.1 Convolutional layer

In contrast to fully-connected MLPs, a convolutional layer's parameters are shared in such a way that the input is convolved with a set of K kernels to compute K output feature maps. Each unit in the k-th feature map $\mathbf{Z}_k$ is connected to a receptive field of neighbouring units in the previous layer via the weights of the kernel. All units in the k-th output feature map share the weights of the kernel $\mathbf{W}_k$. We can shortly write the k-th output feature map as:

$$\mathbf{Z}_k = \mathbf{W}_k * \mathbf{X} + \mathbf{b}_k \tag{3.19}$$

where $\mathbf{X}$ is the input volume and $\mathbf{b}$ is the bias term of the layer [41]. The kernel is convolved over the entire depth of the input volume but is constrained in the width and height by the kernel size. The number of kernels K gives the depth of the output volume, see figure 3.4. For each convolutional layer, we select a range of hyperparameters such as stride, padding and kernel size with stride being the step-size of the convolution. For example with stride equal one, the kernel is applied to each input unit and with stride equal two, the kernel is applied to every second input unit. We can use padding e.g. zero-padding to keep the width and height dimensions as convolution will decrease the dimensions depending on the kernel size.



(a) First feature map **(b)** Second feature map

Figure 3.4: Kernel covering 3×3-receptive field. In (a), the first kernel (red) is convolved with the entire input producing the first feature map of the output volume (red). In (b), the second kernel (green) produces the second feature map of the output volume (green) and so on. Here, the depth of input volume is three, representing an image with 3 color channels (RGB). However, the convolution operation is usually across the entire depth of the input, i.e. the kernels of the second convolutional layer span across all feature maps of the previous layer. This is the power of deep CNNs as layers learn local features based on the previous layers building up hierarchical representations of the input. The features of deeper layers are typically complicated representations build on previous simpler ones.

Similar to equation 3.1, the output units of the feature maps (convolutional fea-

tures) are also subject to an element-wise non-linear activation function. Many activation functions exist [66, 55, 67, 68], but we only use Rectified Linear Units (ReLU) [69]:

$$\mathrm{ReLU}(\mathbf{Z}) = \max(0, \mathbf{Z}) \tag{3.20}$$

which is the most common non-linear transform that has been shown to work very well empirically [66, 70]. The non-linearity lets the CNN learn non-linear features and this specific non-linearity makes the network easier to train [71] as it introduce sparsity on activations.

From an image processing perspective, we can see a convolutional layer as a large filter bank. The key difference is that the weights of the kernels are trainable, i.e. adjustable during training as described in section 3.1.2. Therefore, we can introduce L_2-regularization on weights of convolution layers to avoid overfitting and promote better generalization:

$$E_{L_2}(\theta) = E(\theta) + \sum_j ||\theta_j||_2^2 \tag{3.21}$$

where we add $\frac{1}{2}\lambda\theta^2$ to the loss of each weight in the layer. λ controls the degree of the regularization. This is commonly known as weight decay that penalizes larger weights so kernels do not consists of a few large weight and some smaller weights. This encourages a layer to use all of its inputs a more equally rather than using only some of its inputs a lot. Practically, we tune the value of λ by using figure 3.2b. After a convolutional layer, we typically use pooling layers. Sometimes convolution, non-linearity and pooling operations are combined and referred to as a single layer but we describe it below as an independent operation because it is a separate operation in a network.

3.2.1.2 Pooling layer

The pooling layers are instrumental for CNN as they aim to achieve shift-invariance by reducing the resolution (width and height) of the feature maps [71]. A pooling layer aggregates neighbouring units in each feature map via a max or average function typically, i.e. the depth of the input volume is unchanged but the width and height are decreased. For these subsampling layers, we also have to select hyperparameters such as the stride and pooling size (similar to kernel size of convolutional layers). In this thesis, we use max pooling and average pooling [72], see figure 3.5 for the concept of these operations. We also use global average pooling, a operation that computes the mean of each entire feature map to drastically decrease the spatial resolution into a single vector. The resulting vector is then fed into the output layer or one or more fully connected layers.

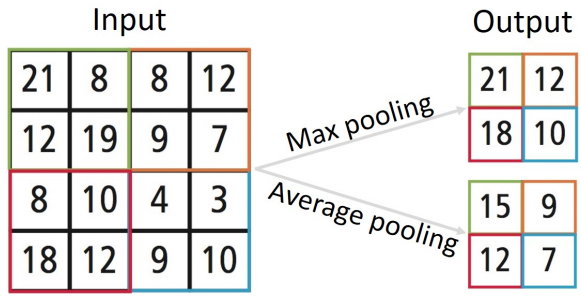


Figure 3.5: Max and average pooling. Here, pooling is performed on a 2-by-2 neighbourhood with stride = 2 resulting in downscaling the resolution (width and height) by factor two. The max pooling operation simply selects the largest value while the average calculates the average of the neighbours. Figure adapted from [73].

3.2.1.3 Fully-connected layer

The fully-connected (FC) layers are traditionally added at the end of the CNNs to perform a high-level reasoning task such as classification in our case. Here, all units of the previous layer are connected to every unit in the fully-connected layer as described in section 3.1.1. With these layers, we introduce a large number of parameters to our model as weights are no longer shared. Hence, fully-connected layers are prone to overfitting which we can combat using Dropout [74]. This method works by dropping units in a layer during training, i.e. setting its incoming and outgoing weights to zero. Which units are dropped for each training iteration is random as we drop each unit in a layer with a fixed probability p independent of the other units of the layer [74]. p is usually between 0.2 and 0.5 but is a hyperparameter for fully-connected layers. Using Dropout, we train slightly different thinned models for each iteration, resulting in simple approximate model averaging at test time [74]. Therefore, we significantly reduce overfitting. Similar to weight decay, we can add Dropout and adjust p while monitoring the learning process.

The basic components of CNNs have now been described, and now we can describe how these are structured together in deep architectures.

3.2.2 Architectures

In this section, we describe the deep CNN architectures used in this thesis and their main differences. Specifically, we first describe the VGG-net [64] which is one of the most commonly used networks. Secondly, we explain the more advanced Inception module [62] and Google's third generation network Inception-V3 [75]. We also introduce the ResNet architecture [63] that is currently the best performing architecture in ILSVRC15 [65]. Due to the scope of this thesis, we use existing architectures that

have been successful on similar tasks [35] because building our own networks usually require longer development time with many iterations and experiments.

3.2.2.1 VGG-net

Inspired by the disruptive AlexNet [61], VGG-net is a very deep CNN developed by the Visual Geometry Group, University of Oxford [64]. We use the most popular version **VGG-19** that consists of 19 trainable layers (16 convolutional, 2 fully-connected and 1 output layer) and 5 max pooling layers. We only change the output layer to a sigmoid operation (equation 3.5), see figure 3.6 for full architecture. All convolutions use ReLU, stride = 1 and zero-padding to keep the output height and width the same as the input. However, all max pooling layers use 2-by-2 neighbourhood with stride = 2 which results in significant dimension reductions. This type of network takes advantage of fixed 3×3 convolutions and a brute-force deep architecture. As a consequence, the network becomes very large in terms of the number of parameters. Since adding more layers will not necessarily increase the performance of CNNs [63], there have been proposed other more complex building blocks such as the Inception module.

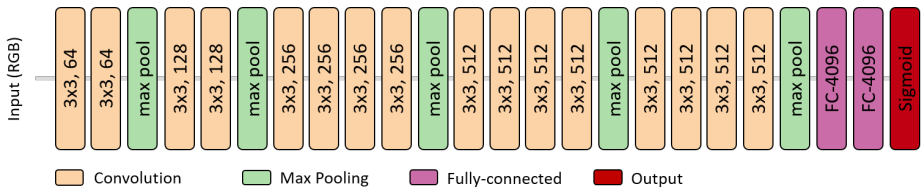
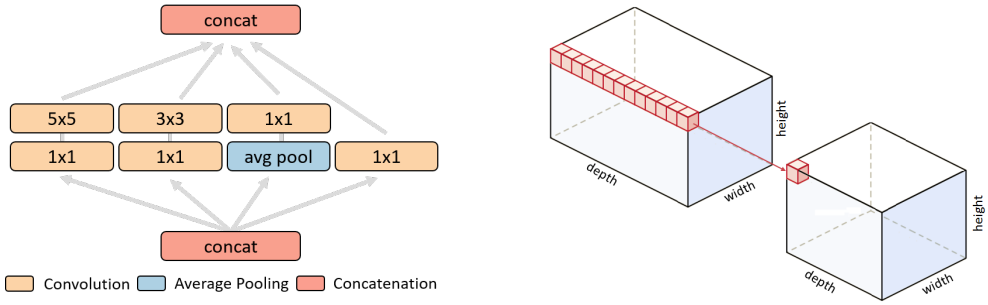


Figure 3.6: VGG-19 architecture. In this network, convolution layers are simply stacked sequentially followed by a max pooling operations producing a 5 convolutional blocks. The kernel size and number of kernel are listed for each layer, for example 3x3, 64 indicates 64 kernels with a size of 3×3 pixels. The two fully-connected layers are added in the end to approximate the classification function. The total number of parameters is 70.368.321 using a 128×128×3 input.

3.2.2.2 Inception-V3

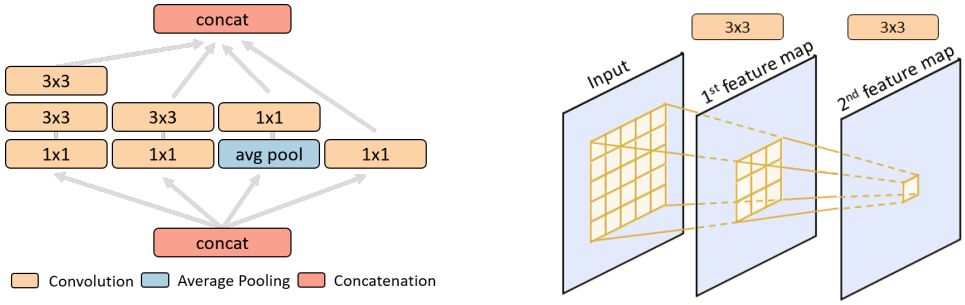
This network is the third generation of GoogLeNet [62] (Inception-V1) from ILSVRC14 that used so-called inception modules inspired by the Network-in-Network structure [76]. The general idea is that the linear sequential structure of convolutional layers can be replaced by a micro network with multiple side-by-side convolutions with different kernel sizes to detect different visual patterns [71]. These representations are then concatenated into the output, meaning simply stacking the feature maps into a larger volume. Another aspect of the inception module is 1×1-convolutions that significantly reduce the number of parameters without increasing the computational cost [62]. See figure 3.7 for the structure of the original inception module.



(a) Original inception module [62] **(b)** 1×1 convolution

Figure 3.7: Key elements of Inception modules. In (a), the original inception module is illustrated. Notice the use of 1×1-convolutions in all branches and how the network fans out into convolutions with different kernel sizes to easier detect features of different sizes. In (b), we see how the 1×1-convolution performs feature reduction as it only changes the depth dimension. The 1×1-parameters are still trainable followed by ReLU, hence the operation works as feature selection on the input feature maps.

There are three main differences between Inception-V1 and Inception-V3; 1) the use of batch normalization layers, 2) the replacement of the 5×5-convolution with two 3×3 sequential convolutions instead, see figure 3.8 and 3) factorization using asymmetric convolutional operations. Batch normalization layers [77] promote gaussian distributions of activations, preventing internal co-variate shifts between layers. This has some nice properties as it results in faster training, acts as a weight regularizer and deals with other initialization problems. By replacing the computational demanding 5×5-convolution, the number of parameters is further decreased without losing significant representational power, see figure 3.8b. The same idea relates to asymmetric convolutions that replace some of the n×n-kernels later in the network with a 1×n-convolution followed by a n×1-convolution [75]. We use the Keras implementation [53] with minor changes e.g. no auxiliary classifier and added extra fully-connected layers with dropout, see figure 3.9 for the full architecture. We keep the specific configuration (number of kernels, padding, stride) similar to the original implementation [75].



(a) Inception module from [75] **(b)** Receptive field

Figure 3.8: Key elements of Inception-V3. In (a), two 3×3-convolutions are stacked instead of 5×5-convolution in figure 3.7a. As shown in (b), a unit in the second feature map still has the same receptive field in the input as a single 5×5-convolution. Each convolution still includes ReLU activation function. These structures are less computational demanding and include more non-linearity than the original inception module.

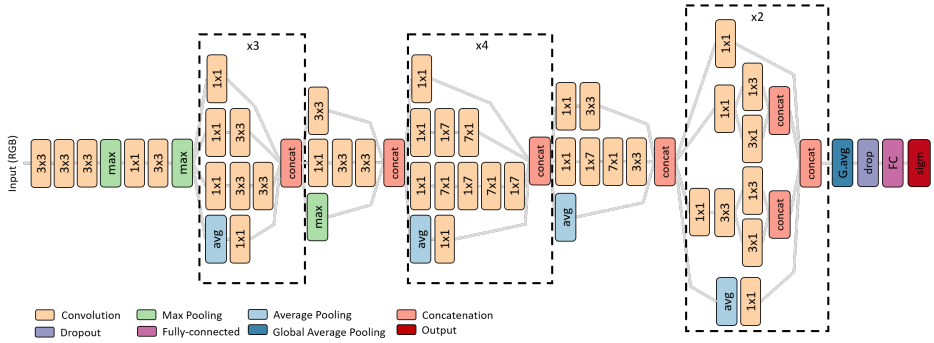


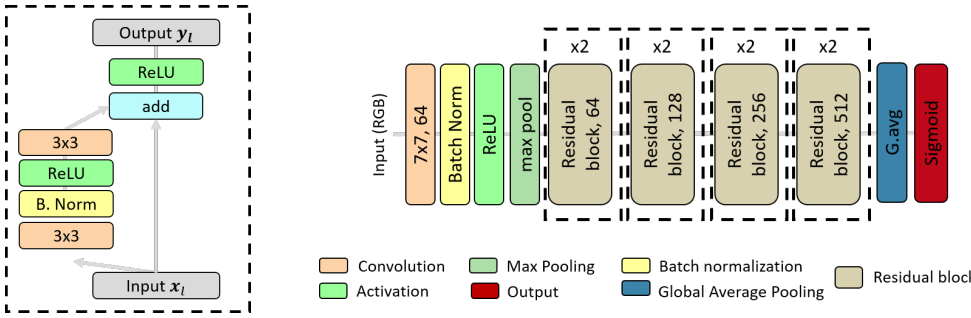
Figure 3.9: Inception-V3 architecture. In this network, the structure is very complex with extensive use of inception modules and asymmetric factorization. The three blocks in dotted boxes are sequentially repeated 3, 4 and 2 times, respectively. We only use one output (sigmoid layer) for the loss function. The total number of parameters is 29.368.321 using the original configuration and a 128×128×3 input which is relatively low compared to a VGG-style network. Figure adapted from [78].

3.2.2.3 Deep residual networks

Deep residual networks (ResNets) [63] refer to an architecture that uses so-called residual blocks:

$$\mathbf{y_l} = \mathbf{x}_l + F(\mathbf{x}_l, \boldsymbol{\theta}_l) \tag{3.22}$$

where in each block l, its parameters only need to learn the residual $F(\mathbf{x}_l, \boldsymbol{\theta}_l)$ w.r.t. to the identity [79]. In a residual block, the input is added to feature maps through skip connections, see figure 3.10a. This structure promotes learning of simple representations in each block which are stacked sequential like the VGG-net. ResNets have been shown to ease training, especially for very deep CNNs and having better performance in image recognition tasks [65]. We use the ResNet-18 architecture, which have 8 residual blocks, each with two convolutional layers, see figure 3.10b. Due to time limits, we did not experiment with deeper versions of ResNet which would have been preferable but left to future work.



(a) Residual block **(b)** ResNet-18

Figure 3.10: Residual networks. In (a), the residual in each block is added to the output feature maps followed by a non-linear transformation. The residual is then learned by the convolutional layers. In (b), the ResNet-18 architecture repeats each residual block sequentially, so the total number of residual blocks is equal to 8. The max pooling layer uses 3-by-3 neighbourhood with stride = 2. The number of kernels of the 3×3-convolutions in the residual blocks is halved for every second block (64, 128, 256 and 512). The total number of parameters is 11.187.841 using a 128×128×3 input. Figure adapted from [63, 79].

3.2.3 Data augmentation

The performance of CNNs depends to a high degree on the amount of labelled training data but also on variability in the training data. Generally, deep CNNs aim to learn the variability of features that are significant for the labelling or classification and discarding the irrelevant features. In computer vision, this means that the label is most likely invariant to certain image transformations [80] and utilizing this knowledge to perform data augmentation has been shown to increase the performance of

deep CNNs [61, 62]. Data augmentation refers to the process of applying some kind of image transformation on the available data without altering the image label, e.g. distortions to pixel intensities or spatial transformations. If performed correctly, the CNN is trained on more data variability which results in more robust models.

Data augmentation schemes have also been used to train CNNs on histopathology images, especially spatial transformation such as rotation, flipping and cropping [35]. These transformations are completely valid as there are no specific orientation for placing tissue sections on glass slides. Furthermore, as mentioned in section 2.2.2.1, one of the largest challenges of applying IA including CNNs to H&E-stained images is the extreme stain variation which in many cases is irrelevant for the classification. Traditionally, CNN-based applications have overcome this by normalizing stain intensities both during training and inference [35, 25, 29, 81]. Recently[2], [26] used color augmentation schemes that have been successfully used for training CNNs on natural images. This involves randomly changing brightness, saturation, hue and contrast of each image according to some fixed parameters. The aim was to augment the color intensities to force the CNN to learn color-invariant features. Independently of [26] but with the same end-goal, we propose another more domain specific data augmentation approach for H&E-stained images.

3.2.3.1 H&E augmentation scheme

Here, we present a domain specific data augmentation scheme that specifically augments H&E-stain intensities by randomly vary fixed H&E-stain vectors. Instead of randomly perturbing image colors as in [26], we deliberately augment the variability that is well-defined by the H- and E-dyes. For digital image processing in histopathology, it is well-recognized that we can define fixed stain vectors for the H and E-colors [82, 83, 10], which is precisely what we take advantage of in our method. The augmentation scheme is described in figure 3.11. We use the color deconvolution method from [84] that use optical density (OD) transform to project the image on to the H- and E-stain vectors, i.e. the RGB image $\mathbf{I}$ is converted as follows,

$$\mathbf{O_D} = -\log\frac{\mathbf{I}}{I_0} \tag{3.23}$$

where I_0 is the intensity when no stain is present [82]. We use H&E specific stain vectors given in [84] such that stain matrix $\mathbf{M}$ is defined as:

$$\mathbf{M} = \begin{bmatrix} 0.650 & 0.072 & 0 \\ 0.704 & 0.990 & 0 \\ 0.286 & 0.105 & 0 \end{bmatrix} \tag{3.24}$$

where H is the first column and E is the second column. These stain vectors represent the stain colors in the OD space. This results in the projection of the image on to

[2] During the work of this thesis

the stain vectors of $\mathbf{M}$:

$$\mathbf{I}_{\text{H\&E}} = \mathbf{M}^{-1}\mathbf{O}_{\mathbf{D}} \tag{3.25}$$

where $\mathbf{I}_{\text{H\&E}}$ holds the intensity of H-stain, E-stain and a residual for each pixel. Here, we use a rather näive approach that perturbs the intensities of $\mathbf{I}_{\text{H\&E}}$ randomly with a range $\pm$[1-10] percentage of the original intensity. This can be done independently or combined on the stain intensities. Finally, we perform the inverse transformation back to RGB-space where we clip values to [0-255] to ensure a valid range for RGB-colors.

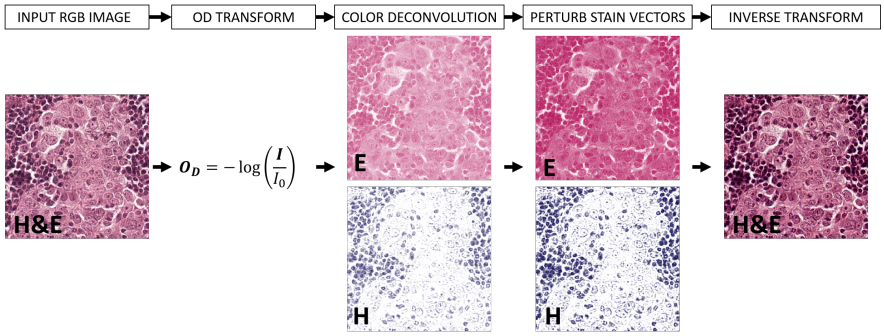
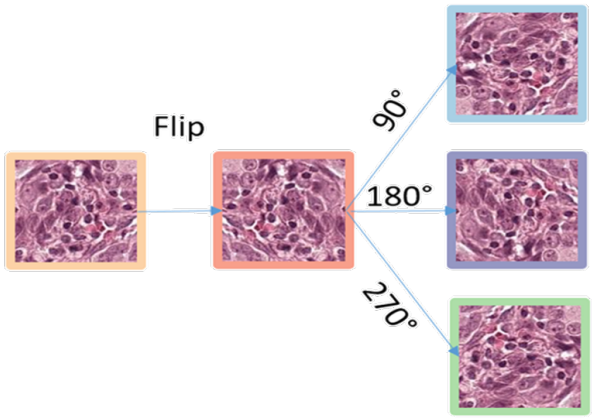
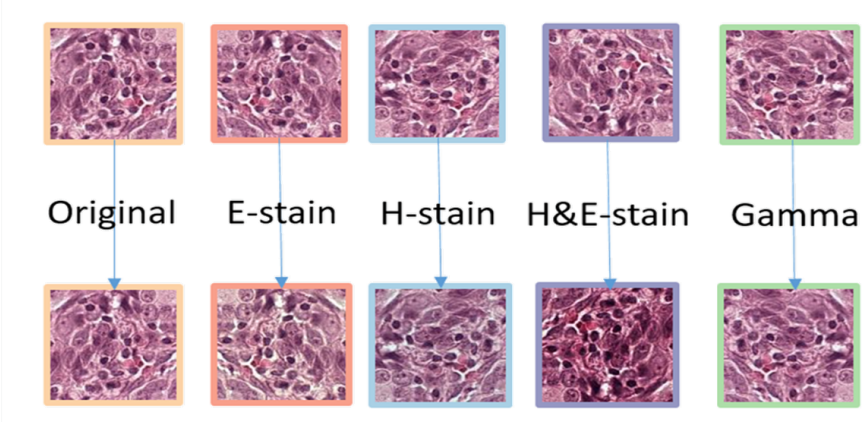


Figure 3.11: Flow chart of H&E augmentation. Here, the steps to augment the H- and/or E-stain are visualized. The H and E image are shown in colors to illustrate the color differences but technically they are single channels. In this specific case both H- and E-intensities are increased to augment an image stained with a different staining protocol than the original.

Practically, we implement the method together with spatial transformations and γ-correction [85] such that the augmented data is both different spatially and color-wise, see figure 3.12. We add the γ-correction as part of the data augmentation framework to augment the different scanner settings when the training data is from multiple pathology laboratories.



(a) Spatial transformations



(b) Color transformations

Figure 3.12: Augmentation scheme. Our total data augmentation consists of 4 spatial transformations (a): vertical flipping and rotating the flipped version 90°, 180° and 270°. For each of the 4 transformations, we vary either the H-stain, E-stain, H&E-stain according to the method in figure 3.11 or apply a gamma-correction with randomly selected γ-value between 0.4 and 1.6. We considered more extensive augmentation by applying the color transformations to a spatial version but due to limited hardware and training time, we chose this approach.

3.3 Summary

Deep CNNs learns hierarchical representations of the input using weight sharing between sequential stacked layers that are powerful for image classification and object recognition. We have implemented three different types of modern architectures, each with their own characteristics. These models can be trained using the backpropagation of the loss function through each modular layer, where the training is a global optimization task to minimize the loss function w.r.t. the parameters. For this, we use mini-batch stochastic gradient descent to iteratively update the parameters of the model until the network has converged. We also proposed a new domain specific approach to augment the variation of known H&E stain intensities. Next, we cover the specific methodology and results of our two algorithms, which we describe separately in chapter 4 and chapter 5.

CHAPTER 4

Training deep CNNs using virtual double staining

In this chapter, we investigate a novel approach to train deep learning algorithms in digital pathology using image registration of differently stained serial tissue sections. The method is applicable to many different staining methods but we present it by automatically labelling H&E-stained images using PCK-stained images. In continuation of this approach, we present a deep learning-based method for automated identification of tumor regions in H&E-stained BCa. We use the H&E tumor regions as basis for VDS Ki67 quantification and compare our method to the pure IHC VDS approach described in section 2.2.4.

4.1 Introduction

Being able to perform H&E-based IA instead of using PCK stained sections has several attractive aspects; First, the H&E staining is 25-30 times cheaper than the PCK staining[1]. Secondly, it is always performed before any IHC staining and thirdly, it is the golden standard that human pathologists use. Due the later, the pathological information is available in H&E sections through morphological context but traditional IA struggles to capture this. This is our basis for using deep learning models which recently have shown great promise for H&E-based IA [35, 26, 24].

The method presented here is an extension of the VDS workflow introduced in section 2.2.4. The general idea is to use specific PCK staining as basis for the H&E labelling. This enables training of the deep learning algorithms to use objective true data instead of subjective evaluations, and the developed analysis will be applied to routine H&E stained samples. An overview is shown in figure 4.1.

[1] A PCK reagent costs approximately 25-30 DKK

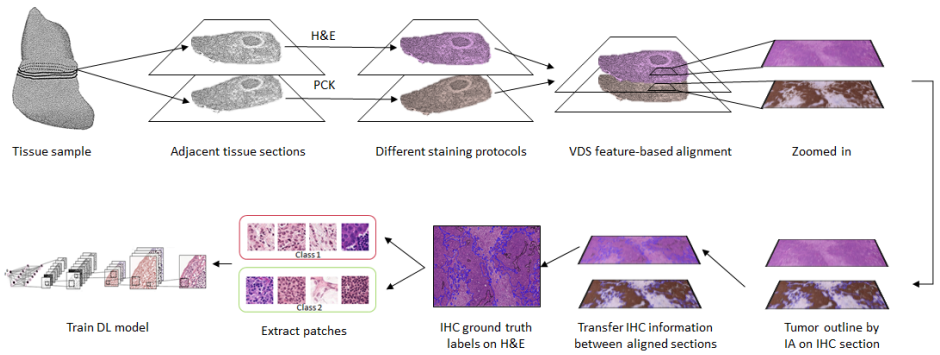


Figure 4.1: Flow chart of training DL models. First, two adjacent section are acquired from the same sample. The first section is stained with the cheap routine H&E staining, while the second section is stained with the more expensive PCK staining that has high contrast between tumor and stoma regions. The WSIs are aligned using image registration so the same tissue structures in both sections are located in the same spatial location. We then use the PCK APP to segment the tumor regions in the PCK section and transfer the results via the alignment. Thereby, we label tumor regions in the H&E stained image which we can use to train deep CNN models.

4.2 Materials and methods

4.2.1 Tissue samples

We reuse images from patients (n=12) diagnosed with breast carcinoma at Herlev Hospital, Denmark. For each patient, 3 tumor blocks were formalin fixed and embedded in paraffin as described in section 2.2.1. From each tumor block, 2×3 serial 3-5 μm sections were taken. Each 3 serial sections make up one collection and stained with H&E, Ki67 and PCK in that order. The data set consists of a total of 72 H&E sections, 72 Ki67 sections and 72 PCK sections (N=216). All sections were scanned using Nanozoomer HT 2.0 (Hamamatsu Phototonics K.K., Hamamatsu City, Japan) at ×40 magnification and saved in their image format; NDPI.

Before any data processing, we randomly split the data set into training, validation and test sets on patient-level to ensure that no WSIs from the same patient are used for both training and testing, see table 4.1. We also visually inspect and remove collections where one or more sections are damaged such as large tissue folds, missing tissue etc. See appendix A for examples on removed collection samples.

Data set	Patients	Collections*	WSIs*
Train	8	39	117
Validation	2	9	27
Test	2	10	30
Total	12	58	174

Table 4.1: Summary of dataset. Each collection consists of three serial sections of H&E, Ki67 and PCK-stained tissue, i.e. for the training set, we have 39 H&E WSIs. *After removing unusable collections.

4.2.2 Manual percentage tumor evaluation

A pathologist with special interest BCa reviewed all 58 H&E sections digitally using VIS without having access to any IHC sections. We provided randomized image data, a spreadsheet for results and a instruction guide to the pathologist. The primary goal of the expert evaluation was to estimate the percentage tumor of each section in a clinical setting.

Our secondary goal was to obtain lesion-level annotations from the pathologist but due to time constraints from the pathologist and this thesis, these are not included in this work.

4.2.3 Image registration of serial sections

As shown in figure 4.2, we perform semi-automatic registration of the sections using Tissuealign^{TM}. We do not perform any manual registration to be as close to a fully-automatic workflow as possible. Unfortunately, the alignment results were not reviewed by a pathologist due to time constraints but as the alignment procedure is used clinically at Danish hospitals, we choose to use the alignment results after reviewed them our self. We obtain three different sets of aligned images:

(a) H&E,PCK-dataset that we use in section 4.2.5.1.

(b) Ki67,H&E-dataset that we use in section 4.2.7.

(c) Ki67,PCK-dataset that we use in section 4.2.4

4.2.4 Ki67 quantification using PCK VDS

In order to establish a benchmark dataset for the Ki67 quantification inside tumor regions, we analyze the Ki67,PCK-dataset with existing image analysis APPs in VIS as described in section 2.2.4. We tune the standard PCK APP and Ki67 APP to the staining intensities, while all other steps are identical to the CE IVD-versions [21, 23].

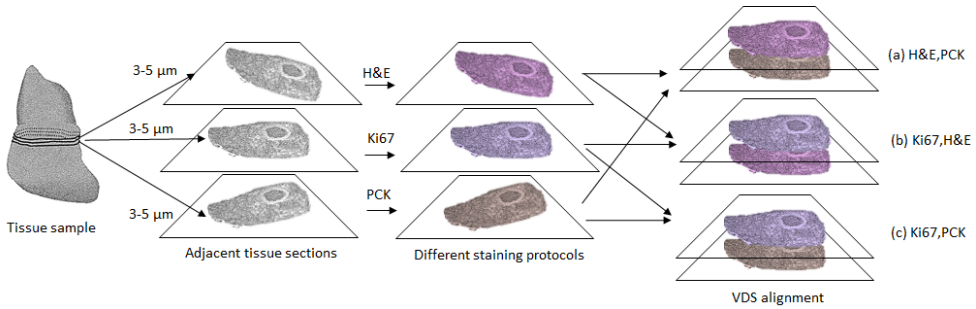


Figure 4.2: One collection with three differently stained sections. Three consecutive sections are cut and stained in the following order; H&E, Ki67 and PCK. We combine the sections using alignment to obtain three subsets; (a) H&E,PCK, (b) Ki67,H&E and (c) Ki67,PCK.

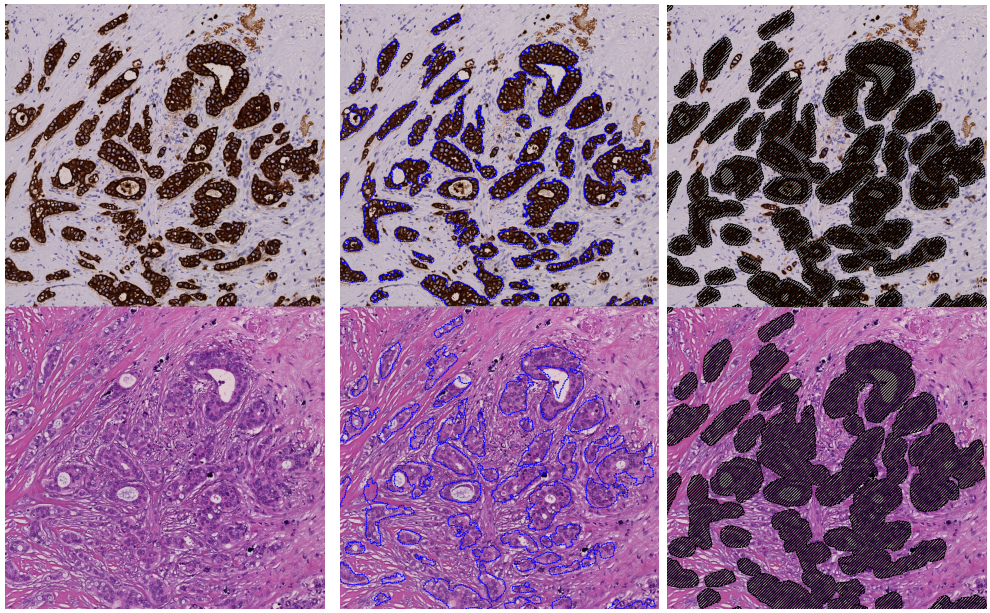
4.2.5 Automated tumor identification using deep CNN

4.2.5.1 Training data

We use the H&E,PCK-dataset from which we want to create a labelled dataset with two classes; tumor and stroma tissue. Tumor regions in the PCK sections are outlined using the tuned PCK APP from section 4.2.4. However, we perform the analysis at 5× magnification and remove PCK-regions smaller than 500 μm^2 to increase the probability of tumor regions being present in both aligned sections. To obtain stroma regions, we dilate the PCK regions to minimize the overlap between tumor and stroma regions, see figure 4.5. From here, we extract small patches (128×128 pixels at 20×) by uniform random sampling (URS) from H&E stained WSIs. We sample 5000 non-overlapping patches from each class in each WSI in the training (n=39) and validation set (n=9) creating a labelled dataset with equal classes, which we refer to $\text{Herlev}_{\text{Normal}}$. Even though the sections originate from the same pathology laboratory, we noticed stain variability in the H&E intensities. Therefore, we also create two other datasets $\text{Herlev}_{\text{Spatial}}$ and $\text{Herlev}_{\text{H\&E}}$. For the later, we use the H&E augmentation scheme from section 3.2.3.1 while for $\text{Herlev}_{\text{Spatial}}$, we only use the spatial transformation of the augmentation scheme. To evaluate the patch-based performance of our models, we perform a full sampling on the test set. I.e. we extract all possible non-overlapping 128×128-patches for both classes to represent the best real test scenario w.r.t. distribution of classes. For the test set $\text{Herlev}_{\text{Test}}$, we do not use any data augmentation.

4.2.5.2 Patch-based classification

For the task of classifying tissue as either tumor or stroma in each WSI, we train a CNN on the extracted image patches using supervised learning. Specifically, we use the CNN models described in section 3.2.2 implemented in Keras [53] with Theano



(a) Aligned **(b)** Tumor regions **(c)** Stromal regions

Figure 4.3: Labelling using IHC image analysis. The top row shows the same PCK image and buttom row show the same H&E image at 10×. In (b), the blue regions are found in the PCK section and considered tumor in the H&E section. In (c), notice that the hatched regions are dilated compared to the blue regions in (b). The non-filled areas are considered stroma regions.

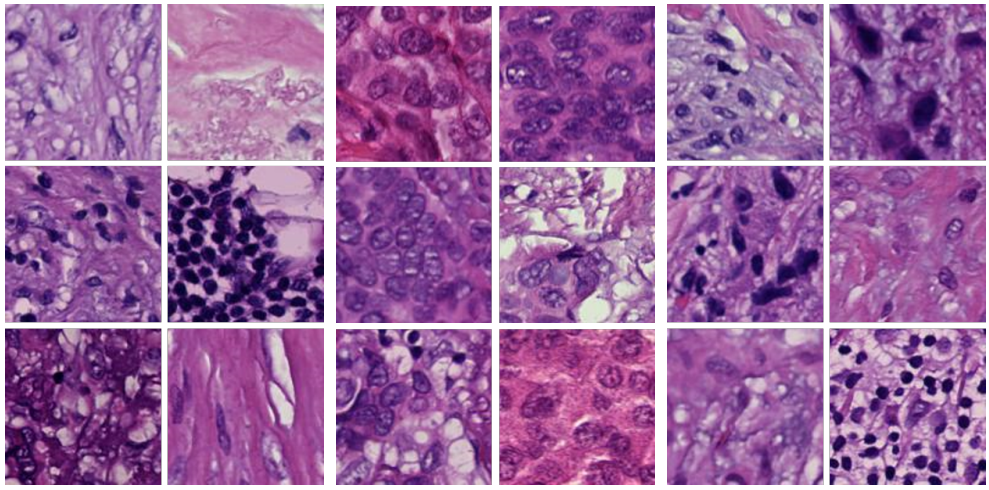
[45] backend. We experiment with SGD, RMSProp and Adam as optimizers where we fine-tune the learning rate and momentum according to section 3.1.3.3. For all experiments, we train our models on a single Nvidia GTX 1080 GPU using a batch size of 32. Early experiments with different pre-processing methods on the raw RGB patches showed that simple linear scaling each channel of the image to [-1,+1] was sufficient. Therefore, all results use this pre-processing method.

4.2.5.3 WSI inference generating tumor heatmaps

To be able to analyze gigapixel WSIs with our trained CNN models, we implemented a framework that use a sliding field-of-view (FOV) analysis, assigning each 128×128 FOV with a probability of being tumor $p \in [0, 1]$. We refer to the output of the analysis as tumor heatmaps as they show the spatial location of tumor probabilities. All analyses are performed at 20× magnification with a stride of 64 pixels, i.e. 50% overlapping patches to obtain a tumor heatmap with sufficient resolution while keeping the processing time within 30 min to 1 hour per slide.

Data set	Tumor	Stroma	Total
$\text{Herlev}_{\text{Normal}}$-train	195K	195K	390K
$\text{Herlev}_{\text{Spatial}}$-train	975K	975K	1.95M
$\text{Herlev}_{\text{H\&E}}$-train	975K	975K	1.95M
$\text{Herlev}_{\text{Normal}}$-validation	45K	45K	90K
$\text{Herlev}_{\text{Spatial}}$-validation	225K	225K	550K
$\text{Herlev}_{\text{H\&E}}$-validation	225K	225K	550K
$\text{Herlev}_{\text{Test}}$	333K	1.07M	1.4M

Table 4.2: Summary of patch-based datasets. These datasets are used to train, validate and train the CNN models. Notice that data augmentation results in 5× the amount of data.

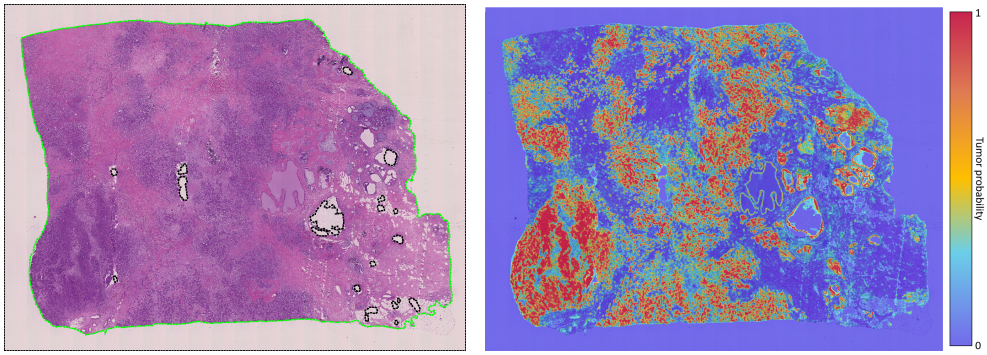


(a) Stroma **(b)** Tumor **(c)** Difficult stroma

Figure 4.4: Example of extracted 128×128 patches. In (a), easy classified stroma patches are shown. In (b), tumor patches differ from stroma as there are larger nuclei with distinct spatial patterns. In (c), more difficult patches are shown as these look similar to tumor patches due to few larger nuclei which probably are infiltrating immune cells.

4.2.6 Automated percentage tumor evaluation

For the task of obtaining percentage tumor evaluation, we first perform WSI inference using the best performing CNN model measured by patch-based accuracy. Secondly, we threshold the tumor heatmap $p > u$ to get the binary regions of tumor (T) and



(a) Gigapixel WSI **(b)** Tumor heatmap overlayed on WSI

Figure 4.5: Tumor heatmap generation. In (a), we use built-in tissue detection in VIS to outline the section (green outline) so the deep CNN only evaluates the relevant patches. Black outlines indicate regions that should be disregarded inside the green outline. In (b), we overlay the tumor heatmap and use color-encoding to represent the spatial distribution of tumor probability.

stroma (S). We then obtain the tumor percentage (T_P) by the area:

$$T_P = \frac{\sum\limits_{\text{pix}\in T} WSI}{\sum\limits_{\text{pix}\in T} WSI + \sum\limits_{\text{pix}\in S} WSI} \tag{4.1}$$

This method requires us to specify a suitable threshold value $0 \leq u < 1$. We solve this by selecting u such that we maximize the Dice Similarity Coefficient (DSC) [86] between the PCK regions and T on the training WSIs. We refer to the optimal threshold value as u_{optimal}. The DSC is based on overlapping regions and calculated as:

$$DSC = \frac{2|A \bigcap B|}{|A| + |B|} \tag{4.2}$$

for two sets of regions A and B.

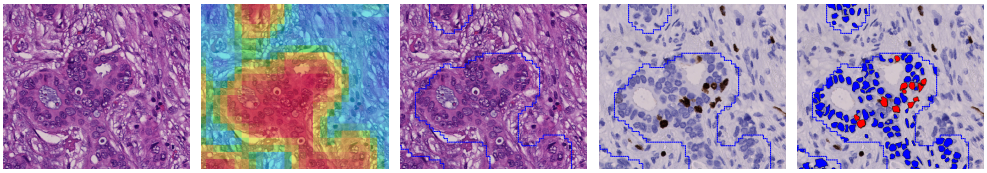
4.2.7 Automated tumor outlining for nuclei quantification

Even though the PCK VDS solves the task of discriminating between tumor and stroma for nuclei quantification, the method is expensive and laborious to perform. Contrary, H&E is a cheap routine staining which is always performed. Therefore, we wish to compare the tumor outlining in H&E sections as the basis for Ki67 nuclei quantification instead of using tumor outlining from PCK. For this tasks, we simply obtain T in the H&E-section on the Ki67,H&E-dataset by thresholding the tumor heatmap $p > u_{\text{optimal}}$. As the sections are aligned, we can transfer T to Ki67-section

and use the tuned Ki67 APP from section 4.2.4 to quantify the PI inside the H&E tumor regions. See figure 4.6 for an example of this process. Furthermore, we use the cut-off at 20% to classify slides into either chemotherapy or no chemotherapy group from table 2.3 on page 12 and compare the agreement using Cohen's kappa value [87]:

$$\kappa = \frac{P_a - P_e}{1 - P_e} \tag{4.3}$$

where P_a is the actual observed agreement (accuracy) and P_e is the estimated agreement.



(a) H&E **(b)** Heatmap **(c)** Tumor region **(d)** On Ki67 **(e)** Ki67 APP

Figure 4.6: Ki67 quantification using H&E VDS. In (b), we threshold the tumor heatmap to obtain the tumor outlines (blue) in (c). Due to the alignment, these regions can be used on the same locations in the Ki67 stained section. In (e), we then use the Ki67 APP to quantify positive (red) and negative (blue) nuclei inside the tumor regions. Notice that the positive and negative nuclei outside the outlines are disregarded in (e) so we only obtain the Ki67 expression for the tumor regions.

4.3 Experiments & Results

In this section, we explain and evaluate our network-related experiments before we describe and discuss the results of our analyses.

4.3.1 Patch-based tumor-stroma classification

4.3.1.1 CNN architectures

We train three models based on the architectures described in section 3.2.2; (ResNet-18, VGG-19 and Inception-V3) for 15 epochs on $\text{Herlev}_{\text{Normal}}$. An epoch is defined as one full iteration over the training set, i.e. the number of parameter updates in one epoch is the total number of patches divided by mini-batch size ($N_m b$=32). After each epoch, we test the model on the validation set. We use SGD_{NAG} for all models but tune the learning rate for each model. The accuracy learning curves are shown in figure 4.7 and we report the patch-based performance on $\text{Herlev}_{\text{Test}}$ in table 4.3. These results indicate that the ResNet-18 is overfitting significantly on the training set, probably due to limited amount of data. The VGG-19 and Inception-V3 converge well on the training set and obtain equal accuracy on the test set. The

VGG-19 network were easier to train regarding the hyperparameter tuning but was much slower to train compared to the Inception network. Due to the large difference in the number of parameters between the two models (VGG-19: 70M vs. Inception-V3: 30M), we choose to perform the rest of the experiments and analyses using the Inception-V3 network architecture. We investigate the relatively low accuracy (∼85%) in the section below.

Model	**Test**	
	Acc.	**Loss**
VGG-19	0.851	0.331
ResNet-18	0.790	0.993
Inception-V3	0.849	0.370

Table 4.3: Patch-based performance results on Herlev$_{\text{Test}}$. Most noticeable here is how ResNet-18 is performing due to overfitting on the training data. The two other network architectures perform similarly on the test set. See section 4.3.1.2 below on why we obtain ∼85% accuracy.

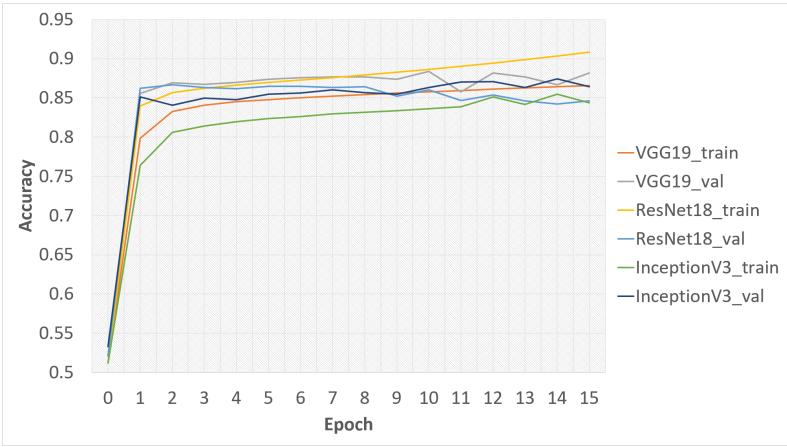
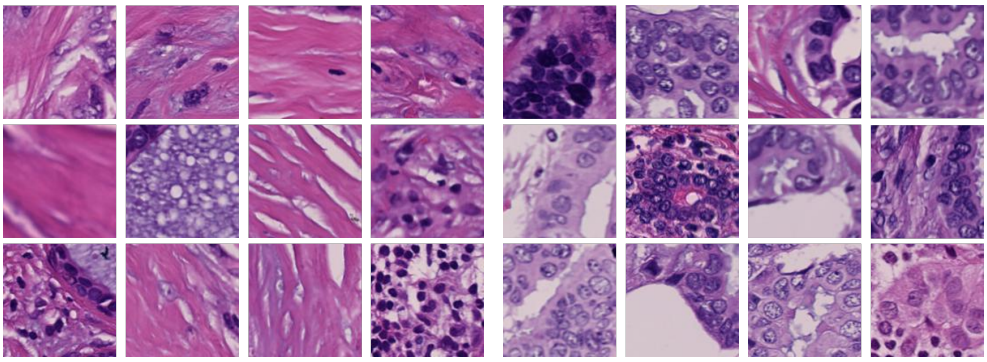


Figure 4.7: Convergence curves for architectures. The VGG-19 net converges a bit faster than Inception-V3 because we could use a higher learning rate while still obtaining sufficient convergence. For the Inception-V3, we use a smaller learning rate to have more steady convergence curve. The ResNet-18 converges much faster (almost within the first epoch) but then starts overfitting (yellow line sudden increase in epoch 9). Here we could have used early stopping by using the parameters at epoch 8 but due to time limits, we leave this to future work.

4.3.1.2 False labelling and generalization

After several unsuccessful attempts to increase the patch-based accuracy by adding an extra fully-connected layer, hyper-parameter tuning and retraining, we investigated the misclassified image patches in the training and validation set, see figure 4.8. We found that approximately 9-12% of the errors were not wrongly classified but falsely labelled due to alignment errors. The alignment errors are probably caused by the physical distance between the H&E and PCK sections, which minimizes the probability of tissue structures being present in both images. This issue is further addressed in section 6.2 in chapter 7. As we are training CNNs models on a large number of image patches, the models are still able to generalize well enough to recognize tumor patches as tumor even though they are labelled as normal. However, this will affect the training of the models and explain why the accuracy is below 90%.



(a) Labelled as tumor, classified as stroma **(b)** Labelled as stroma, classified as tumor

Figure 4.8: False patch labelling. In (a), patches wrongly labelled as tumor are correctly classified as stroma and in (b), patches wrongly labelled as stroma are correctly classified as tumor. This leads to misleading patch-based classification results during testing but also shows how CNNs are able to generalize well over large amounts of data.

4.3.1.3 Optimizers

We perform this experiment to investigate the influence of the different popular optimizers; SGD_{NAG}, RMSProp and Adam. We train three identical Inception-V3 models on the $\text{Herlev}_{\text{Normal}}$-dataset. The learning curves are shown in figure 4.9 with the patch-based accuracy on $\text{Herlev}_{\text{Test}}$ listed in table 4.4. Even though the optimizers converge similarly on the training and validation set, the models trained by Adam and SGD_{NAG} have higher performance on the test set. As the hyperparameters for SGD_{NAG} are easier to tune, we choose this optimization method for the rest of the experiments.

Optimizer	Test	
	Acc.	**Loss**
SGD_{NAG}	**0.849**	**0.370**
RMSProp	0.833	0.389
Adam	0.845	0.371

Table 4.4: Patch-based performance results on $\text{Herlev}_{\text{Test}}$. All models are Inception-V3. The models trained with SGD_{NAG} and Adam generalize better than RMSProp on the test set. This behaviour corresponds well with recent findings in [46].

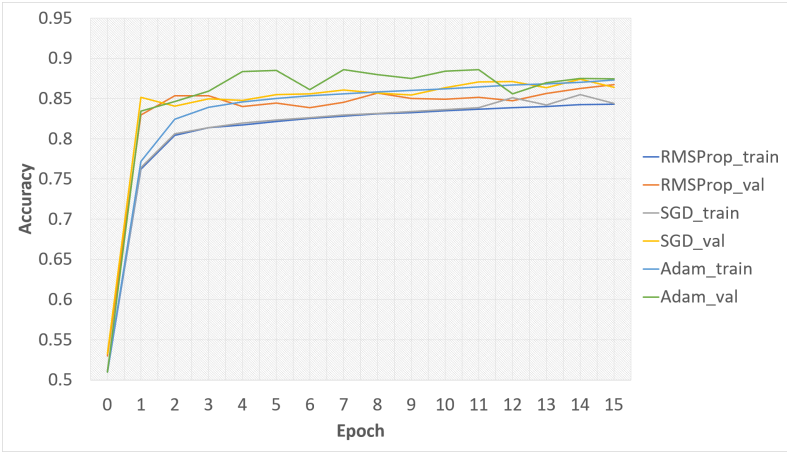


Figure 4.9: Optimizers methods influence on convergence. The Adam optimizer converges faster than RMSProp due to its momentum term. SGD_{NAG} and RMSProp converge similaly on the training set with the validation accuracy ending up to close together for all three methods. These learning curves could indicate that the general learning rate is a bit too high. However, other experiments with lower learning rates showed similar learning curves.

4.3.1.4 Data augmentation

We investigate the influence of data augmentation by training three identical Inception-V3 models on $\text{Herlev}_{\text{Normal}}$, $\text{Herlev}_{\text{Spatial}}$, and $\text{Herlev}_{\text{H\&E}}$ respectively. We report the patch-based accuracy on $\text{Herlev}_{\text{Test}}$ in table 4.5. From these results, we find that using only the spatial transformations do not improve the test accuracy. However, this augmentation method could be useful when the dataset size is smaller. We get the best patch-based results using the H&E-stain augmentation method which indicates that perturbation of colors helps generalization on H&E stained images. As we will see in the next chapter, this improvement is only reinforced when the stain variation increases.

Dataset	Test	
	Acc.	**Loss**
$\text{Herlev}_{\text{Normal}}$	0.849	0.370
$\text{Herlev}_{\text{Spatial}}$	0.844	0.377
$\text{Herlev}_{\text{H\&E}}$	**0.867**	**0.322**

Table 4.5: Patch-based performance results on $\text{Herlev}_{\text{Test}}$. All models are Inception-V3. We see a small increase in accuracy using the H&E-stain augmentation even though the sections originate from the same pathology laboratory. Due to time limits, we did not experiment with random color perturbation from [26] for comparison. This is left to future work.

4.3.1.5 Network size

In this experiment, we investigate if the number of parameters of the Inception-V3 model can be decreased without losing any patch-based accuracy. By number of parameters, we refer to the number of kernels for each convolutional layer and not the total depth of the model (the number of layers). The original model has 32 kernels in the first convolutional layer up to 448 kernels in the deeper convolutional layers. We decrease the number of kernels for each layer with a fixed factor (0.8, 0.6, 0.4, and 0.2) but set the minimum number filter to 16. For example, if the original layer had 80 kernels, a 0.8-factor layer has $0.8 \times 80 = 64$ kernels and so on. For all down-scaled models, we use H&E-stain augmentation. We report the patch-based accuracy on $\text{Herlev}_{\text{Test}}$ in table 4.6. These results indicate that full 30M parameter Inception-V3 possibly has to much capacity, i.e. too many parameters compared to the data. However, we loose $\sim$1-2% accuracy using the smaller models. This is probably due the fact that the full model's number of layers and number of kernels are highly tuned together [75]. Therefore, one should tune the total depth of the model together number of kernels which might give on pair accuracy with the full model's performance. Due to time-limit, we leave this to future work and use the full model trained on $\text{Herlev}_{\text{H\&E}}$ to perform WSI inference.

Dataset	Test		Parameters
	Acc.	**Loss**	
Inception-V3	**0.867**	**0.322**	30.006.721
0.8-Inception-V3	0.860	0.329	17.171.347
0.6-Inception-V3	0.847	0.348	10.319.450
0.4-Inception-V3	0.850	0.341	5.158.523
0.2-Inception-V3	0.849	0.346	1.753.315

Table 4.6: Patch-based performance results on $\text{Herlev}_{\text{Test}}$. We see only a small decrease in patch-based performance when decreasing the depth of the convolutional layers. Even for the 0.2×Inception-V3, we see almost the same performance as for the full network.

4.3.2 Qualitative results of WSI inference

Examples of analyzed H&E images with tumor heatmaps and PCK sections are attached in appendix B.

4.3.3 Quantitative percentage tumor evaluation

In this and the next section, we investigate if a deep CNN model trained using VDS can be used practically on more high-level tasks than patch-based classification. Here, we use the task of evaluating the percentage of tumor in a single section. We compare our method to the manual reading from the human pathologist and the IHC ground truth from the PCK APP. We test if the slope of the fitted line a is significant different from one using a t-test [88] ($H_0 = 1$ @ $\alpha = 0.05$) to indicate if the methods produce the same evaluations. First, we run inference across all WSIs using the trained model use the $u_{\text{optimal}} = 0.7$ as this threshold maximize the median DSC, see figure 4.10.

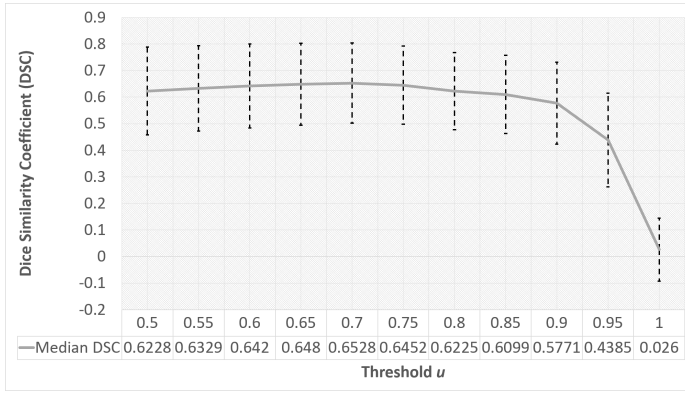
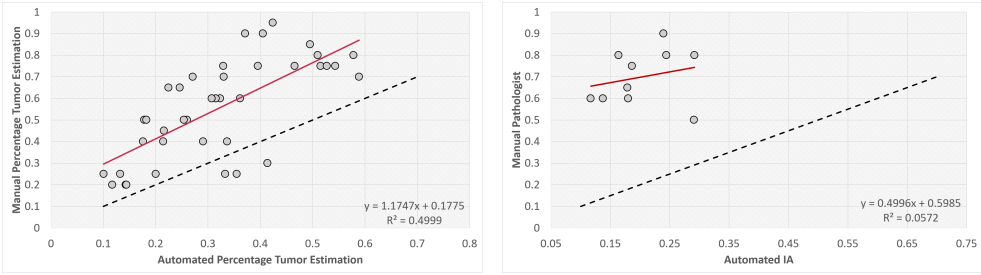


Figure 4.10: Median DSC for different thresholds. We calculate the median DCS for $u = [0.5 - 1.0]$ in steps of 0.05 on the training set. The dashed vertical lines show ± 1 standard deviation (SD). For $u > 0.8$, the DSC drops with quickly. The maximum median DSC=0.6528 is at $u = 0.7$. The DSC is relatively low because the PCK regions are well defined compared to the lower resolution of the tumor heatmap.

The automated percentage tumor evaluation is compared to the manual evaluations in figure 4.11 with unsatisfactory correspondence. By investigating the Bland-Altman plots in figure 4.12, it seems there is a tendency for larger differences between the methods when the average of the methods is large. After consulting with the reviewing pathologist, these results can be explained due to differences in what is considered tumor and stroma regions. A clinical pathologist's estimation is almost macroscopic, i.e. it is a rough estimate that might include stroma tissue inside tumor

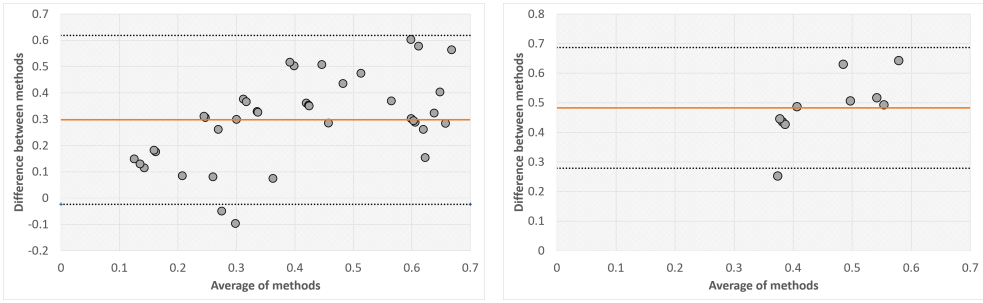
regions. The possible bias observed in the Bland-Altman plots might arise because the pathologist quickly identify larger tumor regions which are then overestimated compared to smaller tumor regions, where the pathologist must use more time. The automated estimation is trained to recognize PCK regions which outlines tumor regions much more precisely than the pathologist. Therefore, our quantitative method is more detail-orientated than the pathologist. Moreover, these results only include manual reading from one single pathologist where a median estimation of a review panel could be closer to the automated estimation.

Therefore, we compare the tumor estimation in H&E to the tumor estimation in the PCK section in figure 4.13. Here, we see an expected higher correspondence between the two approaches. From the Bland-Altman plot in figure 4.14, we do not see the same bias which indicate that this is a better standard to validate our results against. The higher H&E estimations can be explained by the fact that we carry out the analyses at much lower resolution than the one used for the PCK APP. Consequently, some closely identified regions possible merge in the H&E estimation, while these are separable in the PCK analysis. The lower correspondence on the test set might be a result of a small sample (n=10) with a low range of the estimations compared to the training set. Visual inspections of the tumor heatmaps revealed that the tumor regions are located but with a lower probability than the training set. This indicates that the threshold found by the DSC on the training set could be too high. This could potentially be solved by leaving out a subset from the training images for tuning the threshold but this would have require more WSIs. Furthermore, other CNN approaches without the need for a threshold could also might have solve this issue. These approaches are discussed more in details in chapter 7.



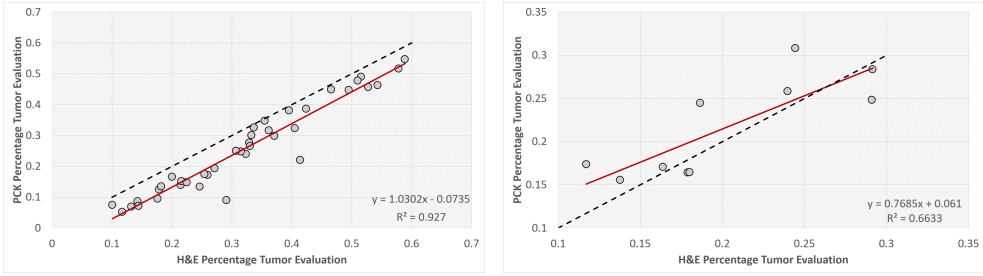
(a) Training set **(b)** Test set

Figure 4.11: Manual vs. automated percentage tumor evaluation. The fitted linear function is shown in red with the perfect correspondence shown with the dashed line. In (a), the correspondence on the training set (n=38) yield an R^2=0.4999 and a=0.7685 significantly different from one ($p < 0.05$) whereas for the test set (n=10), we obtain R^2=0.0572 and a=0.7685 significantly different from one ($p < 0.05$). Generally, we notice that the manual readings are much higher than automated image analysis. Otherwise we do not see sufficient concordance between the manual readings and the automated estimation.



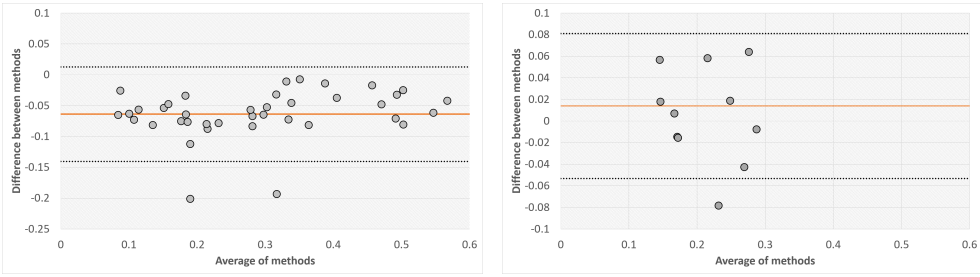
(a) Training set **(b)** Test set

Figure 4.12: Bland-Altman plots for manual estimations. The difference (manual - H&E) is plotted against the average of the two estimations with mean difference (orange line) and 95% confidences levels; $\pm 1.96SD$ (black dotted lines). This plot can be used when none of two methods can be considered ground truth. We see a larger difference sections that include higher percentage tumor. This tendency could indicate that the manual readings might be biased and generally overestimate tumor extent when the tumor percentage is large.



(a) Training set **(b)** Test set

Figure 4.13: H&E vs. PCK automated percentage tumor evaluation. The fitted linear function is shown in red with the perfect correspondence shown with the dashed line. In (a), the correspondence on the training set (n=38) yield an R^2=0.927 and a=1.03 significantly different from one (p = 0.01) whereas for the test set (n=10), we obtain R^2=0.6633 and a=0.7685 significantly different from one (p $<$ 0.05). For the training set, we notice that the H&E percentage tend to be larger than the PCK estimations but there is general satisfactory correspondence between the two methods. For the test set, we have a more random pattern but these results are only for a small test set (n=10). We also note the low range in which the estimations are located.



(a) Training set **(b)** Test set

Figure 4.14: Bland-Altman plots for PCK estimations. The difference (PCK - H&E) is plotted against the average of the two estimations with mean difference (orange line) and 95% confidences levels; $\pm 1.96SD$ (black dotted lines). In contrast to figure 4.12, we do not see the same bias for larger estimations but the overestimation in H&E from figure 4.11 is evident with as small negative mean difference for the training set. However, there are significant outliers in both the training and test set. Using visual inspection, these occur due closely scattered smaller regions with stroma tissue in between them which are segmented well in the PCK section but not using our method (see appendix C).

4.3.4 Ki67 quantification using H&E VDS

We show results from an actual high-level clinical task performed on BCa patients at most Danish hospitals. The objective is to calculate the PI to support the chemotherapy recommendation as described in section 2.2.4. However, we use the H&E tumor outlines from our deep learning-based detection (H&E-Ki67) instead of the PCK section (PCK-Ki67). We compare the resulting PI in figure 4.15. There is good correspondence between two methods with R^2=0.974 and R^2=0.9964 for the training and test set respectively. The slope of the fitted line on the test set is lower than on the training set. As in the last section, we acknowledge that the test sample size (n=10) is generally to small.

Using the Bland-Atman plots in figure 4.16, we notice how the difference between the two methods increases for high proliferating tumors ($\sim$ 2.5-5% points). After visual inspection of these cases, it becomes clear that this is possible due to higher proliferation tumors being less solid, i.e. there are more complex tumor structures with stroma tissue in between tumor regions. Again, our H&E method generates less precise segmentation for theses structure, which then include more stroma close to or in between tumors. Consequently, our method will be more vulnerable to produce a lower PI as negative Ki67 cells are potential included in the quantification results.

When we use the 20%-cutoff for recommending chemotherapy, this small deviation for high PI tumors has almost no influence on the agreement on the clinical outcome as shown in table 4.9. We obtain inter-method agreement of $\kappa_{\text{train}} = 0.8257$ and $\kappa_{\text{test}} = 1.0$ on the training set and test set, respectively. Again these results are based on a relatively small dataset, which needs further validation before such algorithm can be implemented in practice. However, we have shown that it is plausible to perform tumor outlining in H&E stained section using deep CNNs instead of using the PCK stained section.

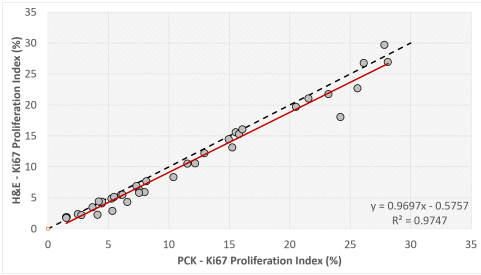
Chemo	**No**	**Yes**
No	30	0
Yes	2	6

Table 4.7: Training set

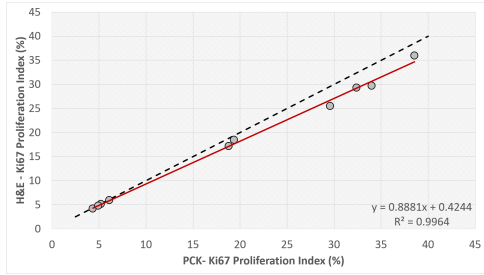
Chemo	**No**	**Yes**
No	6	0
Yes	0	4

Table 4.8: Test set

Table 4.9: Confusion matrices using 20%-cutoff. The left column is PCK-Ki67 and H&E-Ki67 is top row. There is good agreement between the two methods on the clinical outcome with. One of the two errors on the training data is due to a very small difference between PIs around the cutoff (PCK-Ki67 PI = 20.5% and PCK-Ki67 PI = 19.7%). The other error is caused by the intensity dependency of the PCK APP rather than an error in our method as shown in figure C.2 in appendix C on page 84.

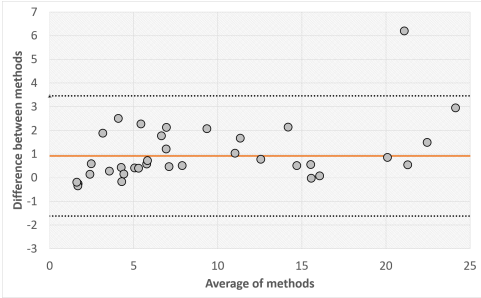


(a) Training set

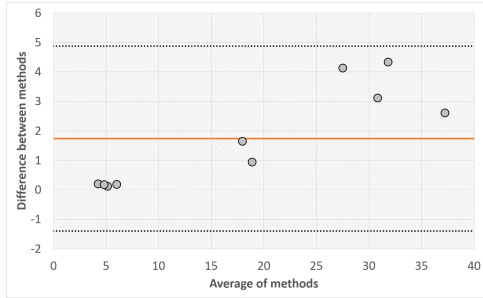


(b) Test set

Figure 4.15: H&E-Ki67 vs. PCK-Ki67 quantification. The fitted linear function is shown in red with the perfect correspondence shown with the dashed line. In (a), the correspondence on the training set (n=38) yield an R^2=0.9747 and a=0.9697 not significantly different from one (p=0.133) whereas for the test set (n=10), we obtain R^2=0.9964 and a=0.8881 significantly different from one ($p < 0.05$). The slope of the fitted function for the test set is lower than for the training set which might be due to a small sample test size.



(a) Training set



(b) Test set

Figure 4.16: Bland-Altman plots for H&E-Ki67 and PCK-Ki67 quantification. The difference (PCK-Ki67 - H&E-Ki67) is plotted against the average of the two quantifications with the mean difference (orange line) and 95% confidences levels; $\pm 1.96SD$ (black dotted lines). For the test set, it seems that when the PI average is higher, the disagreement between the methods is also higher. There is also a general trend that PI H&E-Ki67 is lower ($\sim$ 1-2%). There is an outlier in the training set, where the PI PCK-Ki67 is much higher than our method. We found this to be caused by the intensity dependency of the PCK APP rather than an error in our method as shown in figure C.2 in appendix C on page 84.

4.4 Summary of results

To the best of our knowledge, we are the first to automatically use IHC-information on H&E stained images for training data for deep learning algorithms. We that the found the physical distance between histopathology sections is crucial for this method to be successful. This corresponds well with other VDS-based IA studies [16]. We also showed that it is possible to train a deep CNN to learn discriminative features for tumor-stroma classification in H&E sections, which can be a hard task for handcrafted-features. Moreover, it has been shown that VDS can be used to label H&E sections for training models that then can be implemented and used in combination with existing algorithms and software for digital pathology. We can also conclude that modern CNN architectures perform very similar on detecting tumor and non-tumor but we were unsuccessful to use ResNet-18. Our results indicate that simpler models with fever parameters might be able to solve similar tasks where the general building blocks presented in this thesis can be used as a starting point. We were not able to obtain satisfactory comparison between manual expert percentage tumor estimations and our automated method due to differences in subjective and quantitative definitions. However, we showed that it is plausible to perform tumor outlining in H&E stained sections instead of PCK stained sections when performing Ki67 nuclei quantification which holds great promise for lowering the cost of IHC testing for BCa.

CHAPTER 5

Detecting and classifying lymph node metastases for automatic pN-stage evaluation

In this chapter, we present a deep learning approach to automatically detect and classify BCa metastases. We combine the detection and classification of metastases in multiple WSIs into one outcome; the pN-stage as introduced in section 2.1.1. The method presented here is our submission to IEEE ISBI 2017 Grand Challenge CAMELYON17.

5.1 Introduction

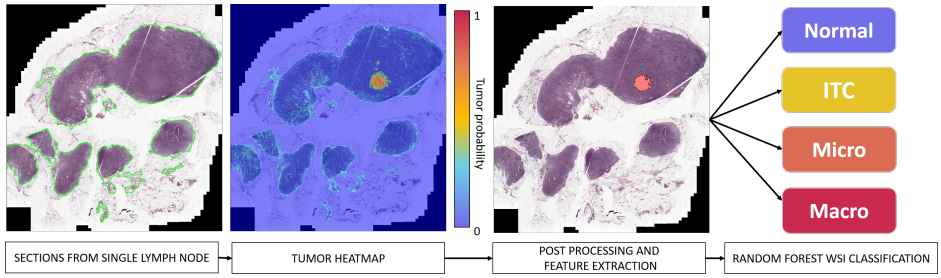
The full algorithm consists on multiple steps using different computer vision and ML methods. Generally, we use a patch-based deep CNN approach to automatically recognize metastases extending [35] by utilizing a deeper CNN network in combination with our proposed stain-specific augmentation method. We also build slide-classifier to categorize WSIs based on the types of metastases found by the deep CNN model. For this task, we use more traditional machine learning on high-level handcrafted-features for each WSI. For the CAMELYON17 competition, we used a simple rule-based system to combine the classification of multiple lymph node slides into the pN-stage. Each step is described in details below and an overview of our approach is shown in figure 5.1.

5.2 Materials and methods

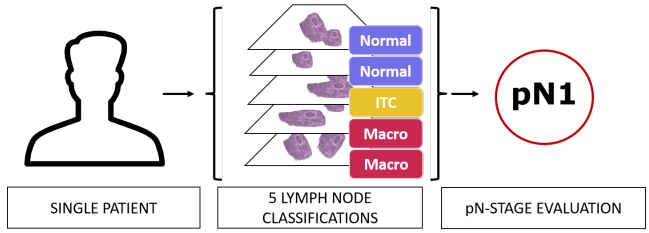
5.2.1 Tissue samples

We use image data of H&E stained lymph nodes sections available through participation in the CAMELYON17 challenge [4]. The total number of images (N=1400) are divided into two main subsets; Camelyon16 and Camelyon17.

The Camelyon16 dataset (Cam16) consists of 400 WSIs from two different pathol-



(a) Metastases detection and WSI classification



(b) pN-stage evaluation example

Figure 5.1: Overview of approach. In (a), we first perform tissue detection to limit the analysis to the relevant image regions. We then use a trained deep CNN to generate tumor heatmaps which we post-process to extract high-level features from each WSI. These features are then used to classify each WSI into one of four classes (Normal, ITC, Micro and Macro). In (b), we use these results to perform patient-level classification to obtain pN-stage evaluation. We use the simplified staging system (table 5.5), which in this case results in pN1-stage.

ogy centres in the Netherlands. This dataset was also used in the CAMELYON16 Challenge [34] where the primary task was to identify metastasis to determine if there was tumor in a WSI or not. This dataset only contains sections with no lesions (normal), micro or macro metastases, i.e. no ITCs. We use this dataset for training and validation.

The Camelyon17 dataset (Cam17) is patient-based dataset with 200 patients allocated equally to train/validation (n=100) and test (n=100). The images originate from five different pathology centres in the Netherlands (two of them are the same from Cam16). For each patient, there are five WSIs, each with tissue sections from a single lymph node. The ground truth for the test set is not publicly available because it is used for ranking qualifying submissions for the challenge.

Before any data processing, we use the original split of Cam16 [34] with an additional validation and test split on the original test set, see table 5.1. For Cam17, we use a patient-level training and validation split to ensure that no WSIs from the

same patient are used for both training and validation. Hence, we have two subsets consisting of 300 and 200 WSIs, see table 5.2.

Data set	Normal	Tumor	Total
Cam16-Train	160	110	270
Cam16-Test-Val	40	25	65
Cam16-Test-Test	40	25	65
Total	240	160	400

Table 5.1: Details on WSI-level of Cam16 dataset.

Data set	Normal	ITC	Micro	Macro	Total
Cam17-Train	164	31	44	61	300
Cam17-Validation	149	4	20	27	200
Total	313	35	64	88	500

Table 5.2: Details on WSI-level of Cam17 training dataset. We can only show the distribution of classes for the official training set. This information is still not public for the test set as the challenge continued after the ISBI workshop.

5.2.2 Manual lesion-level annotation

In contrast to chapter 4, we have manual lesion-level annotations of regions with metastases for a number of WSIs. These annotations are available for all images in the Cam16 dataset while for the Cam17 dataset, these were only available for 10 WSIs per pathology center (total of 50 WSIs). The micro and macro metastases are annotated exhaustively while ITCs are not annotated exhaustively but all ground truth annotations were carefully prepared under supervision of expert pathologists [4].

5.2.3 Manual slide-level classification for pN-stage evaluation

For the patient-level Cam17 dataset, the ground truth slide-label is available for all 500 training and validation WSIs, see table 5.2. For this set, we also have the pN-stage label for all 100 patients. Again, all ground truth labels were carefully prepared under supervision of expert pathologists [4].

5.2.4 Tissue detection

Due to high stain variations from five different centers, the VIS built-in tissue detection was inadequate. Therefore, we create a low-resolution tissue detection APP in VIS. The APP consists of two parts. First, we remove the dark regions arising

during digitization due to the scanners' automatic region-of-interest (ROI) detection, see left-most image in figure 5.1a. This is done by thresholding pixel intensities $I > 0.1$ in the Intensity-Hue-Saturation (IHS) color space [89]. Secondly, we outline the tissue by thresholding pixel-intensities in the negated H&E-Eosin color band at $I_{eosin} > -220$ followed by several morphological post-processing steps; removing very small objects, a close operation and removing objects with mean red intensity below 50 in the median smoothed RGB-color space.

5.2.5 Training set

We use the WSIs from Cam16 together with the manual lesion-level annotations containing two classes; tumor and normal tissue. We sample non-overlapping image patches from both tumor and normal tissue while disregarding normal patches from WSIs reported to have incomplete annotations [26]. We use an image patch size of 128×128 pixels at 20× magnification to keep the input dimensions low but keeping the same receptive field in the tissue as [35] (256×256 pixels at 40×).

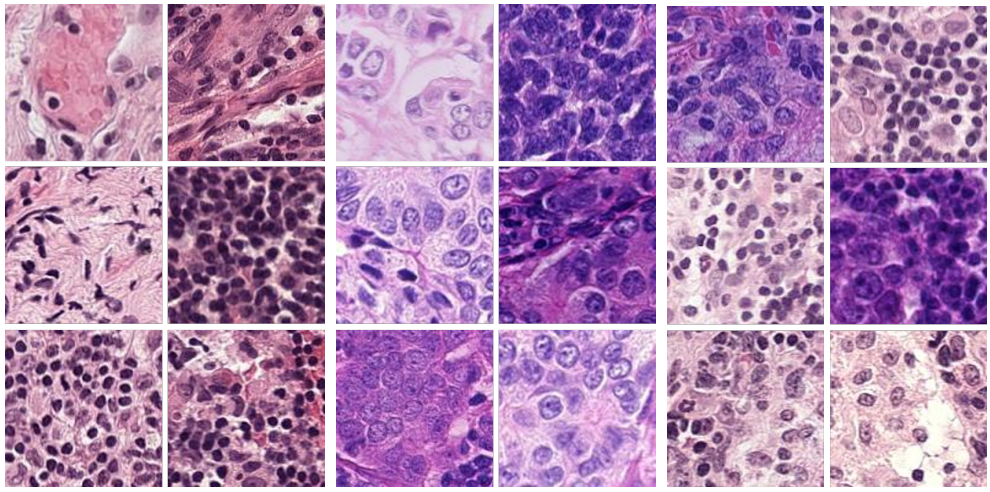
As in the previous chapter, we sample multiple datasets for experimentation and use hard mining similar to [35]. Hard mining refers to retraining a model based on hard examples that were misclassfied in the first sampling. First, we sample patches by URS from the H&E stained WSIs. We sample 2000 non-overlapping tumor patches from each tumor WSI and 1000 non-overlapping normal patches from each normal WSI in Cam16-Train and Cam16-Validation set. Thereby, we create a labelled dataset with equal classes, which we refer to $\text{Cam16}_{\text{Normal}}$. We then use $\text{Cam16}_{\text{Normal}}$ to train a model from scratch (see section 5.2.6 below). We perform hard mining by sampling all normal patches in WSI from Cam16 that were wrongly classified as tumor (n=350K). Based on this, we sample new tumor patches (n=350K) and normal patches (n=350K), creating a more difficult non-balanced data set $\text{Cam16}_{\text{Hard}}$. On this dataset, we perform the H&E data augmentation scheme described previously, creating $\text{Cam16}_{\text{Hard, H\&E}}$. For testing the patch-based performance, we perform full sampling on the Cam16-Test-Test WSIs which we refer to as $\text{Cam16}_{\text{Test}}$.

5.2.6 Patch-based classification

We use the Inception V3-network [75] from section 3.2.2.2. However, we add a 2-layer fully connected network with dropout ($p = 0.2$) before the sigmoid output layer. We implement the model in Keras [53] with Theano [90] backend and trained all models with SGD with NAG (initial learning rate of 0.1, momentum = 0.9) using a batch size of 32 on a single Nvidia GTX 1080 GPU. Early experiments using a fixed learning rate showed that we needed another approach as the learning stopped prematurely. Therefore, we use a learning rate schedule, where the learning rate is dropped by 50% after each 125K gradient update. Thereby, the optimizer performs large gradient

Data set	Tumor	Non-tumor	Total
$\text{Cam16}_{\text{Normal}}$-train	174K	215K	389K
$\text{Cam16}_{\text{Hard}}$-train	350K	700K	1.05M
$\text{Cam16}_{\text{Hard, H\&E}}$-train	1.75M	3.5M	5.25M
$\text{Cam16}_{\text{Normal}}$-val	44K	54K	98K
$\text{Cam16}_{\text{Hard}}$-val	72K	178K	250K
$\text{Cam16}_{\text{Hard, H\&E}}$-val	360K	890K	1.25M
$\text{Cam16}_{\text{Test}}$	591K	7.32M	7.89M

Table 5.3: Summary of patch-based datasets. These datasets are used to train, validate and test the CNN models. Notice that data augmentation results in 5× the amount of data.



(a) Non-tumor **(b)** Tumor **(c)** Difficult non-tumor

Figure 5.2: Example of extracted patches. In (a), easy classified non-tumor patches are shown. These consist mostly of stroma and/or lymphocytes (small dark round nuclei). In (b), tumor patches clearly differs from (a) as the nuclei are larger nuclei with noticeable nucleolus (small dark dot inside nuclei). In (c), more difficult non-patches are shown. These are typically single or multiple immune cells such as macrophages etc. Many of these are very difficult to discriminate from (b) which is why hand-crafted features fall short on this classification task.

steps in the beginning of convergence and smaller steps later for the fine-tuning of the parameters.

5.2.7 WSI inference generating tumor heatmaps

We use the sliding FOV framework, assigning each 128×128 FOV with a probability of being tumor $p \in [0, 1]$ generating tumor heatmaps. Even though a small stride is preferable to increase sensitivity towards small tumor regions, we perform all analyses at 20× magnification with a stride of 128 pixels, i.e. non-overlapping patches. This was crucial in order to finish analyzing the amount of WSIs before the deadline of the challenge (April 6th 2017).

5.2.8 Post-processing and feature extraction

For the task of classifying each WSIs as either Normal, ITC, Micro or Macro, we experimented with different approaches. First, we investigated a CNN classification but early experiments showed that the amount of WSIs were not sufficient for this approach. Therefore, we use post-processing and feature extraction using VIS.

We overlay each tumor heatmap on the corresponding WSI and then threshold the heatmap at $p > 0.5$ to obtain objects which we denote metastatic candidates (MCs). We are not using the Dice Similarity Coefficient (DSC) to tune the threshold because some tumor regions were well annotated while others were very sparsely annotated. This resulted in a misleading DCS when comparing overlapping regions. We also disregard white background pixels (intensity larger than 200 in the IHS-color space).

From the MCs, we select the 5 largest objects based on area using the following method. First, we find the largest object L_1 and include any object less than 0.3 mm away from L_1 as part of L_1. We then do the same for second largest object L_2 in the objects left in MCs and continue until we have the 5 largest objects in the WSI $L = \{L_1, L_2, L_3, L_4, L_5\}$.

For each object in L, we compute different probability and morphological features together with a few global features for all MCs, see table 5.4 on page 61. We also compute the mean filter response of a polynomial blob filter inside each L. The filter size is tuned to be similar to the size of lymphocytes, so this feature represents the number of lymphocytes inside a given object. In total, we compute 42 features for each WSI.

5.2.9 Slide classification

We analyze all WSIs in Cam17-train and Cam17-val with WSI inference followed by post-processing and feature extraction. We use features from these dataset to train and validate a random forest (RF)-classifier to discriminate between WSIs $\in$ {Normal, ITC, Micro, Macro}.

Generally, RF is an ensemble model of many decision trees created using bagging

Feature	Type	Computed for
Area	Morphological	L
Major axis length	Morphological	L
Perimeter	Morphological	L
Max probability	Probability	L
Mean probability	Probability	L
Std probability	Probability	L
Min probability	Probability	L
Mean poly blob	Filter response	L
Max probability	Probability	MCs
Median probability	Probability	MCs

Table 5.4: Summary of WSI features. Computed for L means that we computed these features for each of the 5 largest objects in L. The last two features computed for MCs are global features calculated across all objects in MCs.

[42]. That is, a random subset of the training data is selected to train each tree. In each tree, the splits are also based on randomly selected features. Therefore, this results in dissimilar trees who's predictions are averaged using majority-voting. We choose to use a RF-classifier as it is easy and fast to train and can handle feature importance, i.e. we can inspect which features contribute most to its classification performance. We use the Python implementation in `scikit-learn` [91] to tune the hyper-parameters of the RF-classifier using 3-fold cross-validation on the Cam17-train set. The final hyper-parameters are; Number of trees = 100, max features for best split = $\sqrt{n_{features}}$, minimum number of samples for split = 5, minimum number of samples required to be at a leaf node = 3 and criterion to measure quality of split = entropy.

5.2.9.1 Feature importance

Investigation of the feature importance revealed that not all features contribute to the classification of WSIs. Not surprisingly, the max probability and area of the largest metastases L_1 were the most important features, while features such as std. probability and min probability for the smallest metastases L_5 were not contributing significantly. However, almost all features contributed to some extent to the classification. We did not experiment with feature selection to decrease the number of features, this we leave to future work.

5.2.10 Patient classification

We use the simple rule-based scheme shown in table 5.5 to determine the pN-stage of a patient. This definition scheme is the official pN-staging system of the CAMELYON17

Challenge that simplifies the real clinical scheme from [92]. See figure 5.1b for an pN1-stage example.

Stage	Criteria
pN0	No micro-metastases or macro-metastases or ITCs found.
pN0(i+)	Only ITCs found.
pN1mi	Micro-metastases found, but no macro-metastases found.
pN1	Metastases found in 1–3 lymph nodes (WSIs), of which at least one is a macro-metastasis.
pN2	Metastases found in 4–5 lymph nodes (WSIs), of which at least one is a macro-metastasis.

Table 5.5: pN-stage evaluation. The pathologic node assessment for breast cancer patients. This is a simplified scheme which is used in this thesis.

5.2.11 Negative slide screening

Negative slide screening refers to the task of removing all WSIs which do not contain any metastases while not missing a single positive. Thereby, pathologists only have to review positive WSIs which which holds great promise to reduce their workload. As an additional experiment, we train a RF-classifier separately for this task. In this case, we only have two classes for the slide classification; negative and positive. We train the RF-classifier on the Cam16-Train set using only features from the largest object L_1 and the global probability features from table 5.4.

5.2.12 Evaluation metrics

For the evaluation of the pN-stage classification, we use a five class quadratic weighted Cohen's kappa value [4]:

$$\kappa = 1 - \frac{\sum_{i=1}^{k} \sum_{j=1}^{k} w_{ij} o_{ij}}{\sum_{i=1}^{k} \sum_{j=1}^{k} w_{ij} e_{ij}} \tag{5.1}$$

where k is the pN-stages and w_{ij}, o_{ij}, e_{ij} are elements in the weight, observed, and expected matrices. This metric is widely used for pathology studies with ordered outcome when a small disagreement is less important than large differences between the image analysis and pathologist's ground truth, see table 5.6 for specific interpretation.

Kappa value	Strength of agreement
< 0.20	Poor
0.21 - 0.40	Fair
0.41 - 0.60	Moderate
0.61 - 0.80	Good
0.81 - 1.00	Very good

Table 5.6: Interpretation of kappa value [93].

For evaluating how well our method performs against the leader-board of last year's CAMELYON16 Challenge on classifying WSIs as either positive or negative slides, we use the Area under the curve (AUC) for the Receiver Operating Characteristics (ROC) on WSI-level [34]. This measure represents the trade-off between the sensitivity and the specificity of the binary classification at different thresholds.

5.3 Results

In this section, we describe and discuss the results for our submission to the CAMELYON17 Challenge and compare the performance of our method to the state-of-the-art results.

5.3.1 Patch-based classification

We perform two model iterations to obtain the final trained model. First, we train a model from scratch using $\text{Cam16}_{\text{Normal}}$. Then we retrain new models using $\text{Cam16}_{\text{Hard}}$ and $\text{Cam16}_{\text{Hard, H\&E}}$. From the results in table 5.7, we obtain the highest patch-based classification accuracy=0.966 using both hard negative mining and H&E-augmentation. Consequently, we use the model trained using $\text{Cam16}_{\text{Hard, H\&E}}$ for the WSI inference. Visually, we can investigate the kernels of the first convolutional layer before and after adding H&E-augmentation in appendix D on page 85 to see the effects of the augmentation on the learned weights.

Dataset	**Train**		**Validation**		**Test**	
	Acc.	**Loss**	**Acc.**	**Loss**	**Acc.**	**Loss**
$\text{Cam16}_{\text{Normal}}$	0.943	0.150	0.921	0.217	0.915	0.255
$\text{Cam16}_{\text{Hard}}$	0.950	0.166	0.918	0.258	0.941	0.176
$\text{Cam16}_{\text{Hard, H\&E}}$	0.937	0.190	0.922	0.234	**0.966**	0.121

Table 5.7: Patch-based performance results. We can only directly compare the models' accuracy using the $\text{Cam16}_{\text{Test}}$-set as the validation and training sets are different. We see that hard negative mining definitely improve the performance on the $\text{Cam16}_{\text{Test}}$-set but are probably overfitting on the staining intensities of the training data. Finally, we see that the H&E-augmentation forces the CNN to generalize better as this model has the highest patch-based accuracy on the $\text{Cam16}_{\text{Test}}$ (shown in bold).

5.3.2 Qualitative results of WSI inference

We have attached representative example images with the corresponding tumor heatmap for each of the metastases types (ITC, micro and macro) and false positive regions in appendix E.

5.3.3 Automated pN-stage evaluation

For the automated pN-stage evaluation, our method obtains a kappa score $\kappa = 0.8172$ showing very good agreement with the pathologist ground truth. This score is currently ranked 5th in the CAMELYON17 Challenge. See table 5.8 for the Top 10 performing teams.

We also report the slide-level performance on the Cam17 dataset in table 5.9 using the accuracy score.

Rank	Affiliation	Kappa-score
1	Harvard Medical School - Mass. General Hospital - Center for Clinical Data Science	0.8981
2	Electrical Engineering Department, Eindhoven University of Technology	0.8759
3	Indica Labs	0.8638
4	The Univercity of Tokyo, Tokyo Medical and Dental University	0.8637
5	**Our method**	**0.8172**
6	Imsight Medical Technology, the Chinese University of Hong Kong and Xiamen University	0.7858
7	ContextVision	0.7721
8	Proscia Inc., Carnegie Mellon University and Moffitt Cancer Center	0.7664
9	Middle East Technical University	0.7632
10	Karlsruhe Institute of Technology	0.7315

Table 5.8: Top 10 teams of CAMELYON17. Results submitted until the challenge deadline with our submission shown in bold. We refer to [4] for the full leader-board.

Our method performs well on micro and macro metastases, but struggles to discriminate between normal WSIs and WSIs with ITC. This affects the pN-stage evaluation for especially the pN0(i+) patients which in turn affects the kappa score negatively. See appendix F for the confusion matrices of the classification results on both slide-level and patient-level for the training and validation set. Visual inspection indicates that the CNN model actually generalizes well enough to detect most ITCs but the slide-based features are not adequate for WSI classification purposes due to too many false positive regions in normal WSIs. The drop in performance from training to validation and test set is probably due to the fact that the RF-classifier is overfitting to the training data as it tries to overfit distinguish ITCs from Normal, so the decision rules does not generalize well enough for the test set.

Even though our patch-based results are satisfactory, we believe that the deep CNN model can be improved with more and better class distributed training data. If the tumor vs. non-tumor patch classification is improved then the slide classification could be simplified to simply measuring the presence and the size of metastases, hence we remove the need for handcrafted features and ML-classifiers. These considerations are described as future work in section 6.8.

Dataset	**Accuracy (Slide-level)**	**Kappa-score**
Cam17-Train	0.9400	0.8800
Cam17-Validation	0.7500	0.8100
Cam17-Test	N/A	0.8172

Table 5.9: Slide-level and patient-level performance for the CAMELYON17 dataset. Our slide level accuracy drops for the validation set, which can indicate that the high-level features are not well suited for this task. Again, this relates to the discrimination between Normal WSIs and ITC WSIs. We are not able to show the slide-level performance on Cam17-Test as the challenge is still active and the ground truth is not released.

5.3.4 Negative slide screening

As an additional experiment, we use the Cam16-test set for evaluation of our separate RF-classier for negative slide screening as we then can somewhat compare our results to last year's leader-board [34] and a recently published article from Google Brain [26]. Our RF-classifier obtains an AUC score of 0.9745 (figure 5.3) which is higher than the highest score before the deadline, see table 5.10. This AUC score also surpasses the AUC of the pathologist (0.9660) in the CAMELYON16 study [34, 26]. Our method yields around 60% specificity at 100% sensitivity, which means that we can remove around 60% of the negative WSIs. The errors made by the algorithm involved very small metastases regions clustered in groups which together form a micro metastases. These metastases were regions of high probability but our threshold approach recognizes them as individual small regions. These were too similar to small false positive regions in the negative WSIs equivalent to ITCs in the CAMELYON17 Challenge, i.e. these WSIs were classified as negative. One way to overcome this could be to include regions in close proximity as described in section 5.2.8 and continue iteratively until no more surrounding regions are grouped together. This would not have the same effect on negative slides as the small false positive regions are more dispersed. However, we leave it to future experiments to improve the performance on these cases.

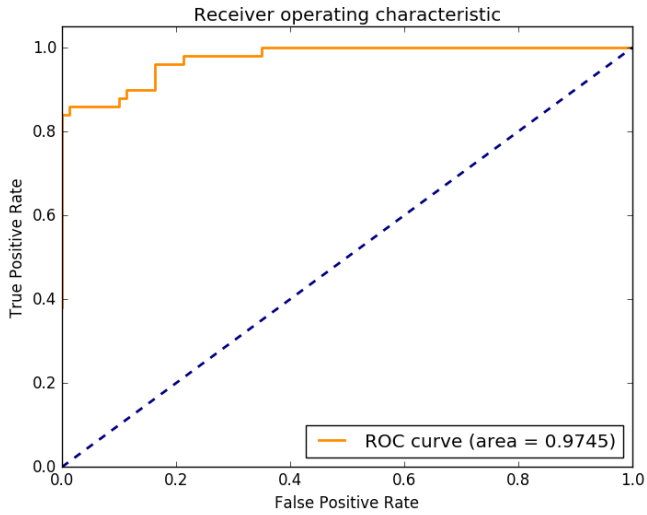


Figure 5.3: ROC curves. We use the RF-classifier to discriminate between positive and negative WSIs on the Cam16-test set obtaining an AUC-score of 0.9745. The true positive rate is equal to the sensitivity while the false positive rate is equal to 1-specificity. This comparison is only partly valid as the patch-based classification is validated on the same data. However, no WSIs in Cam16-test is used in the training data.

Rank	Affiliation	AUC score
1	Harvard Medical School and MIT, Method 3	0.9935
-	Google Brain [26]	0.9860
2	Harvard Medical School, Gordon Center for Medical Imaging, MGH, Method 3	0.9763
-	**Our method**	**0.9745**
3	Harvard Medical School, Gordon Center for Medical Imaging, MGH, Method 1	0.9650
4	The Chinese University of Hong Kong (CU lab, Hong Kong), Method 3	0.9415
5	Harvard Medical School and MIT, Method 1*	0.9234*

Table 5.10: Top 5 teams of CAMELYON16. From the above table it is possible to compare to other methods using the same Cam16-test set. We added the team from Google Brain's results together with our method for comparison. Our method performs very well on this task because there are no ITCs in the dataset. We refer to [34] for the official leader-board. *This submission was the winner of CAMELYON16 before the deadline.

5.4 Summary of results

In this chapter, we have presented our work on an algorithm for automated pN-stage evaluation for BCa patients. We showed how deep CNN models can automatically learn features that can discriminate between tumor and non-tumor image patches with 96.6% accuracy. The models were successfully implemented in a highly effective framework for analyzing WSIs. This framework is used to analyze 1400 WSIs of H&E stained lymph node sections corresponding to more than 4.5 TB of image data. Moreover, we developed a WSI-classifier that is able to detect most micro and macro metastases but has difficulties discriminating between WSIs with ITC and no metastases. This indicates that our method is overfitting on the training data for these two classes, hence other approaches should be used to handle ITCs separately. This is addressed in a discussion on ITCs in the chapter below. We submitted our method for external review on the IEEE ISBI CAMELYON17 Challenge test data set (100 patients), where we obtained a weighted kappa value of 0.8172, showing very good agreement between our algorithm and the human pathologist. Finally, we also tested our deep CNN model to discriminate between positive and negative WSIs on the CAMELYON16 test set excluding ITCs. Here we obtained an AUC-score of 0.9745, hence this automated solution holds great promise to reduce the workload of the pathologists on clinically relevant task.

CHAPTER 6

Discussion

Our general focus of this thesis was to use deep learning to detect breast cancer in H&E stained histopathology images. We have presented a novel approach to generate labelled H&E datasets in chapter 4 which we proved to be a feasible method for supervised training of deep CNNs. We showed how these models can be used when performing higher-level histopathology tasks. In addition to this, we trained a deep CNN to learn the histopathology patterns of lymph node metastases using manual annotations and proved that an algorithm based on such model can obtain state-of-the-art performance on a clinically relevant task. This chapter is devoted to a discussion of the general results and considerations using deep learning in histopathology together with future work revealed by this thesis.

6.1 Isolated tumor cells (ITCs)

ITCs is defined as single tumor cells or a cluster of tumor cells and is strictly not a metastasis, but is defined as one in the challenge [4]. However, lymph nodes containing only ITCs are not counted as positive lymph nodes in clinical practice but pathologists are required to report on ITCs when no macro metastases or micro metastases are detected in a patient's lymph nodes. For this, the presence or absence of ITCs must be confirmed by an IHC staining similar to PCK staining [4]. Our results from chapter 5 indicate that ITCs are very difficult to detect in H&E stained lymph node sections. The winning team of CAMELYON17 showed that it is possible to detect the larger ITCs as they have a higher score but we are critical about the possibility of detecting the smallest ITCs such as single tumor cells in an entire H&E section. As shown by others [35, 26] and our results, it is indeed plausible to screen lymph node sections for micro and macro metastases in H&E using deep learning. Therefore an interesting way to continue this work, could be to increase the sensitivity towards micro and macro metastases in our algorithm in order to remove as many WSIs as possible in H&E sections, and then use IHC-based IA to detect any possible ITCs in the negative slides stained with the proper IHC. This approach then combines both H&E- and IHC-based IA in a complementary manner that could be used to decrease the pathologist's workload without missing a single positive WSI. We leave it to future work to investigate combinatorial methods like these.

6.2 False labelling

The VDS alignment were generally successful but for some H&E and PCK aligned sections, the image registration results were unsatisfactory partly due to the H&E- and PCK-sections not being adjacent sections. Consequently, small tissue structures were often not represented in both sections. If a tumor region is present in the PCK section but located significantly different or not present at all in the H&E section, the automatic labelling produces false annotations in the H&E section. Even though we performed random sampling from giga-pixel images, the labelled training set was not perfectly clean. This influenced the training of CNNs as we simply showed wrongly labelled examples to the model. It also affected how we could validate the patch-based performance because a perfect score (100% accuracy) would practically not be the best performance as this would mean overfitting. The insufficient alignment also meant that we were not able to perform hard mining on this dataset which could have increased the patch-based performance.

The obvious solution to this issue is to strictly use adjacent tissue sections that are cut as thin as possible to increase the probability of cell and tissue structures being present in both sections. However, this was not an option for this thesis as the data were reused from a previous study, but we see it as an essential requirement for future studies. Another requirement regarding tissue preparation is careful placement of tissue sections on glass slides to avoid tissue folding, missing tissue etc. We found it necessary to disregard close to 20% of the dataset due to these artifacts.

False labelling also apply to the manual annotations in histopathology. First, the manual annotations must be treated as subjective and potentially biased towards the individual annotator, i.e. annotations should be obtained from multiple annotators and then averaged. Secondly, we observed annotations that were very sparse or in some cases missing in the CAMELYON17 Challenge dataset which also affected training and testing of CNNs. Others have proposed to crowd-source these tasks to a large cohort of non-pathologists [94] but this is still subject to human subjectivity. Therefore, we consider our automatic labelling using IHC VDS as a more suitable and scale-able alternative when the basics are done correctly.

6.3 H&E stain augmentation

In order to handle the H&E stain variations found in the data, we proposed a new data augmentation scheme that aims to vary the specific color vectors of the known dyes. We showed that it boosted the patch-based performance on both aspects of this thesis. The effects were largest on the lymph node metastases classification due to the inclusion of images from five different pathology laboratories. On the carcinoma data from Herlev, we only gained a small performance increase because the data originates from the same laboratory. Our method directly aims to learn the stain variation

between staining protocols instead of trying to remove it by color normalization techniques. For practical reasons and time considerations, the scheme is implemented in a simple manner where the stain intensities are perturbed very naively. We believe that our method can be optimized in many ways and is therefor an interesting research topic for the future. An interesting way to develop this method is to learn a H&E color model as part of the augmentation and then randomly generate color perturbations from the model comparable to learning spatial augmentation models [80]. Another aspect for validating the proposed method is to compare it to both stain normalization methods and random color perturbations, which has not been done in the thesis.

6.4 Imbalanced class distributions in histopathology

Another important part of training deep CNNs to learn histopathology patterns is how the sampling is performed to represent the distribution of tissue classes. In this thesis, we use offline sampling, i.e. patches are pre-sampled from each slide. This limits the extent of patches seen during training but we avoid biases towards WSIs with large tissue area and we can easily control the balance of classes in the training set. By contrast, recent approaches using online sampling [26] have shown to be advantageous to utilize the vast amount of patches available from giga-pixel WSIs. Here, patches are sampled randomly on-the-fly based on class probabilities which means that the CNN most likely never sees the same example twice. We see the sampling as a natural step to continue for this project as we believe this is essential for the convergence of CNNs.

The class imbalance between histopathology classes is challenging to capture during training. Especially, for the lymph node metastases as there are far more non-tumor than tumor patches. We used a 2:1 class balance to keep the sensitivity towards tumor patches high while still presenting more non-tumor patches. If the true balance between classes are used, the model would probably never learn the tumor class. However, we believe that there are better ways to capture this imbalance and ensure that the CNNs learn discriminate features well early in the training and then slowly learn the distribution of tumor and non-tumor. [24] recently proposed an elegant way for this by gradually increasing the misclassification loss for the non-tumor class. An alternative could be learning discriminative features from an equally balanced dataset and then lock the parameters of the convolutional layers before training continues on a dataset with the true class distribution. Again we leave this to future work to investigate methods like these.

6.5 On the usability of tumor heatmaps

In this thesis, we implemented the CNNs in framework that generates patch-based tumor heatmaps. These heatmaps are attractive for visualization purposes as they

show the spatial distribution of tumor probability. Another useful aspect of heatmaps is that they can be used in combination with other more traditional image analysis features for different purposes. However as shown by our results, this might not be the most precise method for image segmentation tasks due to threshold value. We tried to tune the threshold using the DSC which is a simple and practical solution. There exist more advanced ways of implementing CNN which could be very suitable such as fully-convolutional networks (FCN) [95] or the U-net architecture [96]. These two approaches use up-sampling/deconvolutional layers with transpose convolutions to produce the dense prediction maps instead of a patch probability, i.e. the spatial segmentation is learned as part of the network. Due to time constraints, we did not experiment with these methods even though such CNNs might have improved our segmentation results especially for the tumor estimation and outlining in chapter 4.

6.6 Validation using high-level tasks

Generally, we focused on the methodology around deep learning in this thesis but we implemented and validated our models as part of larger algorithms for automating histopathology tasks. as part of the original aim of this thesis, we investigated the agreement between our deep learning-based method and existing traditional computer vision methods on the clinical outcome for Ki67 assessment. And for the extended aim on lymph node metastases, we compare a high-level patient outcome based on both detection and classification of metastases instead of just measuring the patch-based performance or segmentation results. By doing so, some tumor regions could be missed without influencing the final outcome. This can seem very different from other computer vision domains but in histopathology, it can be difficult to get perfect results because the ground truth/golden standard can be vague and sometimes subjective. In this thesis, we tried to come up with methods to test that our CNNs actually solved the intended objective; namely image patch classification. Our best approach was to perform full sampling on the test WSIs to obtain the correct tissue and class distribution. Until better methods are proposed, we still believe that using higher-level tasks is the best way perform validation, especially for clinically oriented tasks.

6.7 Interpretation of deep learning models

Another aspect of using deep learning in histopathology which is not addressed specifically in this thesis is the interpretation of the learned features and decision rules. We experimented briefly with visualization of the deeper convolutional layers using maximization of kernel activation via backpropagation [97] which have been used to interpret features of natural images. We found that these showed very little insight into the inner workings of a deep CNN trained on histopathology images. The only interpretation we found similar was that the kernels of first layer were mostly color-based features, while the next layers seemed to have learned texture features. Even

though we cannot interpret the learned features of our models, we still know the relatively easy rules, which we tell the networks to follow during the learning process e.g. additions, multiplications etc. When we combine millions of these operations in a deep learning model, it is still very easy for the computer to understand but beyond human comprehension. Therefore, we are currently missing the tools for asking these predictive models how and why they came to the exact decision they did. This is especially relevant for medical imaging applications even if we can prove that our models have better performance than humans because these algorithms might be supporting diagnostics or treatment decisions. Regardless, we have shown that deep learning models are a powerful method in histopathology and believe they will be even more useful in the near future when more and more data becomes available for researchers.

6.8 Future work

There are many other approaches that would have been relevant for this project and we have already mentioned several directions that are not considered further in this thesis. Here we list some of the work that would be the natural next steps in future research.

- **Investigate H&E augmentation** - Improve the scheme by either building or learning a H&E color model which augments colors more correctly based on the physical processes of the staining and ensures that the maximum variability is covered. The scheme could also include more scanner settings than the gamma-correction, e.g. saturation.

- **Validation of proposed VDS-based method** - Our experiments were restricted to a small sample size with non-adjacent H&E and PCK sections. Setting up a controlled experiment using an optimal sectioning scheme would be necessary in order to validate and publish the proposed method.

- **Epithelial-stroma identification in Ki67 stained images** - The proposed training of deep learning models using VDS can be applied to many stain combinations. The most relevant for this project is to use the PCK tumor regions on the Ki67 section, and then learn the tumor-stroma classification directly in the IHC-biomarker. Hence, there is no need for tumor outlining in H&E or PCK.

- **Discriminate between ITCs and normal slides independently** - As discussed earlier, it is necessary to handle ITCs separately from macro and micro. The 3rd place of the CAMELYON17 Challenge used a simple 99% threshold classification, which could be an interesting approach to use in combination with our existing method.

- **Implement online sampling** - Offline sampling led to a limited amount of patches during training, where online sampling would be an integral part of any future work.
- **Convert models into FCNs** - The CNNs used in this thesis could be converted into FCN as described in [95]. This approach has several advantages with regard to the segmentation problems but it would also lead to a significant increase in inference.
- **Ensemble models** - A popular approach to improve DNNs is to use an ensemble multiple models where the prediction is averaged across different models. This would also be an interesting way continue the work of this thesis as many of the top performing teams in CAMELYON16 and CAMELYON17 implemented this in their submissions. However, a drawback of ensemble models is that the computational cost increases for both training and WSI inference.
- **Investigate loss functions for learning histopathology class distributions** - As discussed earlier, one must ensure that the CNNs learn discriminate features well early in the training and then slowly learn the distribution of histopathology class distributions. The influence of different schemes for the loss could be an interesting research topic for future projects.
- **Investigate smaller networks** - We experimented to a very limited extent with smaller networks than the most popular models from the ImageNet competition. Future work should investigate the possibility of building smaller optimized networks for histopathology. One network is the U-net model [96] for segmentation tasks which also has far fewer parameters than the models used in this thesis.
- **Cell-based approach for metastases** - An interesting approach to the CAMELYON17 Challenge is to develop a tumor cell classifier that detects and segments tumor cells instead of larger tumor regions e.g. by using [98]. This could possibly also increase the sensitivity towards ITCs.

CHAPTER 7

Conclusion

The contribution of this thesis is threefold, (i) development of a stain specific data augmentation scheme that enables deep convolutional neural networks to learn H&E stain variations, (ii) proposition of a novel approach to obtain large-scale labelled datasets for deep learning models in digital pathology, and (iii) development of a deep learning based model for automatic detection and staging of lymph nodes metastases in BCa.

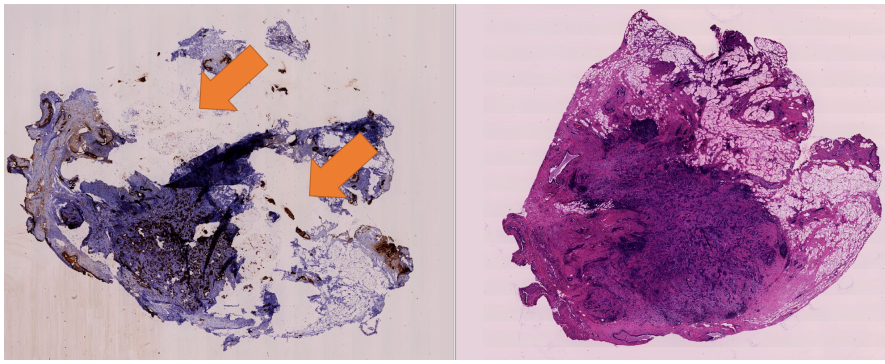
First, the developed augmentation method is capable of improving the performance of tissue-based cancer detection by forcing models to disregard color variability that is irrelevant to the classification. Thus, the proposed framework makes the algorithm more robust than only training on the stain variation available in the existing data. Our method could improve the deployment of algorithms to multiple medical centers by removing the need to tune algorithms for each specific site.

Secondly, the proposed labelling approach is capable of transferring specific IHC information to routine H&E sections creating objectively labelled data on a cheaper histopathology staining. This substantially decreases the need for subjective and costly manual labelling that is currently necessary for developing successful deep convolutional neural networks for computational pathology. We proved that our approach could be used to develop a deep learning based cancer detection in H&E which has promise for lowering costs of IHC-biomarker quantification in BCa diagnostics.

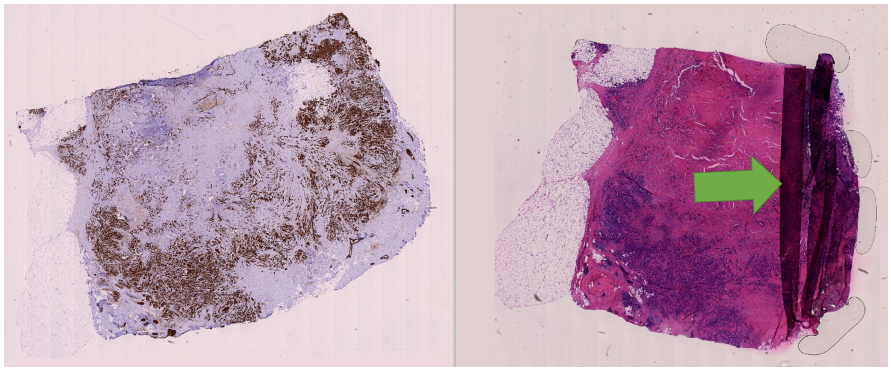
Lastly, our method for detection and staging of lymph node metastases showed state-of-the-art performance on a difficult time-consuming, but clinically relevant task. The developed deep learning model is capable of learning histopathological differences between tumor and normal tissue in lymph nodes from a vast amount of labelled examples. Hence, this automated solution holds great promise to reduce the workload of pathologists while reducing the subjectivity of BCa diagnosis. The future work generated from this thesis will focus on utilizing the combination of H&E and IHC methods to increase performance and exploit larger datasets.

APPENDIX A

Removed WSIs



(a) Missing tissue

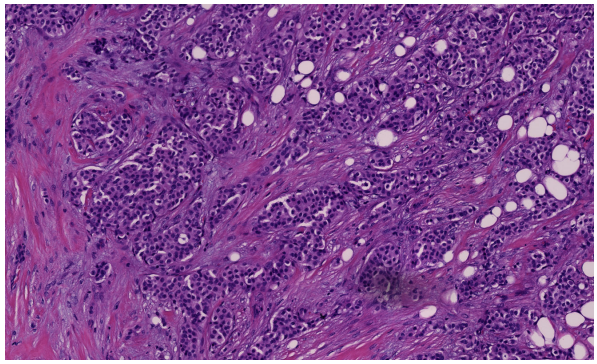


(b) Tissue folding

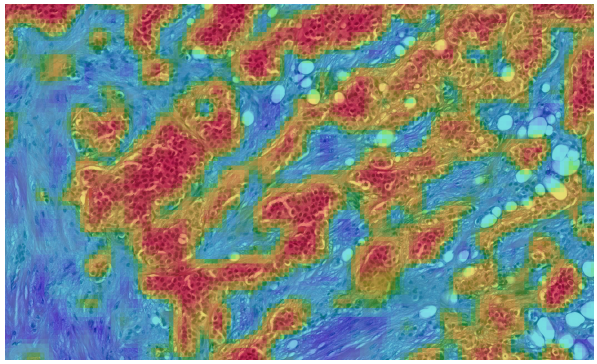
Figure A.1: Removed WSIs from Carcinoma data. In (a), the orange arrows show tissue missing in the PCK section. In (b), the green arrow shows tissue folding in the H&E section. These artifacts would not make sense to align, hence we removed such WSI collections from the dataset.

APPENDIX B

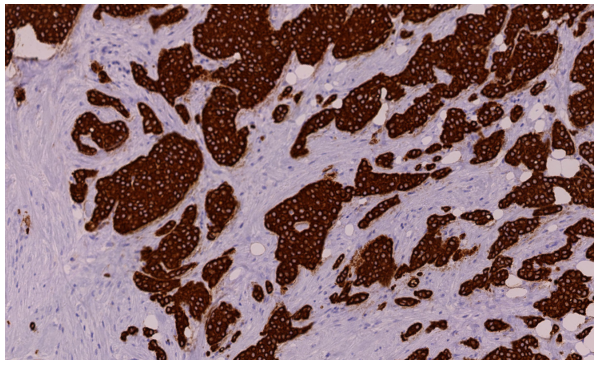
Qualitative results of WSI inference on carcinoma



(a) H&E section

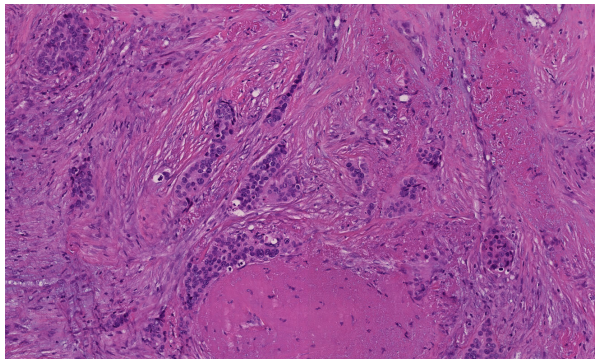


(b) Tumor heatmap

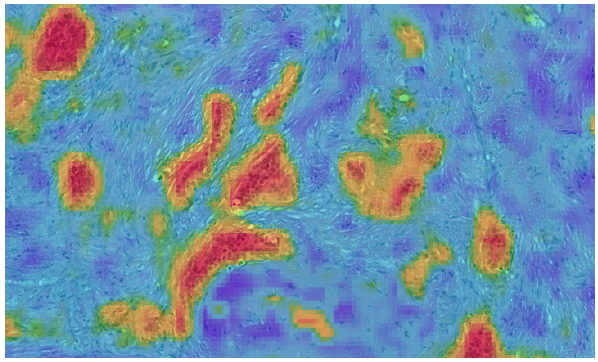


(c) PCK section

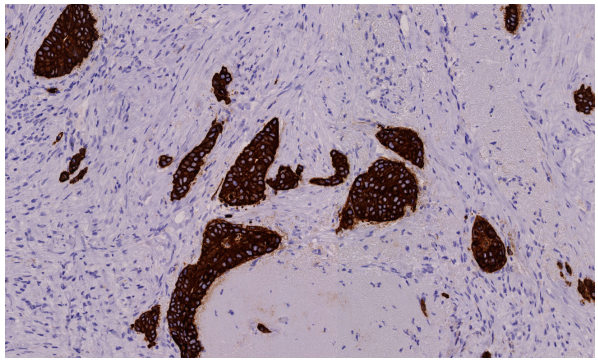
Figure B.1: Tumor heatmap and PCK section at 5×.



(a) H&E section

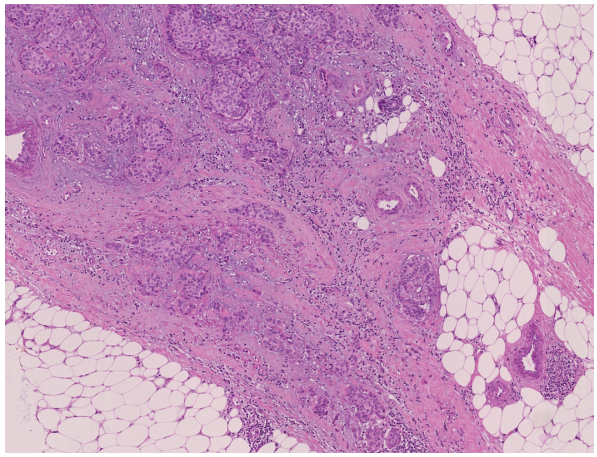


(b) Tumor heatmap

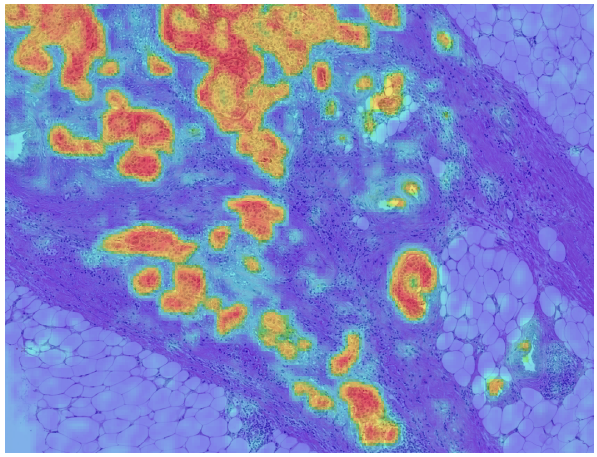


(c) PCK section

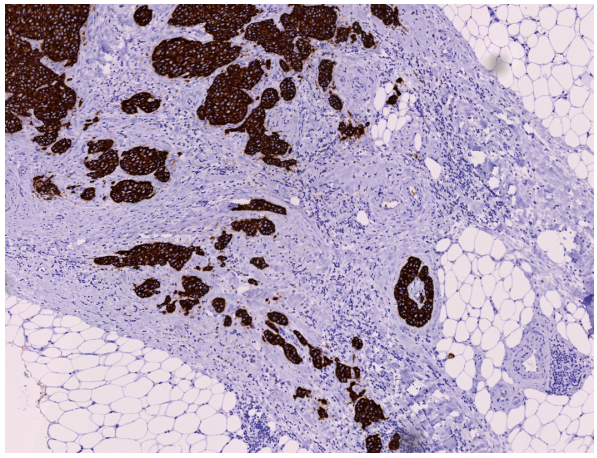
Figure B.2: Tumor heatmap and PCK section at 5×.



(a) H&E section



(b) Tumor heatmap

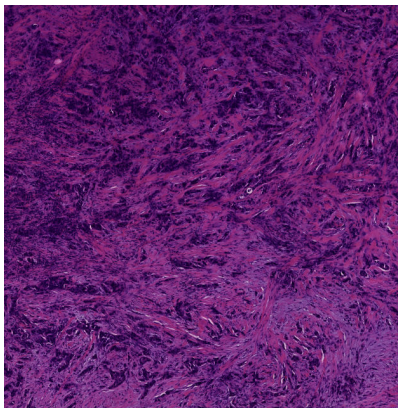


(c) PCK section

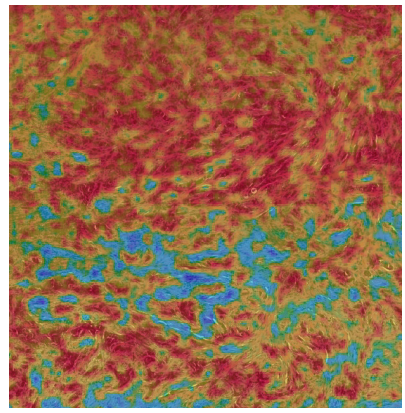
Figure B.3: Tumor heatmap and PCK section at 2.5×.

APPENDIX C

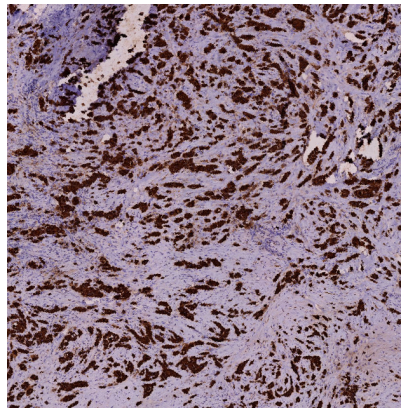
Outliers from discovered from Bland-Altman plot



(a) H&E section

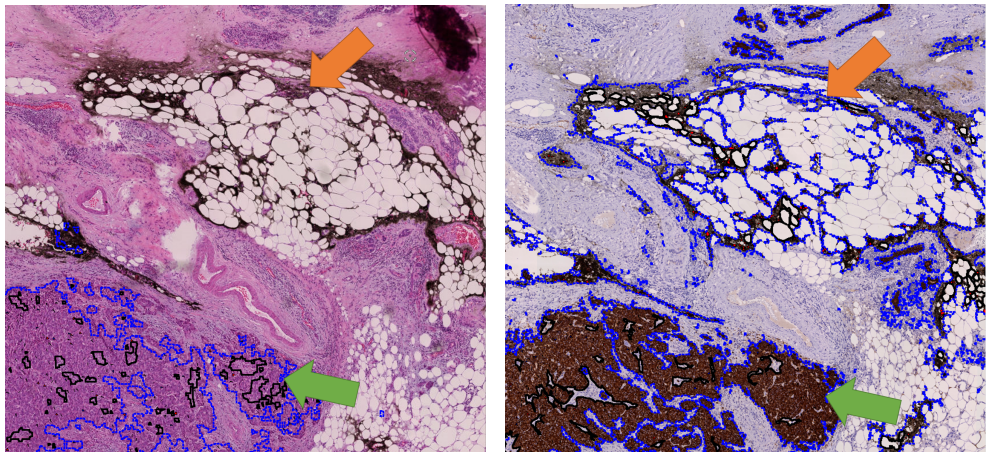


(b) Tumor heatmap



(c) PCK section

Figure C.1: Example of percentage tumor outlier from the test set at 2.5×. Notice how the H&E tumor heatmap cannot segment the fine smaller PCK regions in (c).

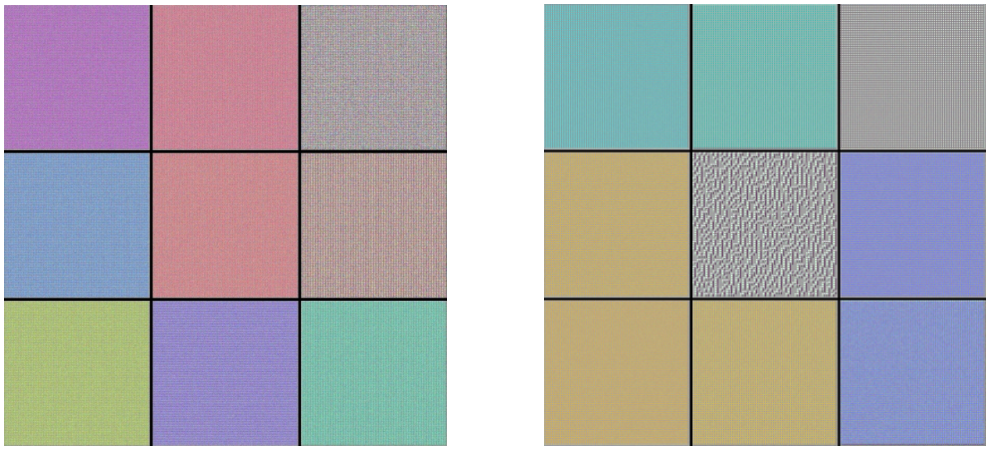


(a) H&E section **(b)** PCK section

Figure C.2: Example of outlier from the test set at 2.5×. The blue arrows show correctly identified tumor regions on both (a) and (b). The orange arrows show wrongly outlined tumor region in the PCK section (b), where the same error is not seen in the H&E section (a). The artifact is probably due to preparation errors before digitization. This shows that the deep CNN classifies regions based on more than stain intensities. Therefore, our method is less vulnerable against staining artifacts.

APPENDIX D

Visualization of first convolutional layer



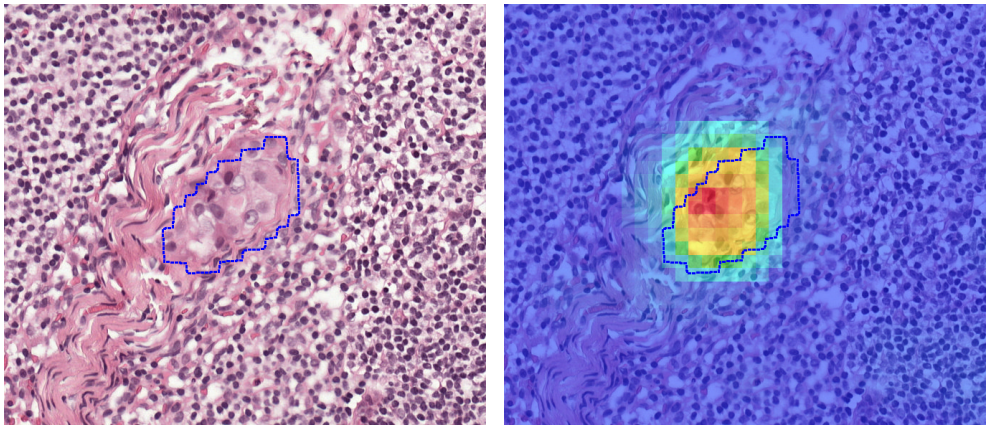
(a) No H&E-augmentation

(b) With H&E-augmentation

Figure D.1: Kernel visualization of the first convolutional layer. To visualize convolutional kernels, we use find the RGB-image that maximize the kernel activations using backpropagation [97]. For both (a) and (b), we show the 9 most activated kernel of the first layer (total = 32). In (a), we can see that color nuance of H&E (purple/dark blue and pink) are present and used heavily. In (b) however, these colors are not being used in the kernels - especially not the pink nuance. We interpret this as the CNN model are not using the H&E-stain intensities to classify tumor and non-tumor, which is the goal of using stain/color augmentation.

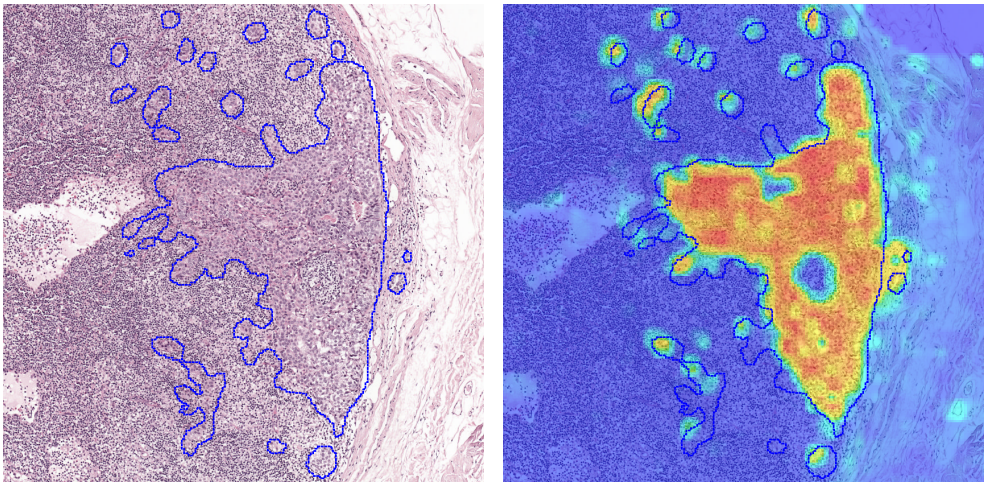
APPENDIX E

Qualitative results of WSI inference on metastases



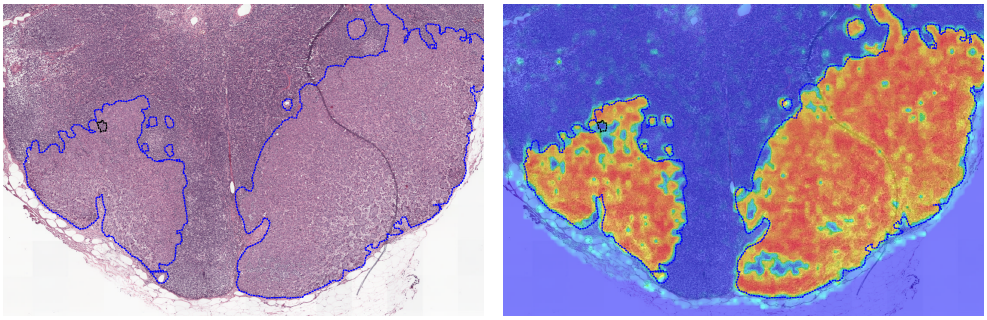
(a) Ground truth **(b)** Tumor heatmap

Figure E.1: ITC. Tumor heatmap of ITC at 20×.



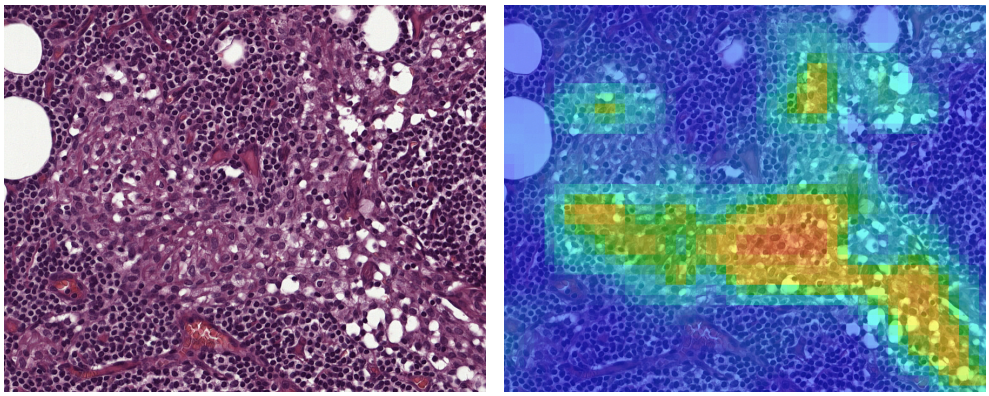
(a) Ground truth **(b)** Tumor heatmap

Figure E.2: Micro. Tumor heatmap of micro metastases at 5×.



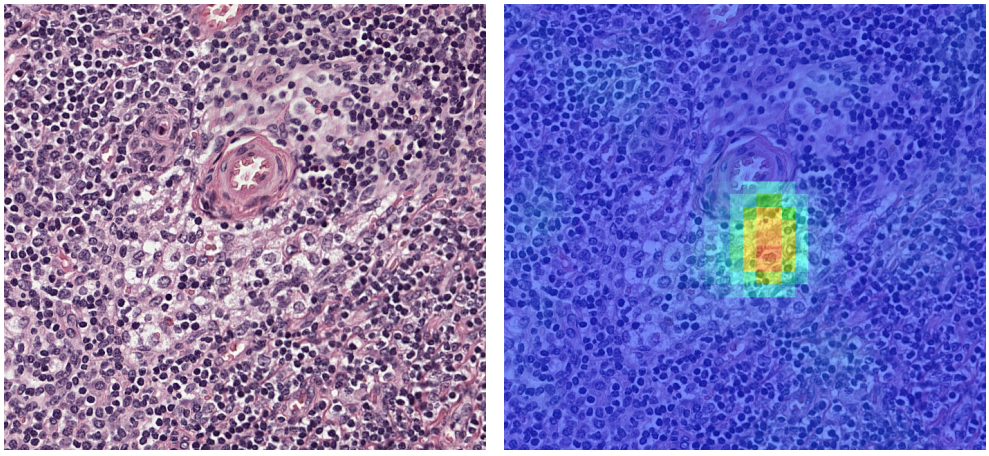
(a) Ground truth **(b)** Tumor heatmap

Figure E.3: Macro. Tumor heatmap of macro metastases at 2.5×.



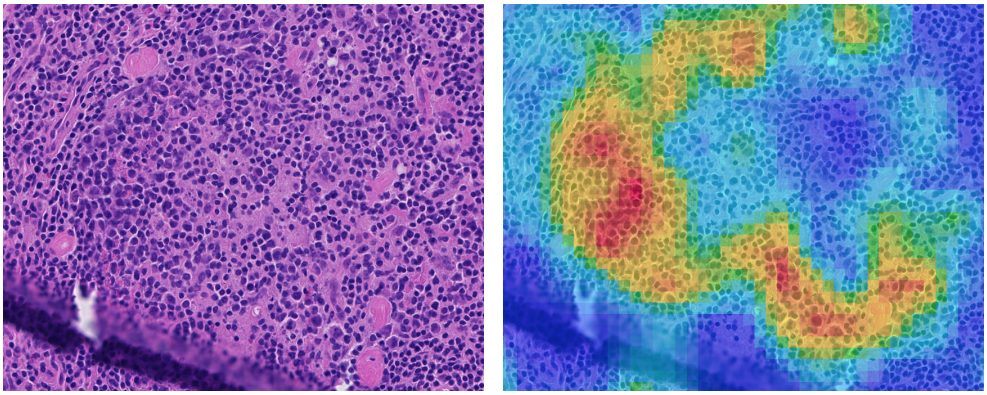
(a) Ground truth (normal tissue) **(b)** Tumor heatmap

Figure E.4: False positive ITC. Tumor heatmap of false positive ITC at 10×.



(a) Ground truth (normal tissue) **(b)** Tumor heatmap

Figure E.5: False positive ITC. Tumor heatmap of false positive ITC at 20×.



(a) Ground truth (normal tissue) **(b)** Tumor heatmap

Figure E.6: False positive Micro. Tumor heatmap of false positive micro metastasis at 10×.

APPENDIX F

Confusion matrices for slide-level and patient-level classification

	Normal	**ITC**	**Micro**	**Macro**
Normal	162	0	0	2
ITC	11	20	0	0
Micro	8	0	34	2
Macro	1	0	2	58

Table F.1: Confusion matrix for the slide-level classification task on the training set. Our predictions and ground truth shown are horizontally and vertically, respectively.

	Normal	**ITC**	**Micro**	**Macro**
Normal	136	0	9	4
ITC	4	0	0	0
Micro	3	0	14	3
Macro	0	0	2	25

Table F.2: Confusion matrix for the slide-level classification task on the validation set. ur predictions and ground truth shown are horizontally and vertically, respectively.

	pN0	**pN0(i+)**	**pN1mi**	**pN1**	**pN2**
pN0	7	0	0	1	0
pN0(i+)	1	9	0	0	0
pN1mi	1	0	10	1	0
pN1	0	0	0	16	0
pN2	0	0	0	2	12

Table F.3: Confusion matrix for the pN-stage classification task on the training set. ur predictions and ground truth shown are horizontally and vertically, respectively.

	pN0	**pN0(i+)**	**pN1mi**	**pN1**	**pN2**
pN0	14	0	0	2	0
pN0(i+)	2	0	0	0	0
pN1mi	1	0	5	1	1
pN1	0	0	0	8	1
pN2	0	0	0	2	3

Table F.4: Confusion matrix for the pN-stage classification task on the validation set. Our predictions and ground truth shown are horizontally and vertically, respectively.

Bibliography

[1] Jacques Ferlay, Isabelle Soerjomataram, Rajesh Dikshit, Sultan Eser, Colin Mathers, Marise Rebelo, Donald Maxwell Parkin, David Forman, and Freddie Bray. "Cancer incidence and mortality worldwide: sources, methods and major patterns in GLOBOCAN 2012". In: *International journal of cancer* 136.5 (2015), E359–E386.

[2] Kræftens Bekæmpelse. *Statistik om brystkræft*. 2017. URL: `https://www.cancer.dk/brystkraeft-mammacancer/statistik-brystkraeft/` (visited on June 14, 2017).

[3] Peter W Hamilton, Yinhai Wang, Clinton Boyd, Jacqueline A James, Maurice B Loughrey, Joseph P Hougton, David P Boyle, Paul Kelly, Perry Maxwell, David McCleary, et al. "Automated tumor analysis for molecular profiling in lung cancer". In: *Oncotarget* 6.29 (2015), pages 27938–27952.

[4] Oscar Geessink, Péter Bándi, Geert Litjens, and Jeroen van der Laak. *CAMELYON17: Grand challenge on cancer metastasis detection and classification in lymph nodes*. 2017. URL: `https://camelyon17.grand-challenge.org` (visited on April 5, 2017).

[5] American Cancer Society. "Breast Cancer Facts & Figures 2015-2016". In: *Atlanta: American Cancer Society, Inc.* (2015).

[6] Niels Marcussen, Flemming Brandt Sørensen, Susanne Holck, and Torben Steiniche. *Patologi*. 1. edition. FADL's Forlag, 2010. ISBN: 9788777495434.

[7] American Breast Cancer Foundation. *Female Breast Anatomy*. 2016. URL: `http://www.abcf.org/assets/uploads/FemaleBreastAnatomy.jpg` (visited on June 18, 2017).

[8] Clive R. Taylor and Lars Rudbeck. *Immunohistochemical Staining Methods, Sixth Edition*. 2016. URL: `http://www.agilent.com/cs/library/technicaloverviews/public/08002_ihc_staining_methods.pdf` (visited on June 18, 2017).

[9] John D Bancroft and Marilyn Gamble. *Theory and practice of histological techniques*. Elsevier Health Sciences, 2008.

[10] Babak Ehteshami Bejnordi, Geert Litjens, Nadya Timofeeva, Irene Otte-Höller, André Homeyer, Nico Karssemeijer, and Jeroen AWM van der Laak. "Stain specific standardization of whole-slide histopathological images". In: *IEEE transactions on medical imaging* 35.2 (2016), pages 404–415.

[11] Adnan Mujahid Khan, Nasir Rajpoot, Darren Treanor, and Derek Magee. "A nonlinear mapping approach to stain normalization in digital histopathology images using image-specific color deconvolution". In: *IEEE Transactions on Biomedical Engineering* 61.6 (2014), pages 1729–1738.

[12] Marc Macenko, Marc Niethammer, JS Marron, David Borland, John T Woosley, Xiaojun Guan, Charles Schmitt, and Nancy E Thomas. "A method for normalizing histology slides for quantitative analysis". In: *Biomedical Imaging: From Nano to Macro, 2009. ISBI'09. IEEE International Symposium on*. IEEE. 2009, pages 1107–1110.

[13] Erik Reinhard, Michael Adhikhmin, Bruce Gooch, and Peter Shirley. "Color transfer between images". In: *IEEE Computer graphics and applications* 21.5 (2001), pages 34–41.

[14] CR Taylor and Richard M Levenson. "Quantification of immunohistochemistry—issues concerning methods, utility and semiquantitative assessment II". In: *Histopathology* 49.4 (2006), pages 411–424.

[15] Thomas Scholzen and Johannes Gerdes. "The Ki-67 protein: from the known and the unknown". In: *Journal of cellular physiology* 182.3 (2000), pages 311–322.

[16] R Røge, R Riber-Hansen, S Nielsen, and Mogens Vyberg. "Validation Of Virtual Double Staining For Estimation Of Ki67 Proliferation Indices In Breast Carcinomas". In: *Diagnostic Pathology* 1.8 (2016).

[17] Visiopharm. *Virtual Double Staining*. 2016. URL: `https://www.visiopharm.com/module/virtualdoublestaining` (visited on June 18, 2017).

[18] M. Grunkin, S.T. Rasmussen, K.A. Bjerrum, and J.D. Hansen. *Feature-based registration of sectional images*. US Patent 8,682,050. March 2014. URL: `https://www.google.ch/patents/US8682050`.

[19] Nina Lykkegaard Andersen, Anja Brügmann, Giedrius Lelkaitis, Søren Nielsen, Michael Friis Lippert, and Mogens Vyberg. "Virtual Double Staining: A Digital Approach to Immunohistochemical Quantification of Estrogen Receptor Protein in Breast Carcinoma Specimens." In: *Applied Immunohistochemistry & Molecular Morphology* (2017).

[20] Fred L. Bookstein. "Principal warps: Thin-plate splines and the decomposition of deformations". In: *IEEE Transactions on pattern analysis and machine intelligence* 11.6 (1989), pages 567–585.

[21] Visiopharm A/S. *CE IVD Ki67 APP*. 2017. URL: `https://www.visiopharm.com/app-center/318-10004-ki-67-breast-cancer` (visited on June 15, 2017).

[22] Danish Breast Cancer Cooperative Group (DBCG). *Recommendation of from DBCG on pathology*. URL: `http://www.dbcg.dk/PDF%5C%20Filer/Kap_3_Patologi_28.05.2015.pdf` (visited on June 21, 2017).

[23] Visiopharm A/S. *CE IVD PCK VDS APP*. 2017. URL: `https://www.visiopharm.com/app-center/355-20002-pck-vds-tumor-detection` (visited on June 15, 2017).

[24] Babak Ehteshami Bejnordi, Jimmy Linz, Ben Glass, Maeve Mullooly, Gretchen L Gierach, Mark E Sherman, Nico Karssemeijer, Jeroen van der Laak, and Andrew H Beck. "Deep learning-based assessment of tumor-associated stroma for diagnosing breast cancer in histopathology images". In: *arXiv preprint arXiv:1702.05803* (2017).

[25] Babak Ehteshami Bejnordi, Guido Zuidhof, Maschenka Balkenhol, Meyke Hermsen, Peter Bult, Bram van Ginneken, Nico Karssemeijer, Geert Litjens, and Jeroen van der Laak. "Context-aware stacked convolutional neural networks for classification of breast carcinomas in whole-slide histopathology images". In: *arXiv preprint arXiv:1705.03678* (2017).

[26] Yun Liu, Krishna Gadepalli, Mohammad Norouzi, George E Dahl, Timo Kohlberger, Aleksey Boyko, Subhashini Venugopalan, Aleksei Timofeev, Philip Q Nelson, Greg S Corrado, et al. "Detecting Cancer Metastases on Gigapixel Pathology Images". In: *arXiv preprint arXiv: 1703.02442* (2017).

[27] Angel Cruz-Roa, Hannah Gilmore, Ajay Basavanhally, Michael Feldman, Shridar Ganesan, Natalie NC Shih, John Tomaszewski, Fabio A González, and Anant Madabhushi. "Accurate and reproducible invasive breast cancer detection in whole-slide images: A Deep Learning approach for quantifying tumor extent". In: *Scientific Reports* 7 (2017), page 46450.

[28] Harshita Sharma, Norman Zerbe, Iris Klempert, Olaf Hellwich, and Peter Hufnagl. "Deep convolutional neural networks for automatic classification of gastric carcinoma using whole slide images in digital histopathology". In: *Computerized Medical Imaging and Graphics* (2017).

[29] Péter Bándi, Rob van de Loo, Milad Intezar, Daan Geijs, Francesco Ciompi, Bram van Ginneken, Jeroen van der Laak, and Geert Litjens. "Comparison of Different Methods for Tissue Segmentation in Histopathological Whole-Slide Images". In: *arXiv preprint arXiv:1703.05990* (2017).

[30] Dan C Cireşan, Alessandro Giusti, Luca M Gambardella, and Jürgen Schmidhuber. "Mitosis detection in breast cancer histology images with deep neural networks". In: *International Conference on Medical Image Computing and Computer-assisted Intervention*. Springer. 2013, pages 411–418.

[31] Haibo Wang, Angel Cruz-Roa, Ajay Basavanhally, Hannah Gilmore, Natalie Shih, Mike Feldman, John Tomaszewski, Fabio Gonzalez, and Anant Madabhushi. "Mitosis detection in breast cancer pathology images by combining handcrafted and convolutional neural network features". In: *Journal of Medical Imaging* 1.3 (2014), pages 034003–034003.

[32] Angel Cruz-Roa, Ajay Basavanhally, Fabio González, Hannah Gilmore, Michael Feldman, Shridar Ganesan, Natalie Shih, John Tomaszewski, and Anant Madabhushi. "Automatic detection of invasive ductal carcinoma in whole slide images with convolutional neural networks". In: *SPIE medical imaging*. International Society for Optics and Photonics. 2014, pages 904103–904103.

[33] Geert Litjens, Clara I Sánchez, Nadya Timofeeva, Meyke Hermsen, Iris Nagtegaal, Iringo Kovacs, Christina Hulsbergen-Van De Kaa, Peter Bult, Bram Van Ginneken, and Jeroen Van Der Laak. "Deep learning as a tool for increased accuracy and efficiency of histopathological diagnosis". In: *Scientific reports* 6 (2016).

[34] Babak Ehteshami Bejnordi and Jeroen van der Laak. *CAMELYON16: Grand challenge on cancer metastasis detection in lymph nodes*. 2016. URL: `https://camelyon16.grand-challenge.org` (visited on April 5, 2017).

[35] Dayong Wang, Aditya Khosla, Rishab Gargeya, Humayun Irshad, and Andrew H Beck. "Deep learning for identifying metastatic breast cancer". In: *arXiv preprint arXiv: 1606.05718* (2016).

[36] Riku Turkki, Nina Linder, Panu E Kovanen, Teijo Pellinen, and Johan Lundin. "Antibody-supervised deep learning for quantification of tumor-infiltrating immune cells in hematoxylin and eosin stained breast cancer samples". In: *Journal of Pathology Informatics* 7 (2016).

[37] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. "Deep Learning". Book in preparation for MIT Press. 2016. URL: `http://www.deeplearningbook.org`.

[38] Warren S McCulloch and Walter Pitts. "A logical calculus of the ideas immanent in nervous activity". In: *The bulletin of mathematical biophysics* 5.4 (1943), pages 115–133.

[39] Donald Olding Hebb. *The organization of behavior: A neuropsychological theory*. Psychology Press, 2005.

[40] Frank Rosenblatt. "The perceptron: A probabilistic model for information storage and organization in the brain." In: *Psychological review* 65.6 (1958), page 386.

[41] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen AWM van der Laak, Bram van Ginneken, and Clara I Sánchez. "A survey on deep learning in medical image analysis". In: *arXiv preprint arXiv:1702.05747* (2017).

[42] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[43] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. "Backpropagation applied to handwritten zip code recognition". In: *Neural computation* 1.4 (1989), pages 541–551.

[44] Nando de Freitas. *Lecture notes, Machine Learning Course: 2014-2015, Department of Computer Science, University of Oxford.* URL: `https://www.cs.ox.ac.uk/teaching/courses/2014-2015/ml/index.html` (visited on June 13, 2017).

[45] James Bergstra, Olivier Breuleux, Pascal Lamblin, Razvan Pascanu, Olivier Delalleau, Guillaume Desjardins, Ian Goodfellow, Arnaud Bergeron, Yoshua Bengio, and Pack Kaelbling. "Theano: Deep learning on gpus with python". In: (2011).

[46] Ashia C Wilson, Rebecca Roelofs, Mitchell Stern, Nathan Srebro, and Benjamin Recht. "The Marginal Value of Adaptive Gradient Methods in Machine Learning". In: *arXiv preprint arXiv:1705.08292* (2017).

[47] Yurii Nesterov. "A method of solving a convex programming problem with convergence rate O (1/k2)". In: *Soviet Mathematics Doklady.* Volume 27. 2. 1983, pages 372–376.

[48] Boris T Polyak. "Some methods of speeding up the convergence of iteration methods". In: *USSR Computational Mathematics and Mathematical Physics* 4.5 (1964), pages 1–17.

[49] Jiquan Ngiam, Adam Coates, Ahbik Lahiri, Bobby Prochnow, Quoc V Le, and Andrew Y Ng. "On optimization methods for deep learning". In: *Proceedings of the 28th International Conference on Machine Learning (ICML-11).* 2011, pages 265–272.

[50] Geoffrey Hinton, Nirsh Srivastava, and Kevin Swersky. "Neural Networks for Machine Learning Lecture 6a Overview of mini–batch gradient descent". In: ().

[51] Diederik Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[52] Sebastian Ruder. "An overview of gradient descent optimization algorithms". In: *arXiv preprint arXiv:1609.04747* (2016).

[53] François Chollet et al. *Keras.* `https://github.com/fchollet/keras`. 2015.

[54] Ilya Sutskever, James Martens, George E Dahl, and Geoffrey E Hinton. "On the importance of initialization and momentum in deep learning." In: *ICML (3)* 28 (2013), pages 1139–1147.

[55] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification". In: *Proceedings of the IEEE International Conference on Computer Vision.* 2015, pages 1026–1034.

[56] Xavier Glorot and Yoshua Bengio. "Understanding the difficulty of training deep feedforward neural networks." In: *Aistats.* Volume 9. 2010, pages 249–256.

[57] Andrej Kaparthy et al. *Stanford CS class CS231n: Convolutional Neural Networks for Visual Recognition.* URL: `http://cs231n.github.io/` (visited on June 11, 2017).

[58] David H Hubel and Torsten N Wiesel. “Receptive fields and functional architecture of monkey striate cortex”. In: *The Journal of physiology* 195.1 (1968), pages 215–243.

[59] Kunihiko Fukushima. “Neocognitron: A hierarchical neural network capable of visual pattern recognition”. In: *Neural networks* 1.2 (1988), pages 119–130.

[60] B Boser Le Cun, John S Denker, D Henderson, Richard E Howard, W Hubbard, and Lawrence D Jackel. “Handwritten digit recognition with a back-propagation network”. In: *Advances in neural information processing systems*. Citeseer. 1990.

[61] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems*. 2012, pages 1097–1105.

[62] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. “Going deeper with convolutions”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pages 1–9.

[63] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pages 770–778.

[64] Karen Simonyan and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556* (2014).

[65] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. “ImageNet Large Scale Visual Recognition Challenge”. In: *International Journal of Computer Vision (IJCV)* 115.3 (2015), pages 211–252. DOI: 10.1007/s11263-015-0816-y.

[66] George E Dahl, Tara N Sainath, and Geoffrey E Hinton. “Improving deep neural networks for LVCSR using rectified linear units and dropout”. In: *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE. 2013, pages 8609–8613.

[67] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. “Empirical evaluation of rectified activations in convolutional network”. In: *arXiv preprint arXiv:1505.00853* (2015).

[68] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. “Fast and accurate deep network learning by exponential linear units (elus)”. In: *arXiv preprint arXiv:1511.07289* (2015).

[69] Vinod Nair and Geoffrey E Hinton. “Rectified linear units improve restricted boltzmann machines”. In: *Proceedings of the 27th international conference on machine learning (ICML-10)*. 2010, pages 807–814.

[70] Matthew D Zeiler, M Ranzato, Rajat Monga, Min Mao, Kun Yang, Quoc Viet Le, Patrick Nguyen, Alan Senior, Vincent Vanhoucke, Jeffrey Dean, et al. "On rectified linear units for speech processing". In: *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE. 2013, pages 3517–3521.

[71] Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, and Gang Wang. "Recent advances in convolutional neural networks". In: *arXiv preprint arXiv:1512.07108* (2015).

[72] Y-Lan Boureau, Jean Ponce, and Yann LeCun. "A theoretical analysis of feature pooling in visual recognition". In: *Proceedings of the 27th international conference on machine learning (ICML-10)*. 2010, pages 111–118.

[73] Rishi Kumar Samer Hijazi and Chris Rowen. *Using Convolutional Neural Networks for Image Recognition, IP Group, Cadence*. URL: `https://ip.cadence.com/uploads/901/cnn_wp-pdf` (visited on June 13, 2017).

[74] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. "Dropout: A simple way to prevent neural networks from overfitting." In: *Journal of Machine Learning Research* 15.1 (2014), pages 1929–1958.

[75] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. "Rethinking the inception architecture for computer vision". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pages 2818–2826.

[76] Min Lin, Qiang Chen, and Shuicheng Yan. "Network in network". In: *arXiv preprint arXiv:1312.4400* (2013).

[77] Sergey Ioffe and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: *arXiv preprint arXiv:1502.03167* (2015).

[78] Jon Shlens. *Train your own image classfier with TensorFlow, Google Research blog*. 2016. URL: `https://research.googleblog.com/2016/03/train-your-own-image-classifier-with.html` (visited on March 19, 2017).

[79] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Identity mappings in deep residual networks". In: *European Conference on Computer Vision*. Springer. 2016, pages 630–645.

[80] Søren Hauberg, Oren Freifeld, Anders Boesen Lindbo Larsen, John Fisher, and Lars Hansen. "Dreaming more data: Class-dependent distributions over diffeomorphisms for learned data augmentation". In: *Artificial Intelligence and Statistics*. 2016, pages 342–350.

[81] Francesco Ciompi, Oscar Geessink, Babak Ehteshami Bejnordi, Gabriel Silva de Souza, Alexi Baidoshvili, Geert Litjens, Bram van Ginneken, Iris Nagtegaal, and Jeroen van der Laak. "The importance of stain normalization in colorectal tissue classification with convolutional networks". In: *arXiv preprint arXiv:1702.05931* (2017).

[82] Jeroen AWM van der Laak, Martin MM Pahlplatz, Antonius GJM Hanselaar, and Peter de Wilde. "Hue-saturation-density (HSD) model for stain recognition in digital images from transmitted light microscopy". In: *Cytometry* 39.4 (2000), pages 275–284.

[83] Babak Ehteshami Bejnordi, Nadya Timofeeva, Irene Otte-Höller, Nico Karssemeijer, and Jeroen AWM van der Laak. "Quantitative analysis of stain variability in histology slides and an algorithm for standardization". In: *SPIE Medical Imaging*. International Society for Optics and Photonics. 2014, pages 904108–904108.

[84] *HistomicsTK Python API*. 2017. URL: `https://github.com/DigitalSlideArchive/HistomicsTK` (visited on March 19, 2017).

[85] Poynton Charles et al. "Digital video and HDTV: Algorithms and interfaces". In: *Morgan Kaufmann Publishers, San Francisco* (2003), page 260.

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

[87] Mary L McHugh. "Interrater reliability: the kappa statistic". In: *Biochemia medica* 22.3 (2012), pages 276–282.

[88] Stat Trek. *Hypothesis Test for Regression Slope*. 2017. URL: `http://stattrek.com/regression/slope-test.aspx` (visited on June 22, 2017).

[89] R Haydn, GW Dalke, J Henkel, and JE Bare. "Application of the IHS color transform to the processing of multisensor data and image enhancement". In: *Proceedings of the International Symposium on Remote Sensing of Environment, First Thematic Conference: Remote sensing of arid and semi-arid lands, 19-25 January, 1982, Cairo, Egypt*. Ann Arbor, Mich.: Center Remote Sens. Information & Analysis, Environ. Res. Inst., Mich., 1982. 1982.

[90] Theano Development Team. "Theano: A Python framework for fast computation of mathematical expressions". In: *arXiv e-prints* abs/1605.02688 (May 2016). URL: `http://arxiv.org/abs/1605.02688`.

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

[92] American Joint Committee on Cancer. *Breast Medium*. 2016. URL: `https://cancerstaging.org/references-tools/quickreferences/Documents/BreastMedium.pdf` (visited on March 19, 2017).

[93] DG Altman. “Inter-rater agreement”. In: *Practical statistics for medical research* 5 (1991), pages 403–409.

[94] Humayun Irshad, Eun-Yeong Oh, Daniel Schmolze, Liza M Quintana, Laura Collins, Rulla M Tamimi, and Andrew H Beck. “Crowdsourcing scoring of immunohistochemistry images: Evaluating Performance of the Crowd and an Automated Computational Method”. In: *Scientific Reports* 7 (2017).

[95] Jonathan Long, Evan Shelhamer, and Trevor Darrell. “Fully convolutional networks for semantic segmentation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pages 3431–3440.

[96] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-net: Convolutional networks for biomedical image segmentation”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2015, pages 234–241.

[97] Francois Chollet. *How convolutional neural networks see the world*. 2016. URL: `https://blog.keras.io/how-convolutional-neural-networks-see-the-world.html` (visited on March 19, 2017).

[98] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. “Mask r-cnn”. In: *arXiv preprint arXiv:1703.06870* (2017).

{2.7182818284}ϑφερτυθιοπσδφγηξκλ