University of
Stavanger

FACULTY OF SCIENCE AND TECHNOLOGY

MASTER'S THESIS

Study programme/specialisation:
Robotics and Signal Processing

Spring/ ~~Autumn~~ semester, 20.20.

Open / ~~Confidential~~

Author:
Ove Nicolai Dalheim

Programme coordinator:
Professor Kjersti Engan

Supervisor(s):

Professor Kjersti Engan, research fellow Rune Wetteland

Title of master's thesis:
Semi-Supervised Image Segmentation of Medical Data

Credits:
30

Keywords:
Bladder cancer, deep learning, semi-supervised learning, convolutional neural networks, tissue segmentation

Number of pages: 91

+ supplemental material/other: 2 + 7z-file

Stavanger, 30.06.2020
date/year

University of Stavanger

Faculty of Science and Technology
Department of Electrical Engineering and Computer Science

Semi-Supervised Image Segmentation of Medical Data

Master's Thesis in Robotics and Signal Processing
by

Ove Nicolai Dalheim

Internal Supervisors

Kjersti Engan, Professor

Rune Wetteland, PhD candidate

June 30, 2020

Abstract

Bladder cancer is the fourth most common cancer type in Norway, and tenth most common on a global scale. More and more tissue samples are sent to pathologists labs, increasing the workload and affecting the waiting time for patients. The corresponding increase is not seen in number of pathologists.

Digitization and scanning of the tissue samples unveil the world of computational pathology, along with the many possibilities within it. Supervised approaches have been proposed within deep learning earlier, however, many express the lack of labeled data as a source for performance degradation. Convolutional neural networks have proven effective on image processing within the field of medicine. Some work based on semi-supervised approaches within computational pathology has been published in the recent years, however, most researchers are exploring supervised methods.

This thesis proposes several methods to tile-wise segment histological images of bladder cancer into six different classes background, blood, damaged tissue, muscle tissue, stroma tissue and urothelium. A multiscale model is fed the tile at three different levels of magnification, and inherits capabilities from the VGG16 network through transfer learning. The proposed methods are either based on semi-supervised learning utilizing probability or clustering within a self-training process, or based on a combination of both expert and non-expert annotations.

The dataset used in this thesis consists of about 360 whole-slide images, of which only 37 contain regions annotated by a pathologist, allowing for only about 10 % of the dataset to be utilized for supervised methods. Through the course of this thesis, a total of 145 new whole-slide images were introduced during training through semi-supervised learning, utilizing about 50 % of the dataset through various methods. The non-expert approach increased the F_1-score for the muscle tissue class with 9.18 % from an initial 79.42 %, and the cluster-baser approach saw an increase of 1.38 % in accuracy from the initial 94.61 %.

The method involving non-expert annotations outperformed both semi-supervised techniques with regards to segmentation, as a semi-supervised method will introduce new uncertain features to a deep learning model. This will have a strong impact on sensitive classes that are poorly represented in the training dataset. That said, a significant improvement is seen by using semi-supervised techniques as well, and scored better than non-expert with regards to classification.

Acknowledgements

I would like to thank professor Kjersti Engan for good ideas and strong guidance, providing both opportunity and motivation for the thesis.

I am grateful for good advice and assistance from research fellow Rune Wetteland. Without his insights on machine learning this thesis would not have been possible.

A strong gratitude goes out to professor II Emiel A.M. Janssen for a good presentation of the transformation from tissue sample to a whole-slide image at Stavanger University Hospital, and for assisting in a better understanding of bladder cancer.

Also, I am grateful to senior consultant pathologist Vebjørn Kvikstad for providing the ground truth annotations in the WSIs that made this thesis possible, as well as constructive feedback.

Theodor Ivesdal deserves a recognition for all his assistance with resources on the GPU lab.

Finally, I would like to thank my family and friends for all their support.

Contents

Abbreviations

AI	*Artificial Intelligence*
CBST	*Cluster-based Self-training*
CNN	*Convolutional Neural Network*
NN	*Neural Network*
PBST	*Probability-based Self-training*
ReLU	*Rectified Linear Unit*
SGD	*Stochastic Gradient Descent*
SL	*Supervised Learning*
SSL	*Semi-supervised Learning*
TCC	*Transitional Cell Carcinoma*
TURBT	*Transurethral Resection of Bladder Tumor*
WSI	*Whole-Slide Imaging*

Frequent table abbreviations:

Ba	*Background tiles*
Bl	*Blood tiles*
Da	*Damaged tissue tiles*
Mu	*Muscle tissue tiles*
St	*Stroma tissue tiles*
Ur	*Urothelium tiles*

Introduction

This chapter presents the structure and purpose of the thesis, and motivates the work done in it. Thesis objectives and structure are also presented.

1.1 Motivation

In Norway, 1 748 patients were diagnosed, and 319 people died from bladder cancer in 2018. The majority of these, at 73 %, were men while the remaining 27 % were women [1]. Since 2001, all available statistics from the Norwegian Institute of Population-based Cancer Research places bladder cancer (including the urinary tract) as the fourth most common cancer diagnosis for men in Norway [2][3][4][5].

On a global scale, 549 393 new patients of both sexes got diagnosed with bladder cancer, while 199 922 people of both sexes died from it in 2018 [6]. This places bladder cancer as the 10th most common cancer type in the world. Bladder cancer is also known as one of the most recurring cancer types, with an average recurrence rate of 36 % for all patients [7]. Another study sets the probability of recurrence after 1 year at 61 % for high risk patients [8].

The tumor is removed from the patient through a transurethral resection, and is examined to evaluate the staging and grading of the bladder cancer. Samples contain cell-level tissue information, and a proper evaluation of the whole sample is a time-consuming process. There is an increase in number of tissue samples sent to pathologist labs for examination, however, without the same increase seen in number of pathologists, affecting the waiting time for patients [9].

Another aspect that traditional pathology faces, is that different pathologists' specific subjective expectations and experience in relation to the same tissue sample may differ [10]. Hence, different pathologist may differ in what they decide is a certain grade and stage of cancer for a given sample.

In modern times, digitalization of the tissue samples results in whole slide images (WSI), which uncovers the field of computational pathology. With computational pathology, numerous methods within image processing can be applied to ease the workload for pathologists. A viable segmentation method can assist pathologists in faster evaluation speeds, as regions of interest can be located faster. In addition, the system could contribute in a computer-aided diagnosis system, which can improve the rate of grading and staging of cancer, and also result in a more unison and objective diagnosis.

1.2 Problem Definition

Deep neural networks require vast amounts of data to become a reliable technique, to such a degree that people are claiming data more valuable than oil [11]. In many cases, the amount of labeled data is not sufficient and other means must be considered, like augmentation, manual labeling, unsupervised learning and more. For histological images, cell and tissue features will differ from cell to cell in a patient, and also from patient to patient. The available data material consists of about 360 WSIs from individual patients, of which only 37 contain annotations indicating tissue type, originating from a pathologist. This leads to a deficit for a neural network (NN) trained on the 37 WSIs, as it has only been trained on about 10 % of the entire dataset. On top of that, far from all the tissue in the 39 WSIs are annotated. This thesis proposes different methods to increase the performance of a NN, based on utilization of unlabeled data from the dataset, in order for the end product to be more consistent, and more robust against misclassifications. The process of utilizing both labeled and unlabeled data in training a NN is known as semi-supervised learning, further explained in Section 3.5.

1.3 Previous Work

During the past 30 years or so, NNs have had a major advance in image and signal processing in general [12]. In the most recent decade, convolutional neural networks (CNN), especially, has proven very powerful when applied to medical tasks in image processing and classification [13][14], also gaining popularity in computational pathology. The most common way to train neural networks (NN) is by supervised learning and backpropagation, requiring a large training dataset of associated relevant ground truth labels. Ground truth labeled samples within medicine is in many cases limited, as producing it is a time-consuming process. Moreover, for the samples to be labeled correctly it must be done by a capable expert with knowledge on that specific type of sample.

Instead of producing ground truth labels, methods exists that allows for CNNs to be trained on the data, by implementing techniques like clustering or unsupervised learning. One method is to use autoencoders in a compression-decompression setup. The network tries to reconstruct the original input, and features are learned at its most compressed state. The decompression part can then be replaced by a small classifier network, associating features to classes in an output layer [15]. The drawback here is that autoencoders, and unsupervised learning in general, will typically not perform as well as models trained with ground truth labels in a supervised manner.

One of the main benefits with CNNs is that a particular feature can be detected wherever it may be located in the image, deeming these types of networks shift-invariant. Intuitively, the initial layers in a CNN can be viewed as raw feature extraction layers, while the last layers can be thought of as more task-specific object detection or classification layers. The network contains many parameters that can be adjusted to improve its performance, which can be a time-consuming process. In addition, large amounts of data is required for the network to fully grasp the complete set of features associated with each class. A method much used to facilitate the initial layers is transfer learning, where the first layers are inherited from a pre-trained network, and the last layers are trained from scratch [16].

Incorporating the above approaches in deep learning, uncovers a method known as semi-supervised learning (SSL). SSL utilizes both labeled and unlabeled data to train a network, and proves very capable in cases where there are small amounts of labeled data, but large quantities of unlabeled data. Graph-based learning is a branch within SSL that often implement clustering algorithms to locate and distinguish inputs in feature space [17]. Self-training is another branch within SSL, and aims to initially train a model on ground truth labels in a supervised manner. Thereafter, weak labels are produced from new unlabeled data, using predictions from the initial model. Finally, a new model is trained on both the ground truth labels and the weak labels [18].

In Skrede et al. [19], a deep learning method for prediction of colorectal cancer outcome was proposed. The tumors are removed, and further imaged through a process similar to that described in Section 2.4. The proposed method involves ten individual CNNs, where half are fed images at 10x resolution and the other half are fed 40x resolution images. The output of the 10 models are then evaluated to decide on either a good or bad prognosis. The method utilizes tile-wise binary classification, and is trained on over 12 million tiles. The method focuses on assisting pathologists in diagnosis, and claims an increase in precision of a diagnosis by 62 % [20].

In McKinney et al. [21], a CNN model for detecting breast cancer in X-ray images is proposed. The model is based on TensorFlow, and features three CNNs in parallel that each are fed the X-ray image at different levels of magnification. The outputs of each CNN is compared, and the network was trained on X-ray images from almost 29 000 patients. The system was proven to outperform the normal cancer prediction, in which normally two radiologists works together on diagnosing the patient based on the X-ray images. The model reduces false positives with 5.7 % and false negatives with 9.4 % in data from the USA.

In Cheplygina et al. [18], a survey on approaches in SSL in medical images is presented. It popularizes the use of both transfer learning and semi-supervised learning in the recent years, and underlines benefits of utilizing the assumption that tissue located closer to each other are more likely to be of same class. In Dercksen et al. [22], both unsupervised and semi-supervised approaches are combined and applied to computational pathology. An autoencoder is trained on unlabelled data, and k-means clustering is applied at the most compressed state, i.e. feature space. In Peikari et al. [23], a cluster-then-label approach is taken using support vector machine classifiers. An adaptive threshold is used to remove irrelevant parts of the inputs, which saves processing time. Remaining regions are then split up into tiles, and further separated based on the underlying structure.

In 2017, co-supervisor Wetteland wrote his master thesis titled *Classification of histological images of bladder cancer using deep learning* [24], applying deep learning to the same dataset as the one used in this thesis. After graduating, Wetteland has been further working on the system and dataset, employed as a research fellow at the University of Stavanger. In Wetteland et al. [25], a multiclass tissue classification model is presented, that utilize tile-wise segmentation. An autoencoder is first trained on unlabelled data, and further fine-tuned using labelled data. In Wetteland et al. [26], the latest system is presented, which utilizes three magnification levels, and incorporates transfer learning in three CNNs operating in parallel, as further elaborated in Section 4.2.

1.4 Thesis Objectives

Previous work done in relevant fields of research underline the benefits of utilizing transfer learning and semi-supervised learning when dealing with datasets with small amounts of labeled data. The work done in both McKinney et al. [21] and Skrede et al. [19] highlights the importance of including multiple magnification levels to capture both local details and surrounding context of the tissue at hand. Previous work done on the same dataset as used in this thesis also underline this [26]. Peikari et al. [23] utilize the assumption that tissue that is located closer to each other are more likely to be of same class.

The primary objective of this thesis is to investigate the use of different semi-supervised methods, to see if they are effective in improving the accuracy of the models, without the need for an expensive labeling process.

1.5 Thesis Structure

A layout of the thesis structure is given below.

- Chapter 1: Introduction
  - Motivation and previous work, followed by thesis objective
- Chapter 2: Medical Background Theory
  - Background material necessary for understanding the rest of thesis from a biological point of view
- Chapter 3: Technical Background Theory
  - Background material necessary for understanding the rest of thesis with respect to NN, and the many methods involved
- Chapter 4: Material and Previous work
  - The work of co-supervisor R. Wetteland is presented in brief along with the dataset
- Chapter 5: Methods
  - Methods proposed to solve thesis objective
- Chapter 6 Experiments and Results
  - Experiments done to achieve thesis objective, and their respective results are presented
- Chapter 7 Discussions and Conclusion
  - Discussion on performance of proposed methods, future work, failed attempts etc, and conclusion on best method.

Medical Background Theory

This chapter introduces the fundamental medical knowledge needed to have an understanding of the dataset. Different types of tissue commonly found in the WSIs are present first, followed by anatomy of the bladder, and details on bladder cancer.

2.1 Overview of Tissue Types

Tissue consists of biological cells in a structure that performs a specific function, and incorporates many important tasks involved with maintaining our body [27]. The human body contains four different types of animal tissue. There is connective tissue providing support, epithelial tissue covering and protecting the body, muscular tissue providing movement, and nervous tissue maintaining control and communication. Organs are then built up of a combination of these different tissues types. The most relevant tissue types are connective, epithelial and muscular tissue, which will be further discussed below.

2.1.1 Epithelial Tissue

Epithelial tissue maintains order in the body, and acts as a protective layer for other types of tissue. Epithelial tissue forms layers around the exterior of organs in the body, and also joins connective tissue in generating the skin which covers the outer body. Epithelial tissue exists in three main shapes and sizes depending on their purpose [28]. As shown in Figure 2.1, squamous cells are flat, elongated organisms that can form both simple and stratified structures, making them act as thin membranes. Cuboidal cells are typically in the shape of a cube with equal sides, whereas columnar cells are formed more like a column.

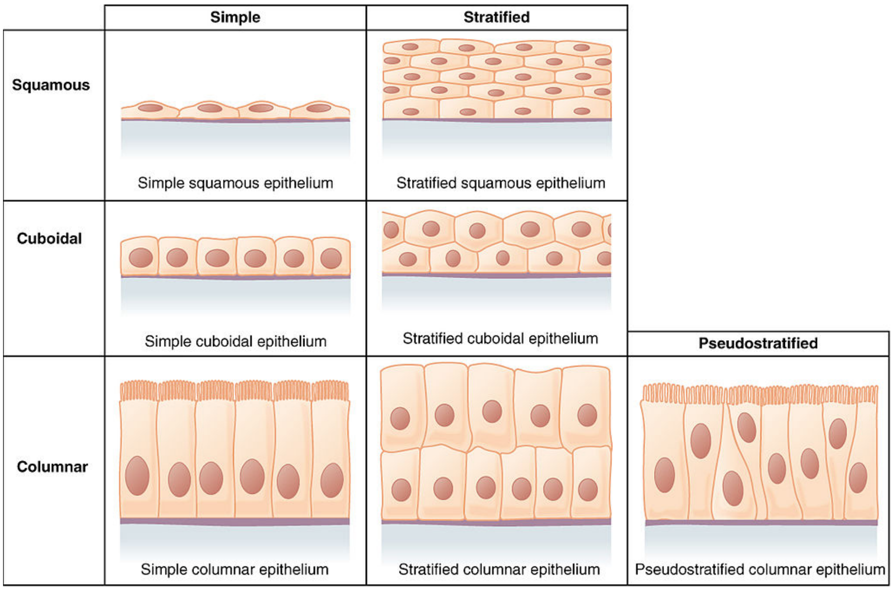

Simple
Stratified
Squamous
Simple squamous epithelium
Stratified squamous epithelium
Cuboidal
Simple cuboidal epithelium
Stratified cuboidal epithelium
Pseudostratified
Columnar
Simple columnar epithelium
Stratified columnar epithelium
Pseudostratified columnar epithelium

Figure 2.1: Epithelial tissue shapes and structures.
This figure is reprinted in unaltered form from Wikimedia Commons, *File:403 Epithelial Tissue.jpg*, licensed under CC BY 3.0 [29]

The cells are polar, where one side connects to the underlying basement membrane, and the other side is exposed to the outside. The layer of epithelial tissue often acts as a selectively permeable, allowing for certain molecules to pass through it. An examples of this type of structure can be seen in Figure 2.2, where an outer purple layer of epithelial cells called urothelium are arranged together to form the mucosa membrane of the inner wall of the bladder, protecting the inner layers of stroma tissue.

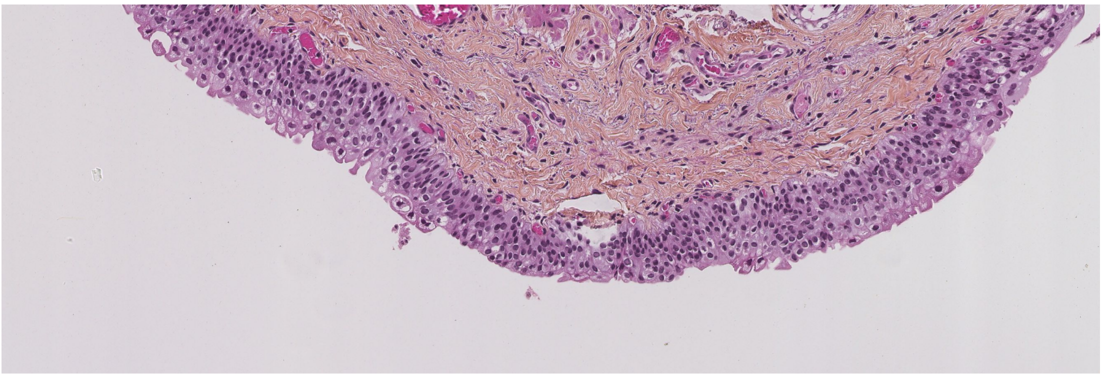


Figure 2.2: Transitional urothelium tissue (purple outline) acting as a mucosa membrane. Extracted at 100x Magnification.

2.1.2 Connective Tissue

Connective tissue is found in most parts of the body, often located between other types of tissue. Connective tissue connects other types of tissue, and acts as the glue keeping the body together [30]. It is also responsible for supplying the body with oxygen and nutrients through the cardiovascular system. All connective tissue have some form of vascularity, however, the most relevant parts with respect to this thesis is stroma tissue and blood cells. Blood cells are uniquely identified by their distinctive color and texture as seen in Figure 2.3. Stroma is a more general term for tissue, consisting of all the tissue that does not have a specific related function in an organ, see Figure 2.4. Stroma may consist of blood, nerves, fat, and other types of connective tissue.

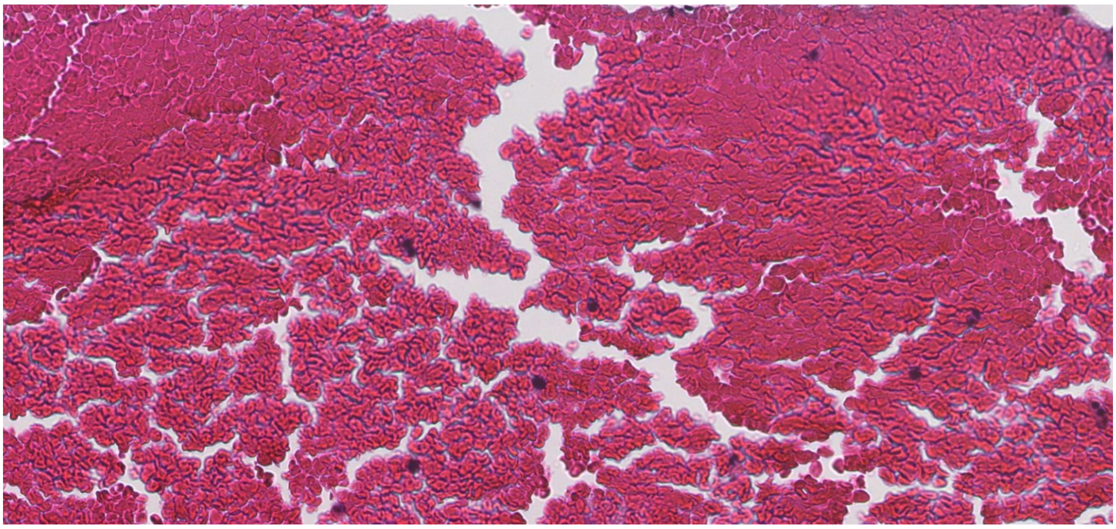


Figure 2.3: Example of blood cells, extracted at 400x level.

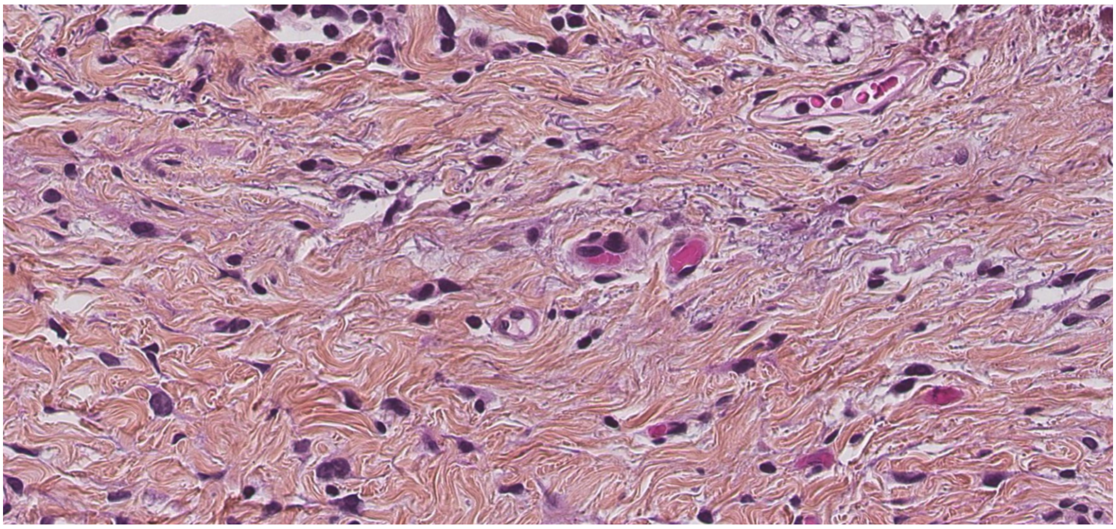


Figure 2.4: Example of stroma tissue, extracted at 400x level.

2.1.3 Muscle Tissue

Muscle tissue is often connected to the skeleton, and provides movement in our body. The constant contracting and relaxing of the muscle is caused by the proteins actin and myosin, turning chemical energy into mechanical energy. As seen in Figure 2.5, muscle fibres appear as long and thin lines of a dark pink color.

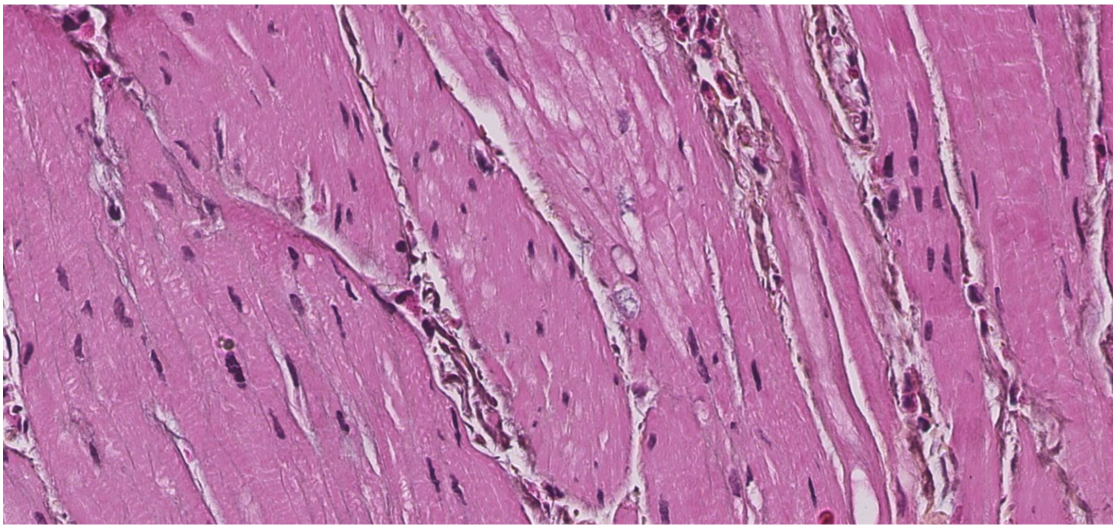


Figure 2.5: Example of muscle tissue, extracted at 400x level.

There are three types of muscular cells in our bodies, namely cardiac, smooth, and skeletal. Cardiac muscle tissue is as the name suggest located in the walls of the heart, and are not relevant to this thesis. Skeletal muscle cells are located between joints and connects to our skeleton, and are also not relevant to this thesis as they appear very directional in the sense that their individual purpose is linear movement of a joint. Smooth muscle cells are under involuntary control, and are located in the walls of hollow organs, such as the bladder. The purpose of smooth muscle tissue is not necessarily a fast linear movement like skeleton muscle tissue, but for instance a slow compression of the bladder [31]. As such, the muscular tissue observed in this thesis may be more or less mixed together in all different directions. That said, on a microscopic level it may still appear quite linear.

Specific muscles in the body often have unique names, like the myometrium specifying the muscle in the uterine wall responsible for uterine contractions. Nevertheless, in this thesis "muscle tissue" will be used to describe muscles that originates in the inner wall of the bladder as further explained in Section 2.2.

2.2 Urinary Bladder

The kidneys filter the blood from metabolic waste products. On average, approximately 180 liters of blood passes through the kidneys per day, however, the average urine output is only about 1.5 liters daily [32]. The urine makes its way to the bladder through the ureters, which connects the bladder and the kidneys. The ureters enters the kidneys at an angle to prevent back-flow when the bladder is full. The ureter also consists of a muscular lining, which helps to pass urine along. The bladder stores the urine until an appropriate time to urinate. An illustration of the bladder is given in Figure 2.6.

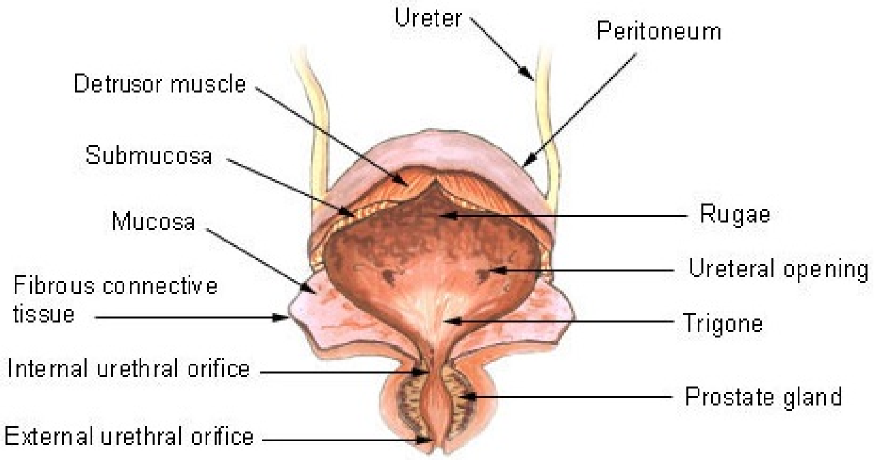


Figure 2.6: Anatomy of the urinary bladder.
This is an altered figure from Wikimedia Commons,*File:Illu bladder hr.JPG*, licensed under public domain [33].

Biologically, the inner-most layer is epithelial tissue, termed urothelium in this thesis, and is a mucosa membrane which protects the body from the urine inside the bladder. Below the mucosa membrane is a basement membrane on top of a layer of connective tissue in the form of stroma, which is called lamina propria and is technically also a part of the mucosa. The bladder has a muscular wall called detrusor muscle that connects to the lamina propria, and contracts when the bladder is emptied, and also expands when it is filled [34].

2.3 Bladder Cancer

The most common form of bladder cancer is urothelial carcinoma, also called Transitional Cell Carcinoma (TCC). It is most common among the elderly, and is strongly associated with smoking and tobacco usage [35]. The name transitional cell carcinoma originates in that the transitional epithelial tissue will evolve and mutate, generating abnormal cells which in turn can progress into carcinoma. TCC is characterized by abnormal tissue in the bladder, and often causes symptoms such as hematuria, i.e. blood in urine. Bladder cancer is the 10th most common cancer type in the world, being about three times more common among men than women [36]. Recurrence is also a very serious aspect of TCC, with an average recurrence rate of 36 % [7] for bladder cancer globally. On a global level, 549 393 people were diagnosed with bladder cancer in 2018, and 199 922 people died from it [6].

The Tumor Node Metastasis (TNM) Classification of Malignant Tumors has defined different stages of cancer depending on its spread. In its earliest stage, the carcinoma is only confined to the urothelium layer, which is called Carcinoma In Situ (*CIS*). As Figure 2.7 illustrates, the tumor will typically grow inwards into the hollow bladder for *Ta* stage. *T1* indicates spread through the basement membrane, and into the stroma. Stages *T2a* and *T2b* indicate that carcinoma has grown into the inner and outer detrusor muscle respectively. If the TCC reaches the outer layer of fat it is graded stage *T3*, and if it has reach tissue of adjacent organs it is graded stage *T4* [37]. In cases where the carcinoma grows into the muscle tissue of the bladder wall, the prognosis is worse and a cystectomy may be necessary, i.e. removal of the bladder. The cystectomy is either partial or radical, where the bladder is partially or fully removed respectively, depending on the spread [38]. If the cancer spreads to nearby lymph nodes it is staged from *N0* to *N3*, and if it spreads to other parts of the body it is staged as *M1*.

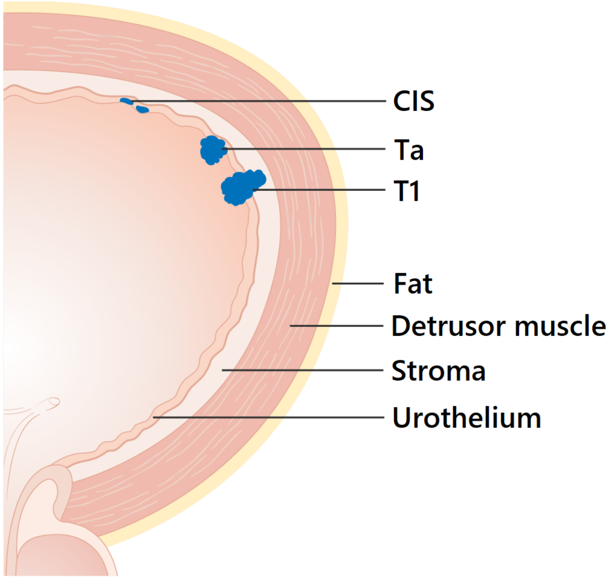


Figure 2.7: Early stages of bladder cancer.
This is an altered figure from Wikimedia Commons, *File:Diagram showing early stage bladder cancer CRUK 442.svg*, licensed under CC BY-SA 4.0 [39]

Cancer grade is also used to diagnose bladder cancer with the use of Worlds Health Organization's (WHO) grading systems, WHO1973 and WHO2004. WHO2004 grades papillary urothelial neoplasm of low malignant potential (PUNLMP) as low or high grade

[40]. WHO1973 grades the tumors from 1 to 3 [41]. The grading is based on the tissue texture, arrangement of the cell nuclei, abnormal appearance of cell nuclei and rapid increase in number of cell nuclei.

In 2017, new guidelines were published by European Association of Urology under the acronym WHO2016. The new guidelines aims to improve clinical management of non-muscle-invasive bladder cancer (NMIBC), i.e. the earliest stages of bladder cancer. WHO2016 offers recommendations graded A, B or C with regards to cystoscopy and other clinical measures for NMIBC. According to the guidelines, it is recommended to use TNM staging for classification of tumor invasion, and a combination of WHO1973 and WHO2004/WHO2016 is recommended for grading [42].

The goal of the diagnosis is to obtain a correct grading and staging of the cancer, as well as the chance of recurrence. This will further be conclusive for a cancer-treatment plan for the patient. Some of the features pathologists look for is abnormal growth in the urothelium tissue. A healthy urothelium region will typically have cells that are aligned and evenly spaced, and are structured in an organized manner. Tumorous urothelium tissue can appear more chaotic and unorganized. Regions that contain muscle tissue are especially interesting, as this can indicate how far the tumor has infiltrated the bladder wall. The tissue regions that contain damaged tissue and blood will not be evaluated from a diagnostic perspective [43].

2.4 Histology and Immunohistochemistry

Histology is the study of cells and tissue from plants and animals, and is a branch of biology. Histology as a field of study has been around since the 17th century when Italian Marcello Malpighi studied different body parts from animals under a microscope [44]. In modern histology, more advanced optical lenses are used, and images are often digital. The use of histology has become an important part of diagnostic procedures within many fields of medicine. Following a tour at the Department of Pathology at the Stavanger University Hospital, where the immunohistochemistry procedure of tissue from bladder cancer patients was introduced, the following was observed in brief:

The tissue is removed from the patient through Transurethral Resection of Bladder Tumor (TURBT) by the use of a resectoscope. This is a long tube-like tool that is probed from the urethra up into the bladder, with a tiny camera mounted at the end. It also holds a tool to remove tissue from the inside of the bladder, like a laser or a heated wire loop. The resulting tissue will often bear mark of edges torn apart or burned off. After the tissue is removed, it is fixed in formalin before being embedded into paraffin. When the paraffin has solidified, it has a similar consistency to tissue, and can more easily be sliced into 4 μm thick slices. This is achieved with a Leica RM2255 microtome, a slicing tool that firmly moves a knife up and down as it cuts into the paraffin. Variation in slice thickness is relatively common, and sources problems like color variation in the resulting image due to different levels of light passing through. The slices are then mounted on a glass slide and stained with Hematoxylin, Eosin and Saffron (HES) by the automatic staining machine, Ventana HE 600. The resulting image will vary depending on staining effectiveness and tissue quality, which is further discussed in Section 3.1.2.

Hematoxylin and Eosin is a widely used staining combination with the two dyes pink and purple. Hematoxylin binds to DNA, which is located in the cell nucleus, coloring them purple. Eosin binds to positively charged compounds like proteins and cytoplasm, and will color them pink or red [45]. Saffron is added to distinguish stroma tissue from muscle tissue, i.e. coloring stroma tissue yellow/orange while muscle tissue remains pink from the Eosin [46].

Technical Background Theory

This chapter takes a technical look at the histological images, and offers a brief history of neural networks. Neural networks are explained, and different techniques used in conjunction with them.

3.1 Histological Images

Histological images are digital images formed by scanning a histological slide, i.e. a tissue sample. An example of a histological image is shown in Figure 3.1.

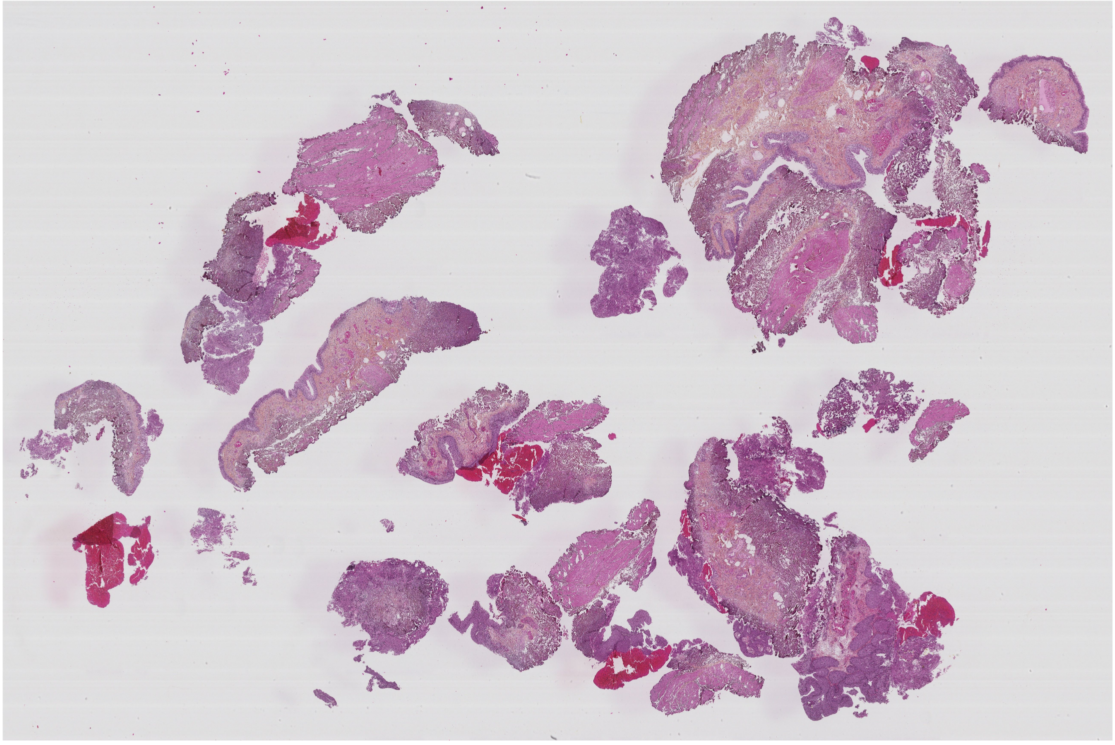


Figure 3.1: An example of a histological image in the dataset, extracted at 25x magnification.

3.1.1 Tissue Classes

WSI is also known as virtual microscopy, and uses an array of lenses to capture smaller images at high resolution, which in turn can be compiled together to form an image of the whole slide. There are mainly six types of tissue or cells found in the WSI used in this thesis. These make up the classes explained in Table 3.1, and also illustrated in Figure 3.2.

Table 3.1: Description of tissue classes used in this thesis.

Class	Description
Background	Consists of white or light gray colored pixels, or small parts of debris, ink spots or other irrelevant features
Blood	Red blood cells, distinctively red in color and texture-wise high quantity of small cells
Damaged	Any type of cell or tissue that has been damaged due to burn from biopsy or otherwise torn apart
Muscle	Smooth muscle fibers, distinctively pink in color and texture-wise elongated cells
Stroma	Group of tissue types that involve blood, nerves, fat and other types of connective tissue
Urothelium	Urothelium tissue from the mucosa membrane in the bladder

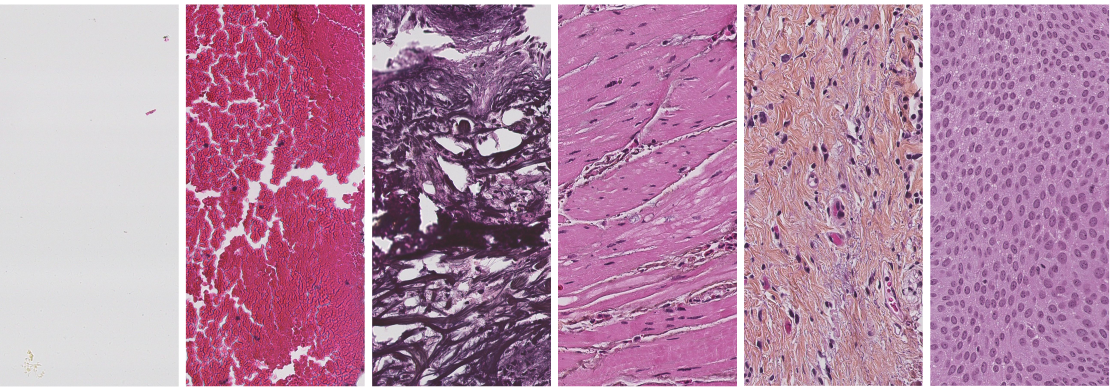


Figure 3.2: The different classes used in the deep learning model, extracted at 400x magnification. From left to right: Background, Blood, Damaged, Muscle, Stroma and Urothelium.

The classes blood through urothelium are considered the five foreground classes.

3.1.2 Image Quality

The scanning procedure of the SCN400 captures the sample at 400x magnification, which enables pathologists to study cells up close. On the contrary, this leads to some variations in image quality. When the lens is focused, small variations in distance from the cell to the focused area can cause areas of tissue that are out of focus, as can be seen in Figure 3.3. This reduces the detail quality in the region, and can in some cases make for a bigger challenge with regards to classification.

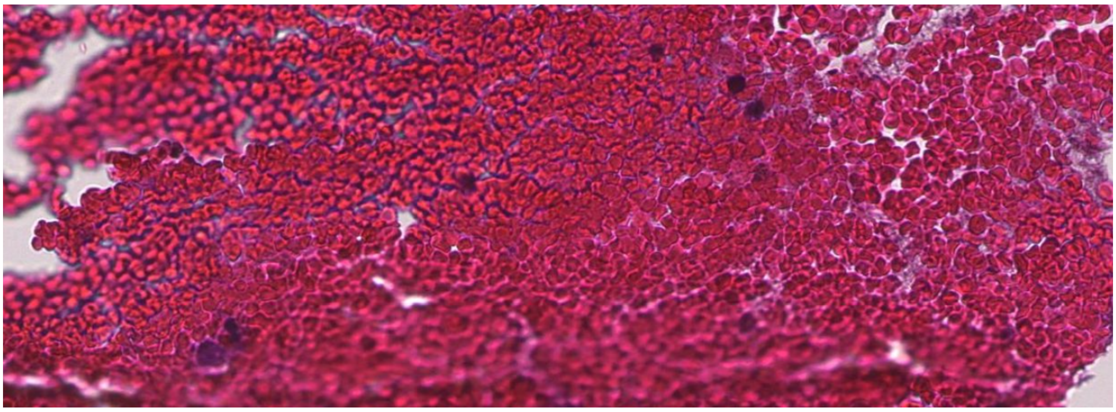
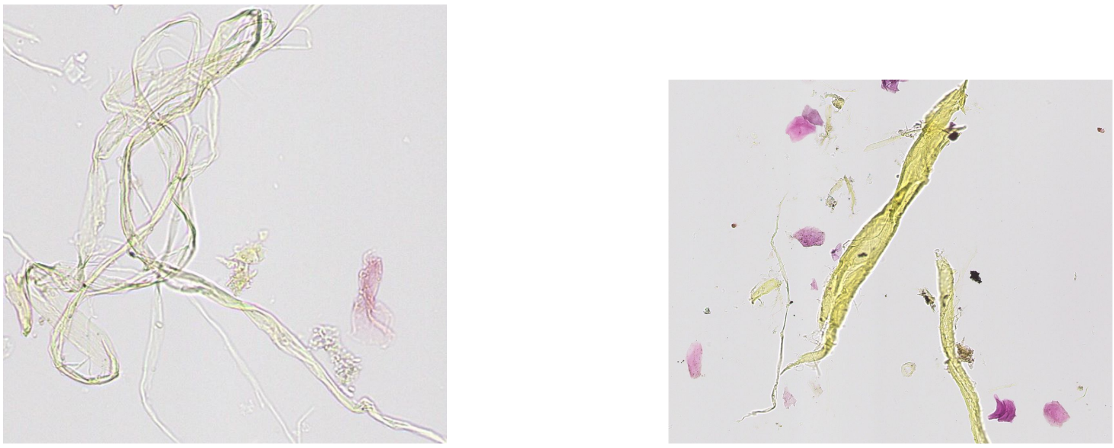


Figure 3.3: Example of an area that appears to be out of focus. Extracted at 400x magnification.

The tissue samples are prepared in a sterile zone at the hospital, however, debris can still be found on many of the images. The debris can originate as damaged tissue torn from the main sample, as this can naturally occur when it is prepared. Debris may also come from dust particles etc. that somehow makes its way in front of the lens, or onto the sample. Figure 3.4 shows some examples of debris found in the dataset.

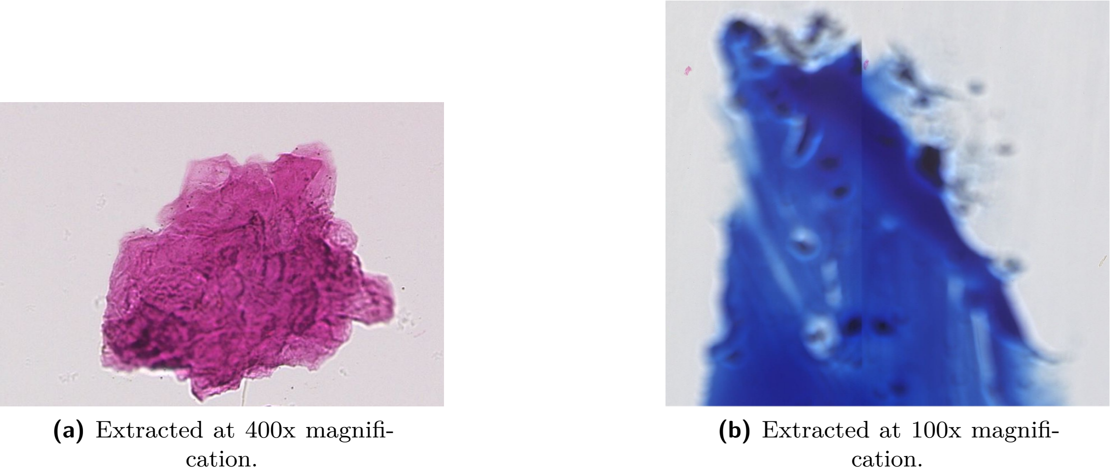


(a) Thin debris.

(b) Debris with some ink stains.

Figure 3.4: Example of debris found in the images from the dataset. Extracted at 400x magnification.

Certain WSIs contain areas that appear to be pure ink stains from either the staining procedure, or from colored markers used by pathologists to indicate regions of interest early on. Different coloured lines or spots can be seen in areas where no tissue appears to be present. Figure 3.5 shows some examples of this.



(a) Extracted at 400x magnification.

(b) Extracted at 100x magnification.

Figure 3.5: Example of ink spots found in the images of the dataset.

In an effort to make the images contain as few amount of cells inwards in the picture, the slides are cut at a very small thickness of 4 μm. A direct consequence of this can be seen in several of the images, as the tissue appear to be folding. The degree of folding will vary from a single short fold, to entire areas crumbled together, and appears to particularly affect larger clusters of red blood cells. An example of this can be seen in Figure 3.6.

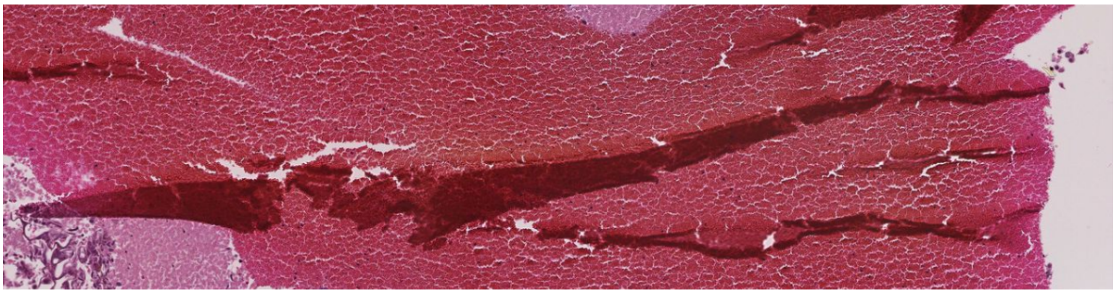


Figure 3.6: Example of an area that appears to be folded together. Extracted at 100x magnification.

Some images appear to have a shadow print right next to it. This can be observed in some areas in Figure 3.1, appearing as a shadow to the left of the original shape. This does in most cases not cause any issues as the shadows are of low contrast.

As mention previously, variation in slicing thickness and variation in staining effectiveness can both cause color variation. This can be seen in Figure 3.7, however, its affect can be observed on other tissue types as well. Consequently, this can lead to the region having different features in the different color channels, which are detected by individual kernels, see Section 3.3.2. Others have tried to normalize the stain colors in histological images [47], however, this has not been a subject in this thesis.

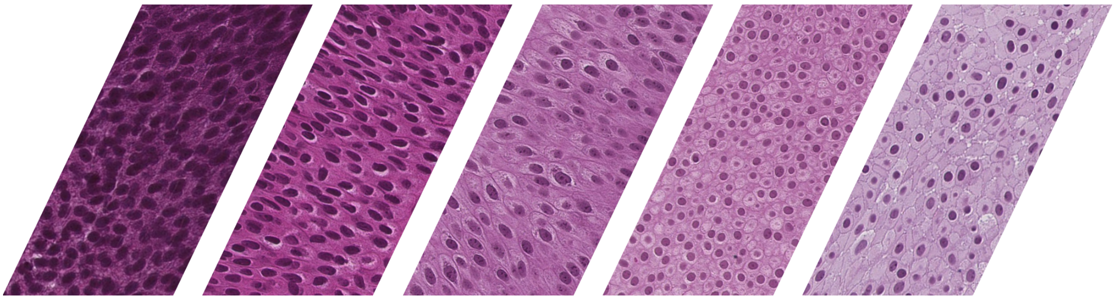


Figure 3.7: Example of color variation in urothelium tissue, from different images in the dataset. Extracted at 400x magnification.

3.1.3 SCN-Format

The images used in this thesis are of a .scn-file extension, which is Leica's own format used to view the images in their own software, Aperio ImageScope. The files are very large, with some files accounting for several gigabytes of storage. The format has a pyramidal structure, to accommodate for rapid zooming in and out of areas when the pathologists are examining a WSI. This structure also allows for easy extraction of images at different magnifications, when the images are so large. SCN-files are based on BigTIFF, which is the equivalent to TIFF-format with a larger offset to allow for larger images to be saved in the same file [24].

Other applications can open the file extension as well, like PyVips [48] which is used in this thesis. The open-source library was developed by J. Cupitt and K. Martinez, and can load specific parts of the image into memory, as the image is too large to load as a whole.

3.2 Neural Networks

This section starts with a brief history on neural networks, and gives an introduction to how they work. Later, more advanced deep learning architectures are presented.

3.2.1 Origin of Neural Networks

The history of neural networks dates back to the Second World War, when in 1943 Warren Sturgis McCulloch attempted to model a simple NN with electrical circuitry, proposing that neurons were the base logic unit of the brain [49]. In 1949, Donald Hebb proposed that our memory was based on interactions between neurons structured in "cell-assemblies" in our brain, and that a particular path in an assembly got stronger each time it was activated [50]. During the 1950s, IBM researched on simulating neural networks utilized within the fields of pattern recognition and information theory. In 1957, Frank Rosenblatt invented The Perceptron, a binary NN with adjustable weights of analogue architecture in-between them. Rosenblatts network was able to classify shapes and letters in images, and was even able to distinguish between some photos of men and women based on gender [51]. The Perceptron was later proven to be very limited in 1969, which to a significant degree halted further development on NN.

In 1970, Seppo Linnainmaa published a paper on automatic differentiation of discrete connected networks, a technique in backpropagation of errors in multilayer NN, still used to this day. The paper did, however, not refer to neural networks as a use case of the method [52]. In 1986, a paper describing backpropagation and its use in NN was published by D. E. Rumelhart, G. E. Hinton and E. J. Williams, which sourced a new interest for NN as it theoretically allowed for approximation of any function. Nevertheless, with the relatively poor computing power at the time, most researches slowly began to work on other techniques [53]. In 1992, max-pooling was first introduced in conjunction with NN [54], and in 2012 Andrew Yan-Tak Ng and Jefferey A. Dean presented a network that could classify cats from unlabeled data [55]. In the years to follow, neural networks trained on Graphical Processing Units (GPU) allowed for larger networks that could process larger inputs like images and video [56].

Today, many of the biggest technology companies out there are utilizing neural networks in their operations and products [57]. Tesla's vehicles implement a driver-assistance system known as Autopilot, which assist the driver in things from lane-centering to automatic emergency breaking in case of danger. The system feeds data back to Tesla for them to further train their Autopilot on [58]. Google utilizes NN in their speech recognition system, photo search and many other platforms, and have developed their own open-source library that can be used for NN among other machine learning architectures, called TensorFlow [59].

3.2.2 Artificial Neurons

Artificial neurons are the elementary units in artificial neural networks, and is a mathematical representation of biological neurons in the brain. A single artificial neuron is shown in Figure 3.8. The orange circles represent the inputs, x_i, to the artificial neuron in green.

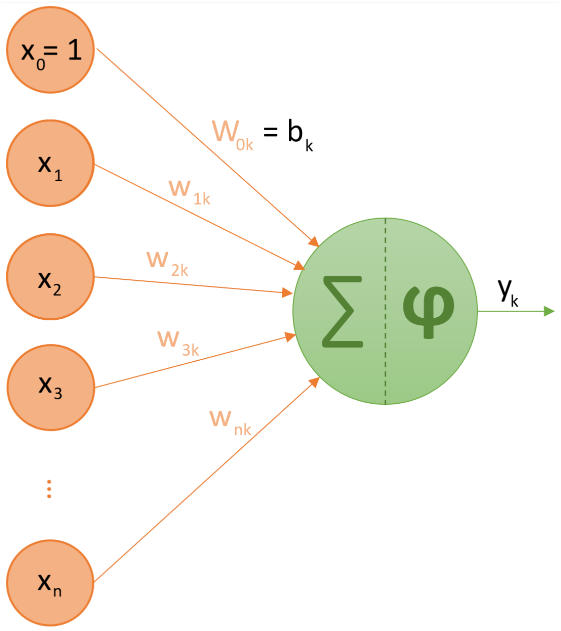


Figure 3.8: Illustration of a neuron.

In neural networks, neurons are often referred to as nodes, and a network can consists of several thousands of them. As such, it is a good practise to assign a number to each node. Node k is composed of two components, a summation part, Σ, and a function, φ, referred to as the nodes activation function or transfer function. The lines connecting the inputs to the node are called weights, so that a particular input can have different affects on the different nodes it is connected to. How much the input, x_i, affects node k is given by the weight w_{ik}, and so forth. This gives us the output of nodes k as y_k in Equation 3.1, with n inputs.

$$y_k = \varphi\left[\sum_{i=0}^{n} x_i w_{ik}\right] = \varphi[x_0 w_{0k} + x_1 w_{1k} + x_2 w_{2k} + ... + x_n w_{nk}] \tag{3.1}$$

Additionally, nodes will normally have a certain bias to allow for a linear shift of its output [60]. This bias is given as $x_0 w_{0k}$, however, since $x_0 = 1 \; \forall k$, the bias of node k is really just w_{0k}, and often simply denoted as b_k. This results in the output given in Equation 3.2.

$$y_k = \varphi[x_0 w_{0k} + x_1 w_{1k} + ... + x_n w_{nk}] = \varphi[b_k + x_1 w_{1k} + ... + x_n w_{nk}] \tag{3.2}$$

For networks involving multiple nodes, the bias is often not included in the overall drawing of the network.

3.2.3 Fully-connected Neural Networks

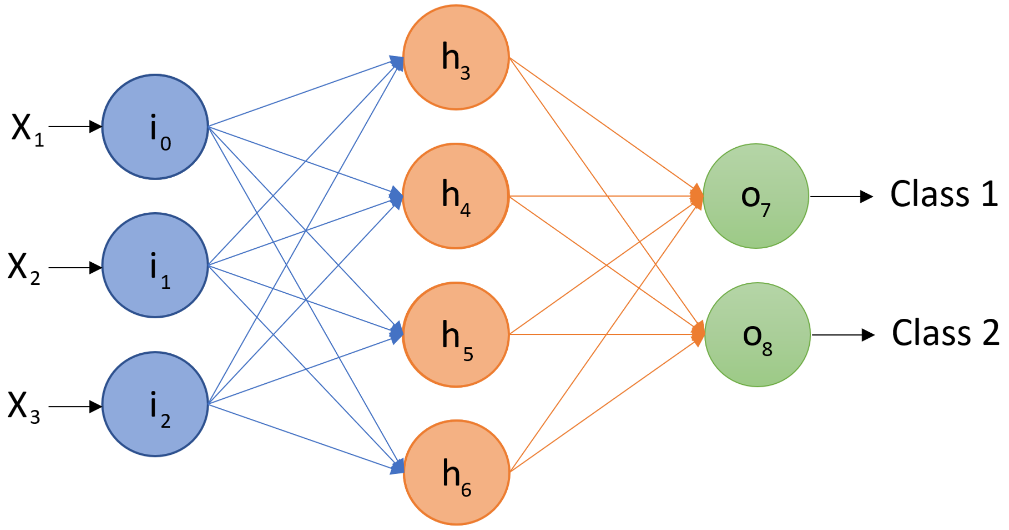


Figure 3.9: Illustration of a simple feed forward fully-connected neural network.

In Figure 3.9, a simple configuration of a fully-connected neural network (FCNN) is illustrated, where all the nodes are connected. Each circle represents a node, which has a number of inputs and outputs. The first layer of nodes is called the input layer, which in Figure 3.9 is made up of nodes i_0, i_1, i_2, and has to match the dimensions of the dataset. The last layer of nodes is called the output layer, nodes o_7, o_8 in Figure 3.9, and must be equal to the number of classes that the FCNN should classify. In between them there may exists several layers of neurons, called hidden layers, which in Figure 3.9 is made up of neurons h_3, h_4, h_5, h_6.

Ultimately, the goal of the network is that when the network is fed new unknown data to the input layer, the correct corresponding output nodes should be activated. In order to achieve this, the network has to be trained so that it learns to distinguish between different inputs, by adjusting the weights and biases in the network. There exists several different methods of learning, as abbreviated in Section 3.5, however, supervised learning will be explained in detail here.

For supervised learning, the network is trained on labeled data, i.e. data that has been assigned a label indicating the correct class, often represented in the form of a vector. More in depth, the weights and biases in the network must be adjusted in such a way that the network learn which nodes that should activate in the output layer. The dataset one wish to train the network on is split into a test dataset and a training dataset, typically training on a significantly larger portion of the dataset then what it is tested on. After the network has been trained, it can be tested on new data to see if the network classify it correctly, i.e. activate the correct output node. Normally, a FCNN will have a far greater size and complexity than what is illustrated in Figure 3.9. Typical characteristics of the NN will be explained in this section.

3.2.4 Activation Function

The output of a node is given by its activation function, which takes parameters from all nodes located in the previous layer. The activation function of a particular node can vary depending on what layer in the network it belongs to, and specific activation functions fulfil different use cases. In some cases, one may want to limit the output of a given node. This can be done by the use of a Sigmoid function, σ, as the activation function, φ, which is shown in Equation 3.3. The Sigmoid function is often used in the output layer to limit the output between 0 and 1 [61]. Referring to Figure 3.9, the output of the output node, o_7, is given in Equation 3.3, where the nodes from the previous layer is represented as h_x. The bias of node o_7 is represented as b_7, and the weights denoted as w_{12} being the weight between node 1 and 2.

$$o_7 = \sigma\left[\left[\textstyle\sum_{i=3}^{6} h_i w_{i7}\right] + b_7\right] = \sigma(h_3w_{37} + h_4w_{47} + h_5w_{57} + h_6w_{67} + b_7) \tag{3.3}$$

$$= \tfrac{1}{1+e^{-(h_3w_{37}+h_4w_{47}+h_5w_{57}+h_6w_{67}+b_7)}}$$

The Sigmoid function will exponentially converge towards 1 for large positive input values, and converge towards 0 for large negative input values. Another well known activation function is the Rectified Linear Unit (ReLU) [62]. The ReLU function simply prevents the output from being negative, as shown in Equation 3.4.

$$o_7 = max(h_3w_{37} + h_4w_{47} + h_5w_{57} + h_6w_{67} + b_7,\ 0) \tag{3.4}$$

Using the Sigmoid function in the output layer of a multiclass NN model can result in the total sum of all the nodes in the output layer being greater than 1, which is problematic in probability theory. A function similar to Sigmoid compensates for this, and is called Softmax. The Sigmoid function limits the output of a particular node to be somewhere between 0 and 1, where as the Softmax function ensures that the entire layer sums up to 1 [63]. The Softmax activation function for the node 0_7 is given in Equation 3.5.

$$o_7 = \frac{e^{o_7}}{e^{o_7} + e^{o_8}} = \frac{e^{(h_3w_{37}+h_4w_{47}+h_5w_{57}+h_6w_{67}+b_7)}}{e^{(h_3w_{37}+h_4w_{47}+h_5w_{57}+h_6w_{67}+b_7)} + e^{(h_3w_{38}+h_4w_{48}+h_5w_{58}+h_6w_{68}+b_8)}} \tag{3.5}$$

The Softmax function is often used in the output layer of a NN, however, it allows for few flaws in the design of the classifier, as it will deliver a sum of 100 % probability every time. In other words, the output can never be 0 % for all the classes, which would be correct if new data is to be classified, where the true label of that data is neither of the classes in the output layer. If the classification problem involves labels that are mutually exclusive, i.e. one sample cannot be more than one class, Softmax must be used of the two. If the data can belong to several different classes in the output layer, Sigmoid can be used.

3.2.5 Cost Function

When a new neural network is initialized, its weights and biases are normally set to random numbers taken from a truncated Gaussian distribution. When it is fed new data during training, a method of quantifying how bad or good the network performed is needed [64]. This is normally done by computing a cost function, and there exist multiple cost functions to choose from. Two much used cost function are cross-entropy, and mean squared error (MSE).

Mean squared error

MSE computes the squared of the differences between the actual output, o_{x_a}, and the correct output, o_{x_c}. This is referred to as the loss of the respective sample, and the mean cost refers to the average of all the losses of all the samples. As an example, the network in Figure 3.9 is fed some data which is resulting in the outputs $o_7 = 0.86$ and $o_8 = 0.14$. The correct label for that particular data corresponds to the outputs $o_7 = 1.00$ and $o_8 = 0.00$, which results in the loss of this example in Equation 3.6, when using the MSE cost function.

$$C = (o_{7_a} - o_{7_c})^2 + (o_{8_a} - o_{8_c})^2 = (0.86 - 1.00)^2 + (0.14 - 0.00)^2 = 0.0392 \tag{3.6}$$

The mean cost is small when the network is close to the true values in the output layer, and grows larger the more incorrect it is.

Cross-entropy

Entropy, with respect to information theory, refers to the probability of certain events. If the probability distributions representing these events is balanced, with each event being just as likely, the events will have a high entropy. If the probability distributions is skewed, with some events being more or less likely than others, the events will have a low entropy. Entropy can be viewed as the spread of the probabilities among the possible events. In entropy coding, this is used to code transmitted data by using the least number of bits to represent the most likely events, and most number of bits to represent the rarest events.

Cross-entropy calculates the difference between the entropy in two distributions. The number of bits used to transmit the average event in one distribution compared to the average event in the other distribution. When a neural network is fed data during supervised training, the data is accompanied by a label. The label is represented as the correct output vector: $y = [o_7,\ o_8] = [1.00,\ 0.00]$, and has a distribution, p, with zero entropy as o_7 is infinitely more likely than o_8 for this input. The actual output of the model, $\hat{y} = [0.86, 0.14]$, can also be represented as a distribution, q. The cross-entropy loss, H, can then be calculated between the real distribution originating from the label, and the distribution of the models current predicted output given by the weights and biases of the model for the specific input:

$$H(p,q) = -\sum_{i=7}^{8} p_i\ log\ q_i = -\left[\frac{1}{2}\ log\ 0.86 + \frac{1-0}{2}\ log\ (1-0.14)\right] \approx 0.302 \tag{3.7}$$

Similar to MSE, the smaller cross-entropy loss, the closer the model is to predicting the correct class.

3.2.6 Gradient Descent

Gradient descent is an algorithm for finding a local minimum of a function. The function at hand must be differentiable, as the gradient descent algorithm calculates the steepest path on the curve to the nearest local minimum [65]. The gradient, ∇, of the function is a vector calculated at a given point by taking the derivative of the function. As an example, a simple two dimensional function is given in Equation 3.8.

$$y = f(x) = x^2, \quad \nabla = \frac{dy}{dx} = 2x \tag{3.8}$$

The iterative formula in Equation 3.9 moves a point in the negative direction of the gradient for each iteration. The learning rate, μ, adjusts the step size to travel down the function.

$$x_{k+1} = x_k - \mu\nabla f(x) = x_k - \mu\left(\frac{dy}{dx}\right)\bigg|_{x=x_k} \tag{3.9}$$

Here, the next state x position, x_{k+1}, is given by the current x position, x_k, plus some step, μ, in the direction of the negative gradient ∇. Thus, the total step is a product of both the steep of the gradient and the learning rate, all in the direction of the steepest descent.

Initially, the cyan colored dot in Figure 3.10 is located at $x_0 = 2$, and by using a learning rate of $\mu = 0.1$, the first 3 resulting iterations of the gradient descent algorithm is given in Equation 3.10, 3.11 and 3.12 respectively.

$$x_1 = x_0 - 0.1(2x_0) = 2 - 0.1(2 \cdot 2) = 1.6, \qquad y_1 = f(1.6) = 1.6^2 = 2.56 \quad (3.10)$$

$$x_2 = x_1 - 0.1(2x_1) = 1.6 - 0.1(2 \cdot 1.6) = 1.28, \qquad y_2 = f(1.28) = 1.28^2 = 1.6384 \quad (3.11)$$

$$x_3 = x_2 - 0.1(2x_2) = 1.28 - 0.1(2 \cdot 1.28) = 1.024, \quad y_2 = f(1.024) = ... = 1.048576 \quad (3.12)$$

The colored dots in Figure 3.10 represent different learning rates for the first 3 iterations of the gradient descent algorithm, where cyan is the initial point $x = 2$, blue is $\mu = 0.0125$, green is $\mu = 0.1$ and red is $\mu = 0.8$. The stippled lines represent the gradient corresponding to the point it is tangent to.

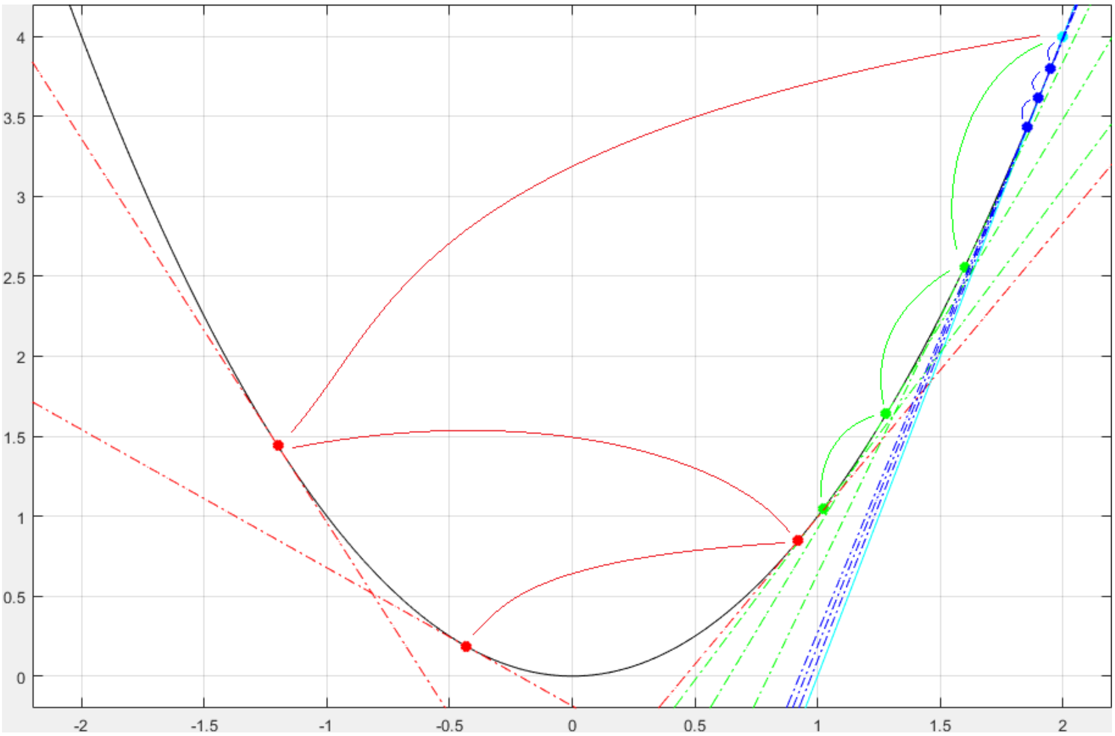


Figure 3.10: Plot of function $y = x^2$ (black) to illustrate gradient descent at different learning rates. Cyan is the initial point $x = 2$, blue is $\mu = 0.0125$, green is $\mu = 0.1$ and red is $\mu = 0.8$. Plot is generated in MATLAB.

To summarize, the learning rate affects how fast the gradient descent algorithm traverses down the path to the local minimum. Too large learning rate will cause an unstable learning process, where the weights are moved too much, possibly classifying a sample as a whole different class. Too small learning rate will make the learning process slow and could possibly get stuck in the process.

3.2.7 Backpropagation

Backpropagation is an algorithm that calculates the adjustment of the weights and biases in the NN, i.e. computing the gradient descent algorithm of the cost function to the NN [66]. The cost function to the NN takes in all its weights and biases as parameters, and the problem quickly becomes multidimensional in contrast to the simple two-dimensional example in Section 3.2.5. In order to apply backpropagation algorithm to the NN in Figure 3.9, we first must find the negative gradient of its cost function, $C(\cdot)$, which is given in Equation 3.13. Here it is assumed no bias in the input layers.

$$-\nabla C(w_{03}, w_{04}, ..., w_{67}, w_{68}, b_3, b_4, b_5, ..., b_8) \tag{3.13}$$

Notice that there is no learning rate involved in Equation 3.13. In theory, the function in Equation 3.13 is finding the steepest descent in a 27 dimensional space. Since the cost function is an average of the cost of all the training samples, the way to adjust the weights and biases depends on every single data in the training set.

The NN in Figure 3.9 has two classes. Intuitively, when we have an output like the one in Equation 3.6, we want to adjust the output of node o_7 to go from 0.86 to 1.00, and similarly the output of node o_8 to go from 0.14 to 0.00. Furthermore, the weights and biases leading to the output of node o_7 should be adjusted up, and the weights and biases leading to the output of node o_8 should be adjusted down. Also, this adjustment should be proportional to the difference in actual and correct output, i.e. +0.14 for node o_7 and -0.14 for node o_8. Looking at node o_7, its activation function is given in Equation 3.3. The three adjustable parameters are the bias, the weights and the activation function from the previous layer. The activation function in the previous layer can not be adjusted directly, however, the weights and biases leading from the input layer to the hidden layer can. Assuming that the ReLU activation function is used in the hidden layer, we can substitute Equation 3.14 into Equation 3.3, resulting in Equation 3.15.

$$h_x = max(i_0 w_{0x} + i_1 w_{1x} + i_2 w_{2x} + b_x, 0) \tag{3.14}$$

$$\begin{gathered}o_7 = \sigma([max(i_0w_{03} + i_1w_{13} + i_2w_{23} + b_3, 0)]w_{37} \\ +[max(i_0w_{04} + i_1w_{14} + i_2w_{24} + b_4, 0)]w_{47} \\ +[max(i_0w_{05} + i_1w_{15} + i_2w_{25} + b_5, 0)]w_{57} \\ +[max(i_0w_{06} + i_1w_{16} + i_2w_{26} + b_6, 0)]w_{67} + b_7)\end{gathered} \tag{3.15}$$

As the name suggest, the output of node o_7 has been propagated backwards until the result in Equation 3.15, which only contains all the adjustable weights and biases along with the three inputs. Calculating the gradient descent of the cost function based on the inputs $o_7 = 0.86$ and $o_8 = 0.14$, and adjusting the weights and biases based on this, would train the network only on that specific input. Hence, the backpropagation algorithm has to be performed for every single data in the training set, and finally take an average of all the desired adjustments in the NN. All these desired adjustments of the NN can be organized in a vector, and will then be proportional by a factor μ to the negative gradient of the cost function of the NN.

It is desired to know how much a change in the weights propagate a change in the total cost function, which is mathematically described in Equation 3.16. How much a change in weight w_{03} affects the cost function is really how much a change in weight w_{03} affects the output of node h_3, and how much that again affects the output node o_7 which in turn directly affects the cost function.

$$\frac{\partial C(\cdot)}{\partial w_{03}} = \frac{\partial h_3(w_{03}, ...)}{\partial w_{03}} \frac{\partial o_7(w_{37}, ...)}{\partial h_3(w_{03}, ...)} \frac{\partial C(\cdot)}{\partial o_7(w_{37}, ...)} \tag{3.16}$$

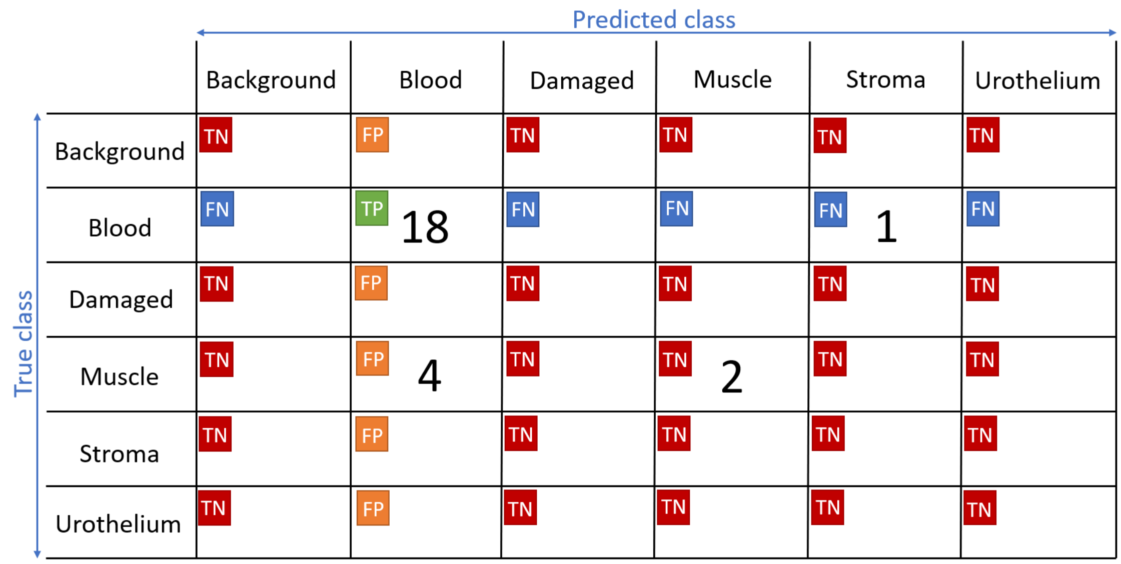
Without going into detail on all the partial derivatives relating to the cost function, the change in all the weights with respect to the total cost function must be found in order to obtain the desired adjustment of them all. Processing all the training data and calculating the desired adjustment for them all based on the backpropagation algorithm requires immense computational power. A method known as Stochastic Gradient Descent (SGD) optimizes these computations by randomly splitting the training dataset into smaller batches of equal size [67], see Section 3.2.9. An approximation of the gradient descent is then calculated by computing the gradient for each of these batches. SGD is also referred to as an optimizer, and there exists several other optimizers that utilizes the base principles of gradient descent in different ways.

3.2.8 Evaluation of Model Performance

The loss function refers to how a NN performs for a specific input. The backpropagation algorithm is performed to correctly adjust the networks interpretation of that data. Eventually the network must be tested on new data it has not seen before, and ways to quantify its performance is needed. Different parameters for performance is presented in this section.

Multiclass confusion matrix

The confusion matrix is a useful tool to analyse a models performance, and several parameters can be extracted from it. As an example, the confusion matrix of a multiclass model is given in Figure 3.11. Here, a NN is trained in a similar fashion to the once in this thesis, with all six classes.



Predicted class

True class

	Background	Blood	Damaged	Muscle	Stroma	Urothelium
Background	TN	FP	TN	TN	TN	TN
Blood	FN	TP 18	FN	FN	FN 1	FN
Damaged	TN	FP	TN	TN	TN	TN
Muscle	TN	FP 4	TN	TN 2	TN	TN
Stroma	TN	FP	TN	TN	TN	TN
Urothelium	TN	FP	TN	TN	TN	TN

Figure 3.11: Example of a multiclass confusion matrix for the six classes used in this thesis. With respect to class Blood: Green = True Positive, Red = True Negative, Orange = False Positive, Blue = False Negative.

The confusion matrix present all the data that the model has been able to classify, in a way that allows for an easy understanding of its performance. In this example, the model is not very good, and has only been able to classify 25 tiles. 19 of those tiles were blood, and the remaining 6 tiles were muscle tissue. For blood tiles, all but one was classified correctly, with one blood tile incorrectly predicted to be stroma tissue. 2 muscle tissue tiles were classified correctly, and 4 muscle tissue tiles were classified incorrectly as blood tiles.

For the rest of the cells in Figure 3.11, cells located in the pattern for a identity matrix, where predicted class meets true class, is referred to as true positive (TP). TP is the number of correctly classified tiles for the respective class. For blood, true negative (TN) refers to all the tiles with true class other than blood, that were classified as anything else than blood. False negative (FN) refers to the number of positive samples that were classified to be negative samples. For blood, that is the number of true blood tiles classified as another class than blood. Finally, false positive (FP) indicates how many samples of class negative that were classified to be class positive, or the number of predicted blood tiles whose true class was another. A healthy looking confusion matrix will have the majority of numbers in these diagonal cells, and few in the rest.

Accuracy

The accuracy describes how accurate the model is with regards to all classes. This number describes how large percentage of all test data were correctly classified as their respective class. P is the total number of positive samples, and N is the total number of negative samples. For the example in Figure 3.11, the accuracy is given in Equation 3.17.

$$Accuracy = \frac{TP + TN}{P + N} = \frac{18+2}{18+4+1+2} = 0.8 = 80\% \tag{3.17}$$

Precision

Precision is calculated per class, and indicate how many of the samples classified as being positive actually were correct. For blood this become the rate of true blood tiles out of all tiles predicted as blood. The formula for precision is given in Equation 3.18, along with precision for class blood in the example in Figure 3.11.

$$Precision\{blood\} = \frac{TP}{TP + FP} = \frac{18}{18+4} \approx 0.8182 = 81.82\% \tag{3.18}$$

Recall

Recall, also referred to as sensitivity, is also calculated per class, and indicate the true positive rate, i.e. rate of true blood tiles out of all tiles with true label blood. The recall for class blood from the example in Figure 3.11 is given in Equation 3.17.

$$Recall\{blood\} = \frac{TP}{TP + FN} = \frac{18}{18+1} \approx 0.9474 = 94.74\% \tag{3.19}$$

Specificity

The specificity function describes how robust the model is at rejecting true negatives. For blood tiles this becomes the rate of tiles predicted as other tissue types than blood out of all the tiles that are not true class blood. The formula can be seen in Equation 3.20.

$$Specificity\{blood\} = \frac{TN}{N} = \frac{2}{3} \approx 0.6667 = 66.67\% \tag{3.20}$$

F_1 Score

The F_1 score combines both precision and recall in one measure, as shown in Equation 3.21. A F_1 score of 100 % is the equivalent of perfect precision and sensitivity.

$$F_1\{blood\} = 2 \cdot \frac{Recall \cdot Precision}{Recall + Precision} = \frac{36}{36+4+1} \approx 0.8780 = 87.80\% \tag{3.21}$$

A good practice is to look at individual F_1-scores along with allover accuracy, as the F_1 includes both precision and recall. A perhaps even better practise is to study the confusion matrix to see if any significant misclassifications sticks out, like the example in Figure 3.11 where 2/3 of all muscle tiles get classified as blood.

3.2.9 Common Machine Learning Terms

Samples

One sample is one element in the dataset. For this thesis, one sample refers to one 128x128x3 tile at 400x magnification, one 128x128x3 tile at 100x magnification, and one 128x128x3 tile at 25x magnification, all centered at the same area in a WSI. It can be accompanied by a corresponding output vector or scalar, indicating how the model should interpret the data, i.e. the label [68].

Epoch

One epoch is defined as one pass-through of the whole training dataset. A good practice when training a model is to set a limit to how many epochs it should be trained for. A better practice is to set an acceptable limit to number of consecutive epochs with a validation loss smaller than some value, i.e. early stopping [68].

Batch-size

The dataset is split into mini-batches that are processed independently. The backpropagation algorithm is run for every batch. For a dataset with 1 000 samples, a batch-size of

100 would make one epoch take 10 mini-batches to complete. In general, the larger the batch-size, the faster the training process will be, and the more accurate the adjustment of weights will be each time as more data is involved. On the other hand, as batch-size increases, more available memory on the GPU is required, and will in many cases limit the mini-batch size to be below 256 samples [68].

Training dataset, test dataset and validation dataset

In the context of NN, a dataset is normally split into a training dataset and a test dataset. During training, the model is trained on the data in the training dataset. In order to evaluate model performance during training, a proportion of the training dataset is reserved for a validation dataset, typically the last 15 % of the dataset. The model evaluates performance by using the validation dataset at the end of each epoch. The validation dataset is not shuffled during training, but remains the same for every epoch to prevent multiple varying local minimums in the cost function. When the model is finished training, its performance is tested on the datatest set, which have not previously been involved in training for the particular model [68].

3.3 Convolutional Neural Networks

Convolutional neural networks are popular within the field of image processing, and differentiates from traditional FCNN in that not every node is connected to each other. Instead, a element-wise matrix multiplication is computed between the input and a kernel in order to detect features [69]. This makes the convolutional layers well suited to detect patterns in images, as the layer convolves a kernel across the entire image. In turn this allows for specific features to be detected in all parts of the image, and defines convolutional layers as shift-invariant. The kernel is, like the weights and biases, initialized with random numbers and as it is trained, and becomes more and more specific to what sort of feature it detects. For an intuitive approach, general convolution is presented first, followed by the operations within the convolution layer.

3.3.1 Feature Detection with Convolution

Figure 3.12 contains a kernel for detecting thin edges, an input and a corresponding feature map. The formula for discrete 2D convolution is given in Equation 3.22, where A and B represent two 2D matrices.

$$y(i,j) = A * B = \sum_m \sum_n A(i-m, j-n)B(m,n) \tag{3.22}$$

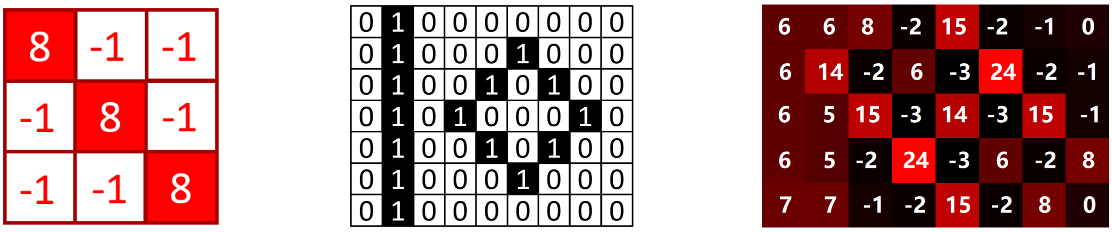
Applying the convolution formula between the 2D input image, A, in Figure 3.12b and 2D kernel, B, in Figure 3.12a, results in the first element produced at $i = 2, j = 2$ as is shown in Equation 3.23. The values computed for $i < 2$ and $j < 2$ would cause negative coordinates in the input image, and would require zero-padding. This is neglected here.

$$\begin{aligned} y(2,2) &= \textstyle\sum_{m=0}^{2} \sum_{n=0}^{2} A(2-m, 2-n)B(m,n) \\ &= A(2,2)B(0,0) + A(2,1)B(0,1) + A(2,0)B(0,2) \\ &+A(1,2)B(1,0) + A(1,1)B(1,1) + A(1,0)B(1,2) \\ &+A(0,2)B(2,0) + A(0,1)B(2,1) + A(0,0)B(2,2) \end{aligned} \tag{3.23}$$

It is important to flip the kernel both horizontally and vertically prior to convolution, as one of the matrices is always indexed by $-m, -n$ by the definition of convolution. By denoting the flipped kernel of size 3x3 as $\widetilde{B}$ where $\widetilde{B}(m,n) = B(2-m, 2-n)$, the convolution becomes as presented in Equation 3.24.

$$\begin{aligned} y(2,2) &= \textstyle\sum_{m=0}^{2} \sum_{n=0}^{2} A(2-m, 2-n)\widetilde{B}(m,n) \\ &= \textstyle\sum_{m=0}^{2} \sum_{n=0}^{2} A(2-m, 2-n)B(2-m, 2-n) \\ &= A(2,2)B(2,2) + A(2,1)B(2,1) + A(2,0)B(2,0) \\ &+A(1,2)B(1,2) + A(1,1)B(1,1) + A(1,0)B(1,0) \\ &+A(0,2)B(0,2) + A(0,1)B(0,1) + A(0,0)B(0,0) \end{aligned} \tag{3.24}$$

This is the equivalent of element-wise multiplication of the two matrices $A(0:2, 0:2)$ and B. Similarly, the remaining values for $y(i,j)$ is calculated to obtain the output in Figure 3.12c, which is referred to as the feature map. In the feature map, the features in the input image, in this case features of a 45° line, are detected in space and identified with larger numbers corresponding to how similar the feature is to the kernel.



8	-1	-1
-1	8	-1
-1	-1	8

(a) Kernel *B*.

0	1	0	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0
0	1	0	0	1	0	1	0	0
0	1	0	1	0	0	0	1	0
0	1	0	0	1	0	1	0	0
0	1	0	0	0	1	0	0	0
0	1	0	0	0	0	0	0	0

(b) Input *A*.

6	6	8	-2	15	-2	-1	0
6	14	-2	6	-3	24	-2	-1
6	5	15	-3	14	-3	15	-1
6	5	-2	24	-3	6	-2	8
7	7	-1	-2	15	-2	8	0

(c) Feature map y.

Figure 3.12: Convolution in 2D.

3.3.2 Convolutional Layers

For CNNs applied to RGB images, the input becomes three dimensional, and so does the kernel. One kernel for each color channel convolves across the entire image, and computes the element-wise multiplication. The resulting dot products are then summed up and stored as a single pixel in the output feature map, which is then passed to the next layer in the CNN. As an example, Figure 3.13 shows the output feature map after three kernels have been utilized in a convolutional operation with the three channels of the input image.

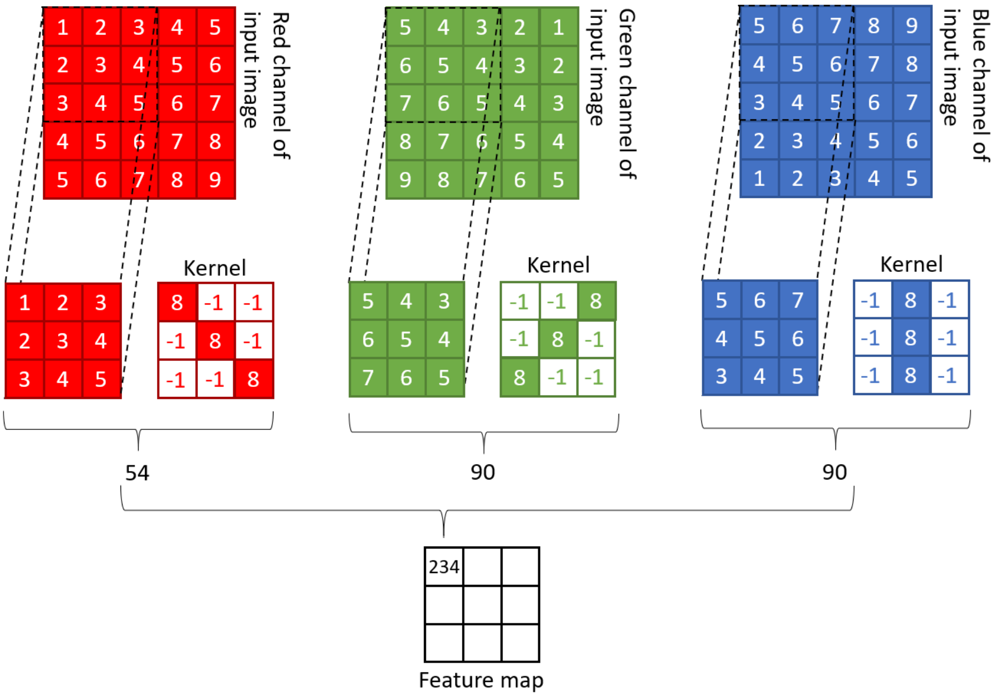


Figure 3.13: Illustration of convolutional operation for a 3D RGB input.

As the model is trained, different kernels are generated in the convolutional layers. As such, one can think of the kernel as a sort of filter that finds features that are similar to the kernel. The resulting output will be large if a specific area in the input image produces a high number when the element-wise multiplication is calculated with a kernel representing some feature. In other words, if a specific feature is located in the top left corner of an image during training, the same feature would be detectable during testing wherever it is located in the image. Convolutional neural networks also enables feature sharing, as features in the first convolutional layers may be shared with different kernels in the next convolutional layer when they are concatenated.

3.3.3 Pooling Layers

Pooling layers are often used in the combination with convolutional layers for down-sampling the feature map, reducing its dimensions. The two most common techniques are max-pooling and average-pooling. Max-pooling is typically used to pass forward only the most important level pixel in a fixed area in the feature map. This fixed area is referred to as a kernel, and slides over the image much like the convolutional layer. This makes it so that the most significant features detected in the convolutional layer will be forwarded to the next layers. The most common configuration is to use a 2×2 kernel that strides 2 pixels each time [70]. For average-pooling, the average of the pixels in the kernel is calculated and passed forward. Max-pooling is illustrated in Figure 3.14, where a 4x4 input image is reduced to a 2x2 output image.

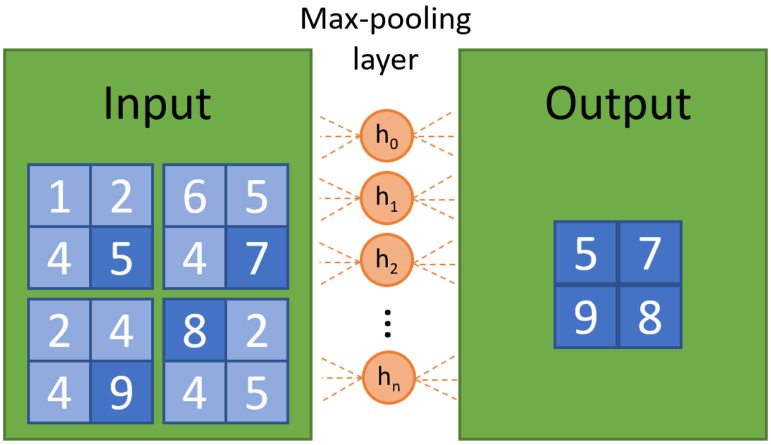


Figure 3.14: Simple illustration of the operation performed in a max-pooling layer.

Ultimately, the use of pooling layers reduces the computational load, as there are fewer weights and biases in the layers after the max-pooling layer. Its usage can also help to diminish issues related to overfitting, as explained further in Section 3.3.5.

3.3.4 Fully-connected Layers

Fully-connected layers, in the context of CNNs, are the layers located after the convolutional and pooling layers, and receives features vectors from them. The ultimate task of the fully-connected layers is to convert all the features vectors from the convolutional layers into the final output layer where each node represent a class. The first layer in the fully connected layer is often referred to as a flatten layer, as it transforms the feature map from the previous convolutional or pooling layer into a vector. The goals is that if a specific node in the flatten layer has a high output, then a certain feature exists in the input.

After the flatten layer, a number of fully connected layers will typically follow. This part of the CNN behaves like a normal FCNN, with all its weights and biases being adjusted based on the a backpropagation algorithm. The activation function used is typically ReLU, and in the final output layer either Sigmoid or Softmax, depending on the class problem.

3.3.5 Dropout Layers

A problem with small datasets in deep neural networks is overfitting, and the network may seek out to extract unwanted features from the training set, and further base its knowledge on these when classifying. Overfitting is a term used for deep neural networks that has too high capacity to capture the wanted features, and alternatively instead finds more detailed features. Underfitting is used to describe networks that have too low capacity, and cannot fully grasp the extent of the wanted features. These unwanted features may not be visible to the human eye, like repeated low noise in a texture.

Dropout layers are a way of dealing with such, and effectively sets the input equal to zero for a predefined percentage of nodes at random in the layer [71]. What nodes in the layers that are dropped updates at a predefined frequency each step when the model is trained. Using dropout layers is referred to as a regularization method, as it allows the network to focus on a smaller amount of features when some of the nodes are dropped.

Dropout only applies during training of the network, but can cause issues as the sum of all inputs is altered with fewer nodes present. To compensate for this, the remaining nodes in the dropout layer are scaled up by the inverse ratio of the dropout rate [72]. When the network is finished with training, dropout is no longer used.

3.4 Transfer Learning

Transfer learning in neural network terminology refers to using a NN that has been trained for another task in a new NN in order to solve a new task. There are large amounts of parameters to go about when setting up a new NN. How many layers, number of neurons per layer, learning rate, dropout, activation functions etc. Depending on how relatable the task in the trained NN is to the task in the new NN, the gain from transfer learning will vary. Intuitively, the initial layers of the deep neural network can be viewed as feature detection. One may also think of the last layers as the most task-specific object detection layers. Therefore, typically the first layers are inherited from the old network and the last layers are trained from scratch [73].

A network may be transferred in full, or partially, to a new model. This can be done as the low-level features, like vertical or horizontal lines and edges, exist in both the dataset for the transferred model and the new model. In addition, one can select to train the adopted part of network, or keep it as is. If the latter is chosen, the remaining parts of the new model has to be fine-tuned for it to fit the new task.

3.5 Supervised, Unsupervised and Semi-supervised Learning

Before going into details on the matter of semi-supervised learning, it is important to understand what supervised and unsupervised learning is. Supervised learning is the training of a NN by the use of labeled data. Moreover, the dataset consists of data that previously have been assigned a specific label indicating what class it belongs to. During training, data is passed through the network and the backpropagation algorithm is performed on the network for it to learn how to differentiate the different labels. Finally, the network is tested on labeled data that have not been used in training, in order to further quantify the performance of the model [74].

Unsupervised learning is the training of a NN by the use of unlabeled data. The dataset consists of data that has not been assigned any labels. Therefore, the network can only try and find new patterns that previously have not been detected. The technique often takes use of cluster analysis to identify areas of shared textures, patterns, sizes, colors or other attributes [74]. Another method in unsupervised learning is autoencoders, where the network effectively trains on encoding and decoding the original input. Intuitively, the input is compressed from its pure input state through a NN to a compressed state of smaller dimension. Then the image is reconstructed using the inverted encoder as a decoder. Further, the reconstructed image is compared to the original image, and the weights and biases are adjusted based on this [15]. Finally, the encoder can be replaced with a classifier network in order to associate the most compressed features to different classes in the output layer.

Semi-supervised learning is a branch within machine learning that falls somewhere between supervised and unsupervised learning. Both labeled and unlabeled data is used to train the NN, often more of the latter. It is beneficial in cases where there is small amounts of labeled data, but large quantities of unlabeled data [74]. Training is performed in different ways depending on the assumptions made about the unlabeled data. Data of same label are likely to be clustered together, and can be exploited by the use of clustering involved in the semi-supervised learning process. Another semi-supervised method is called self-training, and aims to train a NN on labeled data in a process often referred to as iteration 0. This iteration 0 model is then used to classify unlabeled data, and the NN is further re-trained in iteration 1 on the newly labeled data. More and more unlabeled data can be labeled for each iteration as the model hopefully gets more and more accurate.

3.6 Data Augmentation

As mentioned earlier, augmentation is a technique much used when the dataset consists of a small amount of labeled data. By making reasonable modifications to the labelled data, one can increase its size and further improve the performance of an AI model. There exists multiple ways of doing this from small variations in color to rotating entire images.

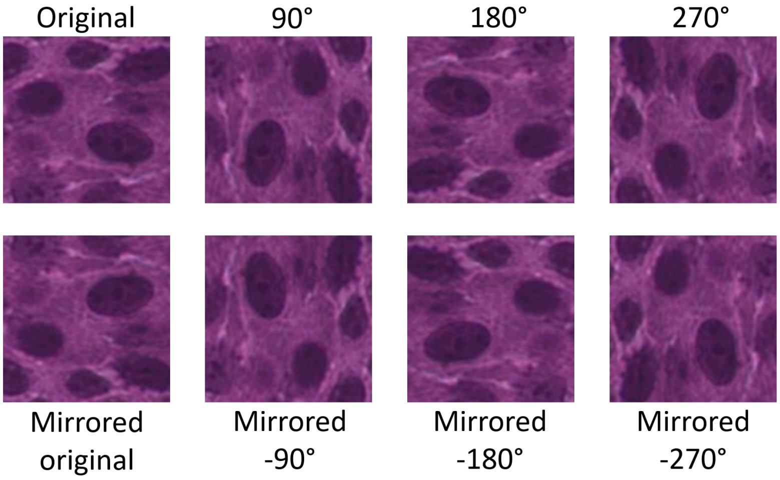


Figure 3.15: Augmentation of an original image (top left) to produce seven new augmented versions.

For histological images there are quite a few augmentation techniques that are applicable. When the tissue is put onto the mold as described in Section 2.4, the orientation of the tissue is not taken into account. The rotation of the tissue sample relative to the glass slides appears to be random, as the rotation of tissue is not taken into account during staging and grading. Hence, a valid augmentation technique is rotation, as shown in Figure 3.15.

Other techniques that can be applied to histological images include vertical and horizontal shift of images, random rotations of smaller degrees, augmentation based on different zoom levels, color and contrast, and more. Looking back at the convolutional layers in Section 3.3, a convolutional layer by itself is shift invariant in that the convolutional kernel is applied across the entire image, and features should be detect no matter where they are located. As such, augmentation by shifting is in general not considered a valid option for CNNs.

3.7 Label Types

With an underlying understanding of what labeled data is in the context of whether the data is labeled or not, this section focuses to enlighten knowledge on different types of labeled data. There exists no international organization to dictate how labels should be assigned to data of different origin, however, there is a general understanding of what can be considered ground truth labeled data and weakly labeled data.

Manually marked ground truth labels, or ground truth labeled data, is data that one can only assume has been assigned a true label [75]. In the context of medical images this usually means that an expert with a specific medical education and experience has identified and classified an image, or a part of an image. Even though different medical experts have different experiences, and hence might have different opinions on a label, ground truth labels are without a doubt more precise than data labeled by a person without a medical background.

Weak labels, or weakly-labeled data, is data that has been assigned an imprecise or inaccurate label. These labels distinguish from ground truth labels in that there was no expert involved directly in the labeling process. Weakly-labeled data can originate from unlabeled data that has been classified by a NN trained on strong-labels of same type. Weak labels might also refer to data where some meta-information exists to such a degree that it can be categorize in some way [76]. For instance, if there exist information with regards to recurrence about a patient with bladder cancer, then there exist a weak label indicating recurrence in the corresponding WSI. This is, however, not specific to a given area in the WSI.

In this thesis, ground truth labeled data originate from a pathologist at the University Hospital in Stavanger. The non-expert labels originate from non-expert annotations made by the author, and are not considered ground truth. An overview is presented in Table 3.2.

Table 3.2: Label types used in this thesis.

Label name	Label origin
Ground truth	Ground truth annotations made by a pathologist.
Cluster-Weak	Automatically generated labels made by a deep learning model trained on ground truth labels. Labels are then selected with the cluster-based self-training method.
Non-expert	Manual annotations made by the author (O. N. Dalheim), and reviewed by co-supervisor (R. Wetteland).
Probability-Weak	Automatically generated labels made by a deep learning model trained on ground truth labels. Labels are then selected with the probability-based self-training method.

Material and Previous Work

This chapter give details on the dataset at hand, as well as insight on the previous work done by co-supervisor, R. Wetteland, that this thesis build on.

4.1 Data Material

The dataset consists of about 360 WSIs from individual patients that have undergone TURBT at Stavanger University Hostpital. Each WSI have, for privacy concerns, been assigned with a number n, and the notation WSI-n will be used to denote WSI from patient number n. To distinguish WSI with ground truth labels from the others, WSIs with ground truth labels start with WSI-0n, and unlabeled WSIs start with WSI-1n. There are seven datasets used in this thesis, four consisting of ground truth labels, two consisting of weak labels, and one of manual labels. An overview is presented in Table 4.1.

Table 4.1: Overview of datasets used in this thesis.
GT = Ground truth, C = Cluster-weak, NE = Non-expert, P = Probability-weak.

	Label	**Ba**	**Bl**	**Da**	**Mu**	**St**	**Ur**	**Total**
$D_{gt}\{train_1\}$	GT	21 423	16 949	28 452	8 061	3 614	25 151	103 650
$D_{gt}\{test_1\}$	GT	5 589	2 883	5 155	1 905	1 261	4 577	21 370
$D_{gt}\{train_2\}$	GT	21 423	16 949	27 404	8 061	3 614	26 467	103 918
$D_{gt}\{test_2\}$	GT	5 589	2 883	6 203	1 905	1 261	3 261	21 102
D_{pw}	P	20 300	20 036	20 176	20 416	20 229	20 082	121 239
D_{cw}	C	21 281	42,630	24 817	48 359	52 794	31 731	221 612
D_{ne}	NE	0	17 899	25 134	15 142	24 245	32 981	115 401
D_{dm}	P	100 011	100 348	100 487	100 221	100 046	100 121	601 234

The datasets D_{cw} and D_{pw} are both extracted from same WSIs in the unlabeled dataset through two different semi-supervised approaches, further explained in Sections 5.3 and 5.2.2 respectively. Dataset D_{ne} is extracted through non-expert annotations made by the author, and on the same set of WSIs from the unlabeled dataset, as explained in Section 5.4. Finally, dataset D_{dm} is extracted relatively similar to the probability-based dataset, further explained in the model duplication experiment in Section 6.5.

Manually Marked Ground Truth Dataset

On 37 of the WSIs, WSI-001 to WSI-037, annotations have been made by a pathologist from Stavanger University Hospital. The WSIs are annotated on the 400x magnification level, from where tiles are extracted during preprocessing, see Section 4.2.1. The manually marked ground truth dataset, D_{gt}, consists of 125,020 tiles extracted from these 37 patients. The dataset is then further divided into two training datasets and two testing datasets, with the difference that WSI-002 is swapped from the training dataset to the test dataset, and WSI-015 is swapped from the test dataset to training dataset. This was done after a thorough analysis of the misclassified images for models trained on $D_{gt}\{train_1\}$, and tested on $D_{gt}\{test_1\}$, revealed some unique features in WSI-015, see Section 6.3. An overview of the WSIs that make up the ground truth dataset is given in Appendix A.

Table 4.2: Ground truth labels that make up the dataset D_{gt}, and corresponding tiles.

	Ba	**Bl**	**Da**	**Mu**	**St**	**Ur**	**Total**
Total tiles	27 012	19 832	33 607	9 966	4 875	29 728	125 020
% of dataset	21.6%	15.9%	26.9%	8%	3.9%	23.8%	100%
WSIs	7	5	9	4	5	28	39
avg. tiles/WSI	3 859	3 966	3 734	2 491	975	1 062	3 205

Probability-weak, cluster-weak, and non-expert datasets

The labels used to produce the probability-weak dataset, D_{pw}, originate from the probability-based self-training method, explained in Section 5.2.2. For the cluster-weak dataset, D_{cw}, the labels originate from the cluster-based self-training method in Section 5.3. The non-expert dataset, D_{ne}, consist of labels originating from non-expert annotations made by the author. Finally, the model duplication dataset, D_{dm}, originate through a process similar to the probability-based self-training. An overview is given in Table 4.3. A more detailed overview can be found in Appendix A, or in the attached file *patients_model_duplication.txt*.

Table 4.3: Overview of datasets D_{pw}, D_{cw}, D_{ne} and D_{dm}.

	Ba	Bl	Da	Mu	St	Ur	Total
				Probability-weak dataset			
Total tiles	20 300	20 036	20 176	20 416	20 229	20 082	121 239
% of all	16.7%	16.5%	16.6%	16.8%	16.7%	16.5%	100%
WSIs	46	18	42	28	42	46	46
Avg. tiles/WSI	441	1 113	480	729	481	437	2 636
				Cluster-weak dataset			
Total tiles	21 281	42 630	24 817	48 359	52 794	31 731	221 612
% of all	9.6%	19.2%	11.2%	21.8%	23.8%	14.3%	100%
WSIs	34	23	35	29	41	26	44
Avg. tiles/WSI	626	1 854	709	1 668	1 288	1 220	5 037
				Non-expert dataset			
Total tiles	0	17 899	25 134	15 142	24 245	32 981	115 401
% of all	0%	15.5%	21.8%	13.1%	21.0%	28.6%	100%
Patients	0	36	36	25	37	41	43
Avg. tile/pat.		497	698	606	655	804	2 683
				Model duplication dataset			
Total tiles	100 011	100 348	100 487	100 221	100 046	100 121	601 234
% of all	16.63%	16.69%	16.71 %	16.67%	16.64%	16.65%	100%
Patients	74	40	93	50	93	99	99
Avg. tile/pat.	1 351	2 509	1 080	2 004	1 075	1 011	6 073

4.2 Previous Work

As previously mentioned, this thesis focuses on utilizing semi-supervised techniques to improve the model built by co-supervisor R. Wetteland. This section focuses on establishing a basic understanding of this multiscale model, and the architecture behind it. First, a brief explanation of the preprocessing routine is presented.

4.2.1 Preprocessing

The annotations in the WSIs are made on 400x magnification level, by a pathologist from Stavanger University Hospital. As and example, the annotations appear as the lines in the left image in Figure 4.1. The annotations were made in Aperio ImageScope, which allows for exportation of the annotations. The format of the output file is .XML, which describes the area as a list of single point coordinates in 3D space called vertices, even though the real dimension is 2D. An example of a XML-file from the process of producing ground truth labeled tiles is shown in Listing 4.1.

The XML coordinates eventually form a line surrounding the tissue that is labeled with a tag to indicate tissue type. The preprocessing algorithm then searches through the annotated area to find optimal start coordinates for the tiles, to fit as many tiles as possible inside the region. Each tiles is of size 128x128 pixels, and when the optimal coordinates are found, the tiles are extracted at three different magnification levels, as illustrated in Figure 4.1.

```
    <?xml version="1.0"?>
- <Annotations>
  - <Annotation>
    - <Regions>
      -<Region grade="1" creator="pathologist" tags="Urothelium">
        -<Vertices>
          <Vertex Z="0" Y="153586.5700245903" X="55817.54038623767"> </Vertex>
          <Vertex Z="0" Y="153580.6791154994" X="55802.8131135104"> </Vertex>
                                                  ...
          <Vertex Z="0" Y="153586.5700245903" X="55817.54038623767"> </Vertex>
        </Vertices>
      </Region>
```

Listing 4.1: XML-file example. Each WSI have a corresponding XML-file containing the coordinates for the ground truth annotations. Vertices for each region are stores as X, Y and Z coordinates.

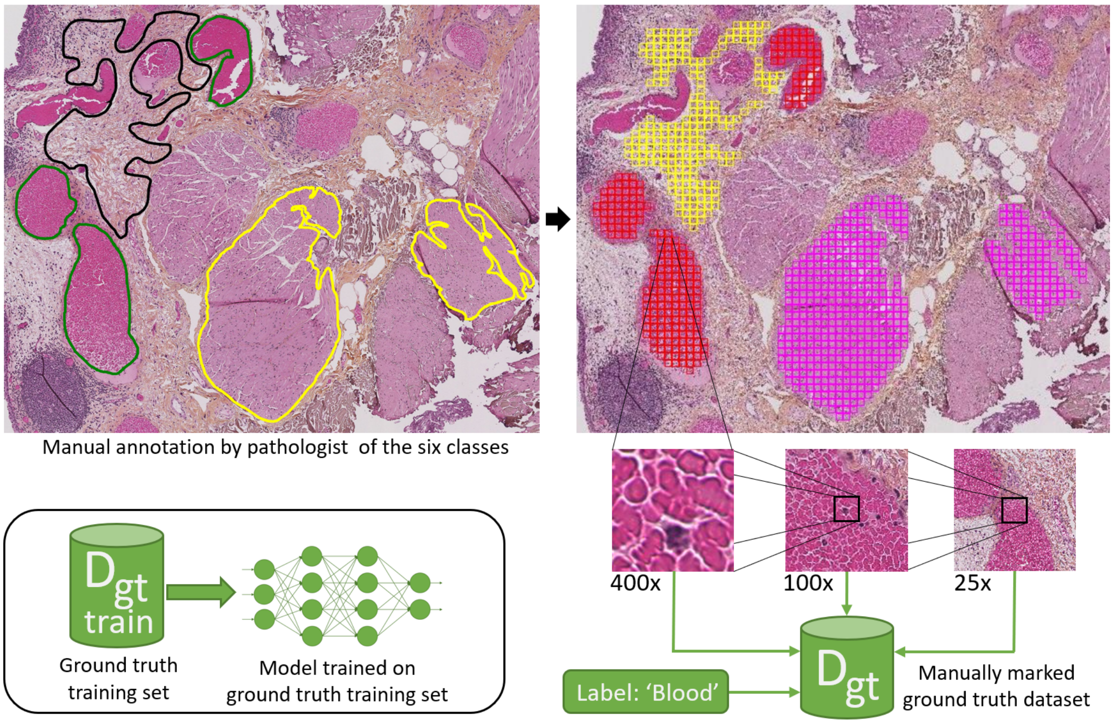

Manual annotation by pathologist of the six classes
400x
100x
25x
Dgt train
Ground truth training set
Model trained on ground truth training set
Label: 'Blood'
Dgt
Manually marked ground truth dataset

Figure 4.1: Origin of ground truth labels that are used to train models through supervised learning. Image on the left contains annotations, from which the tiles in the right image are extracted. The coordinates of the tile is then saved at three different levels of magnification, along with its corresponding ground truth label.

The lower magnification tiles (25x, 100x) have a larger field-of-view than the high magnification tile (400x), allowing the model to capture both local details and the surrounding context. The coordinates at the three magnification levels are then saved to the dataset D_{gt}, accompanied by its corresponding ground truth label.

4.2.2 Multiscale Model

This section presents the multiscale model referred to as TRI-CNN in Wetteland et al. [26]. The models architecture utilizes transfer learning, in that three individual VGG16 models are inherited, and used as the initial layers. As such, the VGG16 model is presented first.

VGG16

The VGG16 model [77] is a pre-trained CNN, originating from the University of Oxford. The model is trained on over 14 million images belonging to 1000 different classes from the ImageNet dataset [78]. As such, the model has identified a fair bit of features, which it uses 3x3 sized kernels to detect. An illustration of the network is given in Figure 4.2. The network consists of five sequential CNN blocks that each extract features, and compress them before they enter the next layer. From left to right: block 1 and 2 contain 2 convolutional layers each, followed by a pooling layer. The blocks 3, 4, and 5 contain 3 convolutional layers followed by a pooling layer. This is then further compressed to a single 1x1x512 fully connected layer after the last pooling layer of block 5. The final

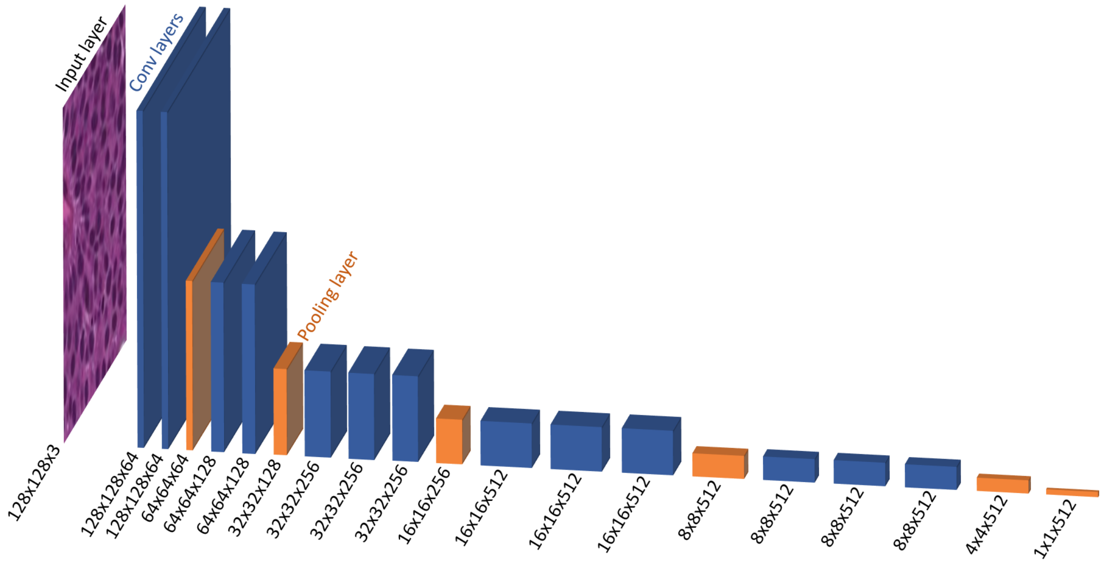


Figure 4.2: Illustration of the VGG16 model used through transfer learning. The blue boxes represent convolutional layers, orange boxes represent pooling layers.

layers of the VGG16 models are three fully connected layers followed by a 1x1x1000 Softmax output layer, that are not inherited nor illustrated.

VGG16 can be implemented through the keras library, which allows for several modifications to be made in the model. One can select to have the weights set to random, or import the weights from the final model, as trained on the ImageNet dataset. Once the VGG16 model is incorporated into a new model, the complete model can be trained as a whole, or one can choose to not update the weights in the VGG16 model. The pooling layers can be set to none, max-pooling or average-pooling.

TRI-CNN

In the work of Wetteland et al. [26], three architectures was explored, referred to as MONO-, DI- and TRI-CNN models, utilizing one, two and three VGG16 models in parallel. Results indicated the TRI-CNN model performing best, and thus is used in this thesis. The TRI-CNN model is a multiscale model that uses tile-wise classification, incorporated with three VGG16 models operating in parallel. Multiscale and TRI refers to the use of three different levels of magnification, where each VGG16 model is fed the same tissue area at different levels. The model is aimed towards guiding pathologists to diagnosis relevant areas of the WSIs, however, a future goal is to utilize the model in a computer-aided diagnosis system. An overview of the multiscale TRI-CNN model is given in Figure 4.3.

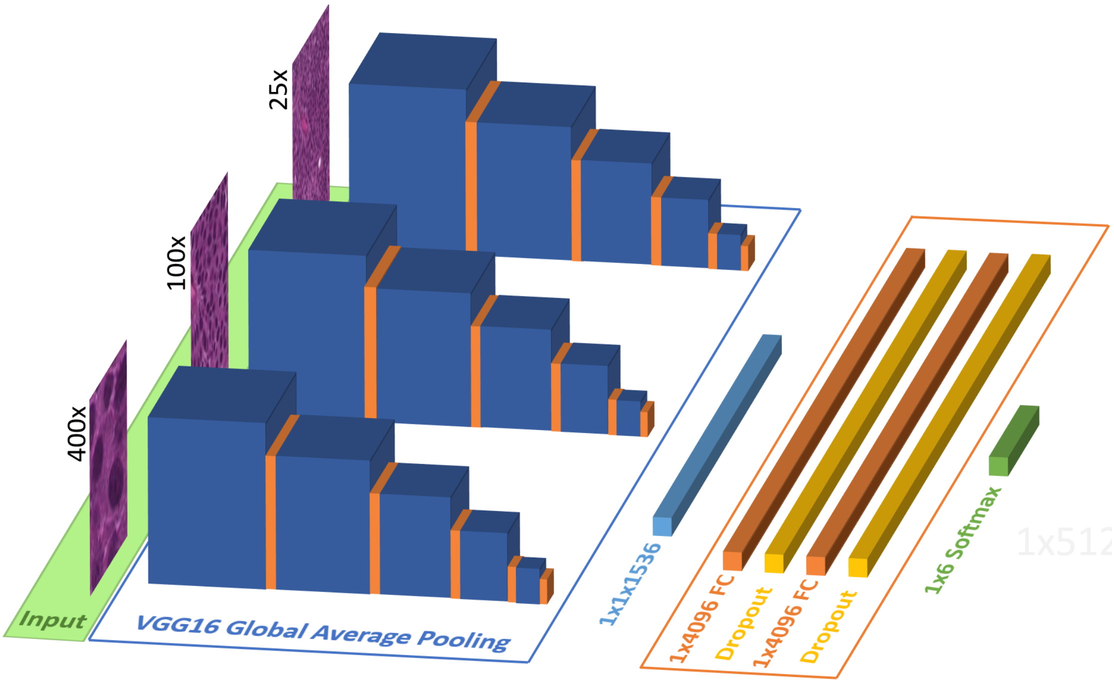


Figure 4.3: Illustration of the multiscale model, TRI-CNN.

The 1x512 output feature vector from each VGG16 network is fused into a single 1x1536 layer followed by a FCNN. The FCNN consists of an alternating sequence of first a fully-connected layer with 4096 nodes, followed by a dropout layer of same size. Two such FC-dropout layers are concatenated, before the final output layer, where a Softmax activation function is used as classifier.

Background tiles are filtered by a 10% threshold, where tiles close to white are discarded. Since a white 8-bit pixel value is represented as 255 in all three RGB values, the top 10 percent will capture tiles that are very close to white, which for the most part is background tiles. As the tissue is stained with specific colors, it will not not be discarded by the filter. Debris of significant size and color, pure ink marks, and similar will not be discarded, and must be classified by the model.

The model is implemented in Python 3.5, with TensorFlow 1.13 [79] and Keras 2.3 [80]. Scikit-learn [81] is used to evaluate the models, and PyVips [48] is used to process the WSIs. All tiles are saved as pickle files [82], in the architecture of Wetteland et al. [26]. Each WSI has a corresponding pickle file that maintains all the tiles belonging to it. Each tile contains three coordinate pairs, one for each level of magnification, along with a label indicating class, and a path identifying which WSI it relates to.

Methods

This chapter gives an in-dept description of the two methods within semi-supervised learning proposed in this thesis. Different methods were developed and tested to produce the best suited labels to increase accuracy, and also performance related to segmentation. Additionally, the same WSIs used in the two semi-supervised methods were labeled by hand by a non-expert to compare results, which is also described in this chapter.

5.1 Initial Supervised Approach

Several models were trained through supervised learning on the ground truth dataset D_{gt}, and an overview on some is given in Section 6.1. This was to evaluate differences in performance when using different levels of magnification, VGG16 models frozen or unfrozen during training, binary classification versus multiscale classification and more. A complete overview of the all models is attached to the thesis as *ALL_MODELS_OVERVIEW.xlsx*.

Models trained through a traditional supervised learning approach are denoted with SL. The letter A refers to augmentation by rotation of tiles being involved to increase the dataset, and is not denoted when no augmentation is used. GT indicate that the model is trained on ground truth labels only. F and U refers to the weights in the VGG16 models being frozen or unfrozen during training respectively. Finally the number 1 indicate that a model is trained on training dataset $D_{gt}\{train_1\}$, and 2 indicate that the model is trained on $D_{gt}\{train_2\}$.

For the different models trained on dataset $D_{gt}\{train_1\}$ and tested on $D_{gt}\{test_1\}$, the model with the highest accuracy and F_1-score across all classes but muscle, is referred to as TRI-GT-SL-AF-1. 2x augmentation of muscle and stroma tissue tiles was used during training of it, and the three VGG16 networks were frozen. TRI-GT-SL-AF-1 was used to predict labels in the 46 unlabeled WSIs, that later were utilized in the two SSL approaches.

5.2 Probability-based Self-training

The probability-based self-training (PBST) method is the most straight forward approach within self-training, designed to prioritize tile selection only according to its probability score across all WSIs. Each of the 46 unlabeled images are split up into tiles of size 128x128 pixels, and is classified by the original model, TRI-GT-SL-AF-1. Every tile that is classified with a minimum probability threshold of 60 % is saved, while tiles classified with lower probability are discarded. Tiles are then selected based on several criteria given in Table 5.1. All models trained using the PBST method are referred to as TRI-P-SSL. An illustration is given in Figure 5.1.

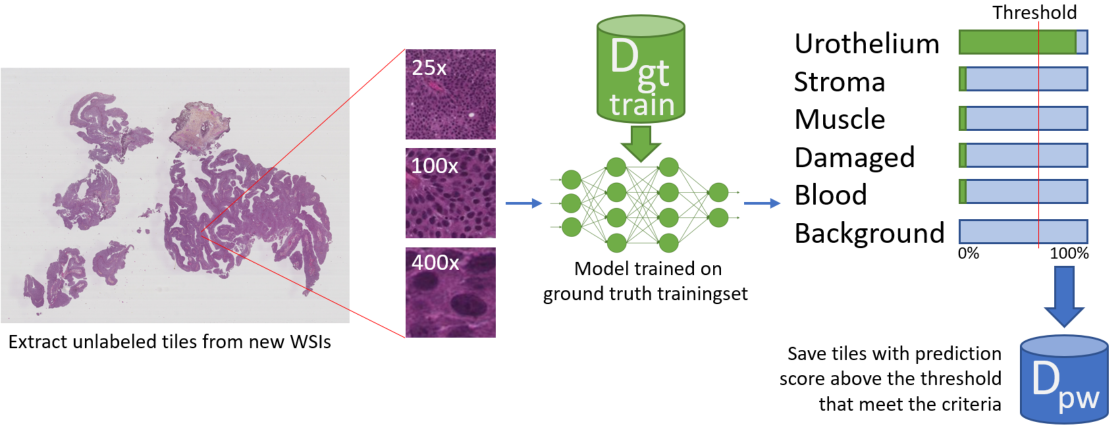

Extract unlabeled tiles from new WSIs
25x
100x
400x
D
gt
train
Model trained on ground truth trainingset
Threshold
Urothelium
Stroma
Muscle
Damaged
Blood
Background
0%
100%
Save tiles with prediction score above the threshold that meet the criteria
D
pw

Figure 5.1: Origin of probability-weak dataset, D_{pw}.
WSIs are predicted using a model train on ground truth tiles. The tiles are then selected through a probability-based approach.

Table 5.1: Tile criteria for the PBST method.

	Ba	**Bl**	**Da**	**Mu**	**St**	**Ur**
Min. tile probability	95%	80%	95%	95%	95%	95%
Max. tiles per WSI	1 900	8 000	710	5 000	5 000	480
Min. tiles per WSI	707	53	277	707	5 688	32 916
Max. tiles tot.	20 500	20 500	20 500	20 500	20 500	20 500

5.2.1 Criteria for Probability-based Self-training

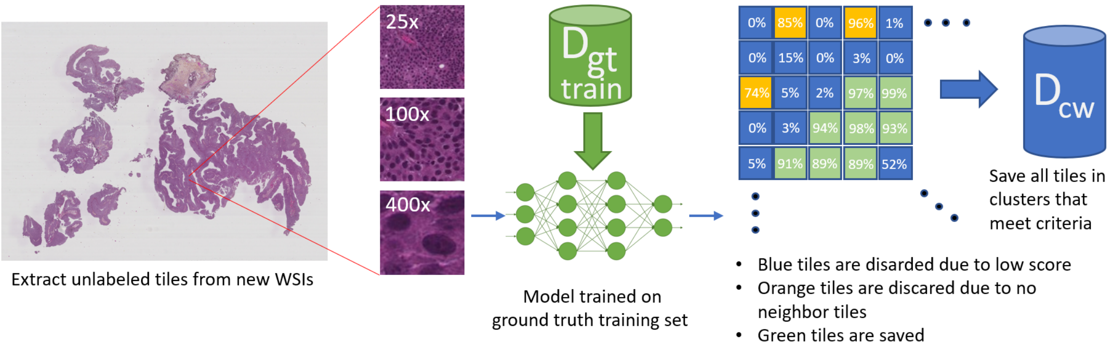
The criteria for the PBST method is set to prioritize tile probability, with a demand for all tiles to be of a minimum probability of 95 %, except for blood. For blood tiles, the probability demand is set slightly lower as a demand of 95 % for these tiles would discarded too many. The minimum tiles per WSI for blood is set very low for the same reason. Minimum tiles per WSI for urothelium is set very high, as most WSIs will normally have over a hundred thousand tiles for this class.

5.2.2 Functionality of Probability-based Self-training

As illustrated in Figure 5.1, the tiles of the WSIs are predicted by a model trained on ground truth labels. First, a scan runs through all the patients and counts the number of sufficient tiles per patient. Patients with an insufficient number of tiles according to the minimum number of tiles per WSI are discarded, and tiles are collected from the remaining patients. For each patient, tiles with the highest probability are collected first, until the maximum number of tiles per WSI has been collected, or no more sufficient tiles remain. All tiles from all WSIs are then appended to an array and sorted based on probability. The tiles with the highest probability are then selected from this array according to the maximum total number of tiles. This is done for each class, and later saved to the probability-weak dataset D_{pw}, as listed in Table 4.3.

5.3 Cluster-based Self-training

The cluster-based self-training (CBST) method bases the selection of tiles on both their probability score, as well as their location relative to other tiles. Similar to the PBST method, the cluster-based method uses model TRI-GT-SL-AF-1 to classify the same unlabeled 46 WSIs. The classified tiles with a minimum probability threshold of 60 % are then selected based on several criteria listed in Table 5.2. A illustration is given in Figure 5.2, and all models trained using the CBST method are referred to as TRI-C-SSL.


Extract unlabeled tiles from new WSIs
25x
100x
400x
D_{gt} train
Model trained on ground truth training set
D_{cw}
Save all tiles in clusters that meet criteria

0%	85%	0%	96%	1%
0%	15%	0%	3%	0%
74%	5%	2%	97%	99%
0%	3%	94%	98%	93%
5%	91%	89%	89%	52%

- Blue tiles are disarded due to low score
- Orange tiles are discared due to no neighbor tiles
- Green tiles are saved

Figure 5.2: Origin of cluster-weak dataset, D_{cw}.
WSIs are predicted using a model train on ground truth tiles. Tiles are then selected through a combination of probability and local neighborhood.

5.3.1 Criteria for Cluster-based Self-training

A cluster is defined as any tile that has one or more adjacent neighbor tiles of the same class and probability larger than 60 %. The criteria for tile selection based on the CBST method is set to prioritize clustered tiles, and to distribute tiles in the WSI as much as possible. The minimum number of tiles per WSI is set to prevent misclassified tiles being included. Normally, if there exist some tissue in a WSI, there will be a large number of tiles of this type. Blood and muscle is set slightly lower due to small blood clusters often appearing inside stroma tissue, and muscle tissue is often seen split up into smaller strains of tissue. In addition, muscle tissue is the least common tissue type in the WSIs.

The minimum cluster size is then set to the same number as minimum number of tiles per WSI, to further prevent including misclassifications. Max tiles per WSI can then be adjusted up and down until a satisfying number of tiles per class is obtained.

Table 5.2: Tile criteria for CBST method.

	Ba	**Bl**	**Da**	**Mu**	**St**	**Ur**
Min. tiles per WSI	50	20	50	20	50	50
Max. tiles per WSI	20 000	20 000	798	4 815	1 440	1 235
Max. clusters per WSI	100	100	100	100	100	100
Min. cluster size	50	20	50	20	50	50
Max. tiles per cluster	20 500	20 500	20 500	20 500	20 500	20 500
Min. avg. cluster probability	60 %	60 %	60 %	60 %	60 %	60 %

5.3.2 Functionality of Cluster-based Self-training

As visualized in Figure 5.2, a model trained on ground truth labels is used to make predictions in the new unlabeled WSIs. An algorithm searches through the WSI and groups tiles into clusters. The average cluster probability is calculated per cluster, and the clusters are sorted after the highest probability. Each cluster originating in the WSI is then sorted into an array, and the program selects the clusters based on the highest probability according to the maximum number of clusters. Then, the maximum number of tiles per WSI is divided by the number of sufficient clusters in the WSI, and an equal amount of tiles are extracted from each cluster. If, at any point in the search, the maximum number of tiles per cluster is not reached, the difference is appended to the limit of the next cluster in line. The tiles are then saved to the cluster-weak dataset D_{cw}, see Table 4.3.

5.4 Non-expert Annotations

In an effort to compare the automatic selection of tiles through the two SSL methods with straight forward manual annotations, regions in 43 of the same 46 unlabeled WSIs were annotated by the author. The labels are referred to as non-expert labels or non-expert annotations, as no medical expert was involved. The annotations were carried out some 4 months into the work, and a fairly basic understanding of the dataset, and the different tissue types, had been established. We had been given a guided tour at the Department of Pathology at Stavanger University Hospital, where the procedure from tumour to WSI was thoroughly illustrated. All annotations in the WSIs were done at 400x magnification level.

Before annotations were initiated, all the annotated regions in the 37 WSIs from the ground truth dataset were examined to increase tissue knowledge further, and also reviewed when in doubt. Some basic ground rules were laid down prior to the annotation process:

- Do not annotate areas if there are uncertainties with regards to tissue type
- Wherever possible, annotate at least one region of each tissue type per WSI
- Prioritize stroma and muscle tissue
- Use prediction maps from the TRI-GT-SL-AF-1 model WSIs to quickly locate large tissue regions

After all annotations were made, each individual area was saved as a jpeg-file extracted at 400x, and examined one by one to verify true class. Some minor adjustments were made, and the final set of annotations was reviewed and approved by co-supervisor Wetteland. Finally, the tiles were preprocessed in the same fashion as the ground truth labels, as previously explained in Section 4.2.1, and further saved to the non-expert dataset D_{ne}, see Table 4.3. All models trained using the non-expert labels are referred to as TRI-NE-SL.

5.5 Implementation

All methods are implemented in Python 3.5. Some files use PyVips [48] to process the WSIs, while most use Pickle [82] to manage tiles. A complete list of Python-scripts is given in Table 5.3, and all python scripts are located in the folder *python_files*, attached to the submitted PDF.

Table 5.3: Python files created or modified during thesis.

	Owner	Description
`CBST.py`	Author	Select tiles through CBST method
`PBST.py`	Author	Select tiles through PBST method
`pickle_combiner.py`	Author	Combine one pickle file representing each class into one
`pickle_model_dupl.py`	Author	Similar to PBST, but sufficient tiles are linearly spaced per WSI
`pickle_modifier.py`	Author	Create pickle files for two-class binary models
`pickle_modifier2.py`	Author	Remove tiles from pickle files by certain criteria
`plot_area_400.py`	Author	Plot pickle files/tiles in an area of a WSI
`plot_pickle_to_wsi.py`	Author	Plot pickle files/tiles in a whole WSI
`test_pickle_input.py`	Author	Count all tiles in all pickle files in a directory
`mode_7c.py`	R. Wetteland	Added functionality to save probability for each tile
`my_functions.py`	R. Wetteland	Added functionality to save probability for each tile
`main_prep.py`	R. Wetteland	Added functionality to extract tiles from histology website
`preprocess_region.py`	R. Wetteland	Added functionality to extract tiles from histology website
`MyFunctions.py`	R. Wetteland	Added functionality to extract tiles from histology website

Experiments and Results

In this chapter the performed experiments are presented along with information on experimental setup. Results are also provided as per experiment.

6.1 Models for Initial Supervised Approach

Several MONO binary two-class classification models were tried out to identify which of the lower magnification levels was best to identify tissue for the different classes. These can be seen in Table 6.1. Early on it was an idea to use multiple models in tile selection, models that were designed for a specific class at a specific level. Initially, the best performance was seen on the model DI-100x-25x from the works in Wetteland et al., however, it was a strong wish among both supervisors and myself to include the most detailed magnification level, i.e. 400x.

After training all the binary MONO two-class classification models, the corresponding binary two-class classification TRI models was trained, and improvement was seen across all classes. Some MONO six-class classification models were also tested for performance changes across different levels of magnification. The most important model was MONO-400x-ALL-AF-1 as it potentially was to predict tiles at 400x magnification. Unfortunately, this turned out to be the worst of the three MONO six-class models.

Table 6.1: Different models trained on the dataset D_{gt}.
MONO, TRI indicate how many magnification levels are used.

	Ba	**Bl**	**Da**	**Mu**	**St**	**Ur**
MONO-100x-BA-F-1	99.75%					
MONO-100x-BL-F-1		98.76%				
MONO-100x-DA-F-1			84.62%			
MONO-100x-MU-F-1				82.78%		
MONO-100x-ST-F-1					86.82%	
MONO-100x-UR-F-1						97.19%

Continued on next page

Table 6.1 – *Continued from previous page*

	Ba	**Bl**	**Da**	**Mu**	**St**	**Ur**
MONO-400x-BA-F-1	95.71%					
MONO-400x-BL-F-1		94.19%				
MONO-400x-DA-F-1			75.85%			
MONO-400x-MU-F-1				76.96%		
MONO-400x-ST-F-1					72.74%	
MONO-400x-UR-F-1						92.11%
TRI-BA-F-1	99.99%					
TRI-BL-F-1		99.20%				
TRI-DA-F-1			86.66%			
TRI-MU-F-1				84.70%		
TRI-ST-F-1					96.27%	
TRI-UR-F-1						97.80%
MONO-25x-ALL-AF-1	99.96%	92.61%	88.30%	72.52%	88.26%	92.87%
MONO-100x-ALL-AF-1	99.79%	98.03%	86.42%	80.17%	87.24%	96.80%
MONO-400x-ALL-AF-1	95.61%	94.37%	76.37%	78.48%	75.15%	91.38%
TRI-GT-SL-AF-1	100.00%	98.59%	89.14%	79.42%	96.44%	98.01%

The idea of using multiple models was soon discarded, and after using much of the same settings and parameters as in the newest publication of Wetteland et al. [26], the model TRI-GT-SL-AF-1 unfolded. This model had a good accuracy and relatively good F_1-scores for most classes, and the course was set to continue forth in a semi-supervised manner using this model.

During training of all models in Table 6.1, the learning rate was set to 0.00015 at a batch-size of 128. SGD was used as optimizer, and the dropout rate was set to 20 %. An early stopping criteria was set to end training after six consecutive epochs of change in validation loss smaller than 1e-6. All methods are implemented in Python 3.5, with TensorFlow 1.13 [79] and Keras 2.3 [80] as machine learning libraries. Scikit-learn [81] was used for evaluation, and PyVips [48] was used to process the images.

6.2 Self-Training vs. Non-expert Annotations

The purpose of this experiment is to compare the three methods CBST, PBST and non-expert annotations, to see which improves performance the most from the initial model TRI-GT-SL-AF-1, with regards to classification and segmentation. Two models were trained for each method, see Table 6.2.

SL is short for supervised learning, and SSL for semi-supervised learning. C indicate that the model is trained through the CBST method, P implies that the PBST method is used, and NE refers to models trained using non-expert labels. A refers to that augmentation by rotation of tiles is involved, and is not denoted when no augmentation is used. F and U refers to the weights in the VGG16 models being frozen or unfrozen during training

Table 6.2: Overview of the best models trained to compare SSL and non-expert annotations. TRI = using all three magnifications, SL = supervised learning, SSL = semi-supervised learning, C = cluster-based, P = probability-based, NE = non-expert, A = augmentation, F = VGG16 frozen, U = VGG16 unfrozen.

	Method	**Augm.**	**VGG16**	**Train**	**Test**
		VGG16 model frozen			
TRI-GT-SL-AF-1	SL	2xST,MU	Frozen	$D_{gt}\{train_1\}$	$D_{gt}\{test_1\}$
TRI-C-SSL-F-1	CBST	No	Frozen	$D_{gt}\{train_1\}$, D_{cw}	$D_{gt}\{test_1\}$
TRI-P-SSL-F-1	PBST	No	Frozen	$D_{gt}\{train_1\}$, D_{pw}	$D_{gt}\{test_1\}$
TRI-NE-SL-F-1	SL	No	Frozen	$D_{gt}\{train_1\}$, D_{ne}	$D_{gt}\{test_1\}$
		VGG16 model unfrozen			
TRI-GT-SL-AU-1	SL	3x	Unfrozen	$D_{gt}\{train_1\}$	$D_{gt}\{test_1\}$
TRI-C-SSL-AU-1	CBST	3x	Unfrozen	$D_{gt}\{train_1\}$, D_{cw}	$D_{gt}\{test_1\}$
TRI-P-SSL-AU-1	PBST	3x	Unfrozen	$D_{gt}\{train_1\}$, D_{pw}	$D_{gt}\{test_1\}$
TRI-NE-SL-AU-1	SL	3x	Unfrozen	$D_{gt}\{train_1\}$, D_{ne}	$D_{gt}\{test_1\}$

respectively. Finally, GT refers to only using ground truth labels during training. For all models trained with the three VGG16 models unfrozen, 3x augmentation is applied to all the foreground classes, i.e. blood, damaged, muscle, stroma and urothelium.

Models TRI-GT-SL-1 and TRI-GT-SL-AU-1 are trained through supervised learning on dataset $D_{gt}\{train_1\}$, see Table 4.2. The models based on the PBST method, TRI-P-SSL and TRI-P-SSL-AU, are trained on the labels in both $D_{gt}\{train_1\}$ and D_{pw}. TRI-C-SSL and TRI-C-SSL-AU are trained with the CBST method on labels from both datasets $D_{gt}\{train_1\}$ and D_{cw}. Models TRI-NE-SL-F-1 and TRI-NE-SL-AU-1 are trained on both datasets $D_{gt}\{train_1\}$ and D_{ne}. Finally, all models are tested on the same test dataset $D_{gt}\{test_1\}$.

6.2.1 Classification

Classification results for all eight models in Table 6.2 are presented in Table 6.3. During training of all models in Table 6.3, the learning rate, optimizer, dropout rate, and early-stopping was set to the same as for the models in Table 6.1. No weighting of the different labels in the datasets were used during training.

6.2.2 Segmentation

A new WSI, referred to as WSI-084, has recently been annotated by a pathologist, and has not been used during training before. As a way to further analyse the individual

Table 6.3: F_1-scores for each of the classes, and overall accuracy for the eight models. Green text indicate the best result within each category, if one unique exist. Acc = Accuracy.

	Ba	Bl	Da	Mu	St	Ur	Acc
TRI-GT-SL-AF-1	100.00%	98.59%	89.14%	79.42%	96.44%	98.01%	94.61%
TRI-C-SSL-F-1	99.99%	96.66%	90.55%	82.54%	95.93%	98.59%	95.12%
TRI-P-SSL-F-1	100.00%	98.64%	90.01%	82.68%	96.14%	98.29%	95.19%
TRI-NE-SL-F-1	100.00%	93.00%	89.37%	86.68%	99.33%	99.19%	95.10%
TRI-GT-SL-AU-1	100.00%	99.88%	87.86%	78.10%	98.10%	99.09%	94.57%
TRI-C-SSL-AU-1	100.00%	98.70%	91.88%	84.71%	95.92%	98.96%	95.99%
TRI-P-SSL-AU-1	100.00%	97.36%	88.21%	82.18%	96.79%	99.45%	94.85%
TRI-NE-SL-AU-1	100.00%	92.42%	89.07%	88.60%	98.48%	99.67%	95.20%

model performance with regards to segmentation, the WSI was split up into tiles and classified by all eight models in Table 6.3, with a minimum probability threshold of 60 %.

Small region

Figure 6.1 shows a small region in WSI-084 that contains a isolated area of blood cells. This area is particularly interesting as it is completely surrounded by background, in that larger parts of the lower magnification tiles (25x, 100x) will contain much background. The corresponding area, with the predictions by all models in Table 6.2, are shown in Figure 6.2.

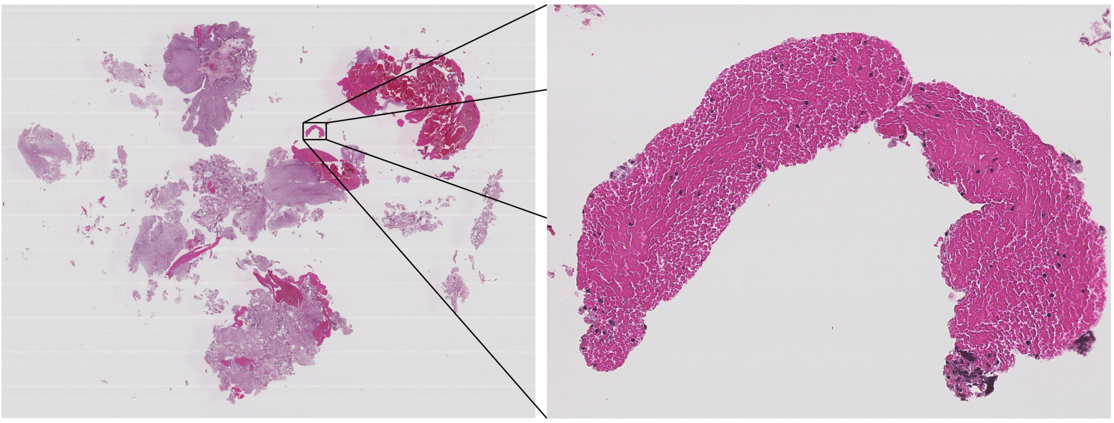
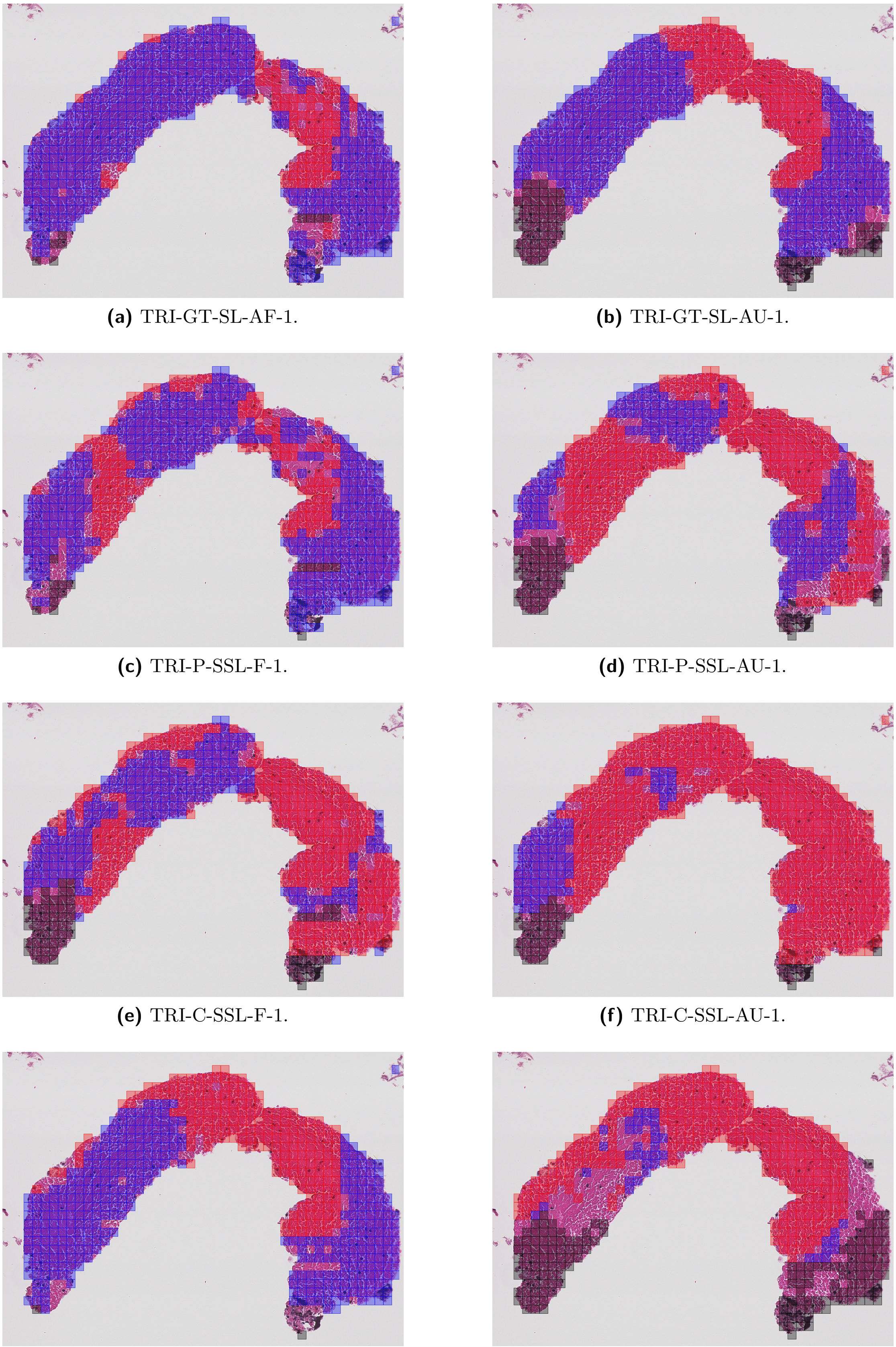


Figure 6.1: Small area of blood in WSI-084.

Model TRI-C-SSL-AU-1 performed best with regards to segmentation, as it classified most of the foreground as its true class, i.e. blood. The cluster-based self-training method also appears to produce the only models that reduces in number of background tiles when augmentation is applied. All methods are classifying more blood tiles instead of urothelium when augmentation is applied.



(a) TRI-GT-SL-AF-1.

(b) TRI-GT-SL-AU-1.

(c) TRI-P-SSL-F-1.

(d) TRI-P-SSL-AU-1.

(e) TRI-C-SSL-F-1.

(f) TRI-C-SSL-AU-1.

(g) TRI-NE-SL-F-1.

(h) TRI-NE-SL-AU-1.

Figure 6.2: Predictions for a region in WSI-084 with ground truth label blood. Color specifies predicted tile class: Blue = Urothelium tissue, Red = Blood cells, Black = Background.

Large region

Figure 6.3 shows a larger region in WSI-084 that contain different tissue types. Predictions for the corresponding area by models in Table 6.2 are shown in Figure 6.4 and 6.5.

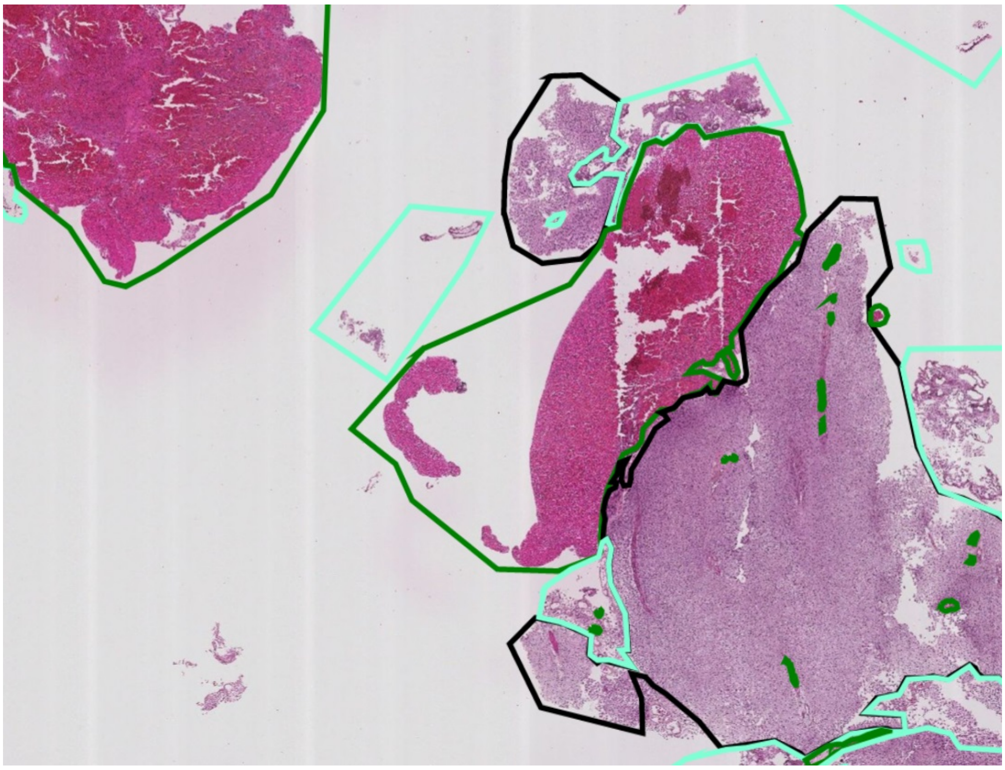
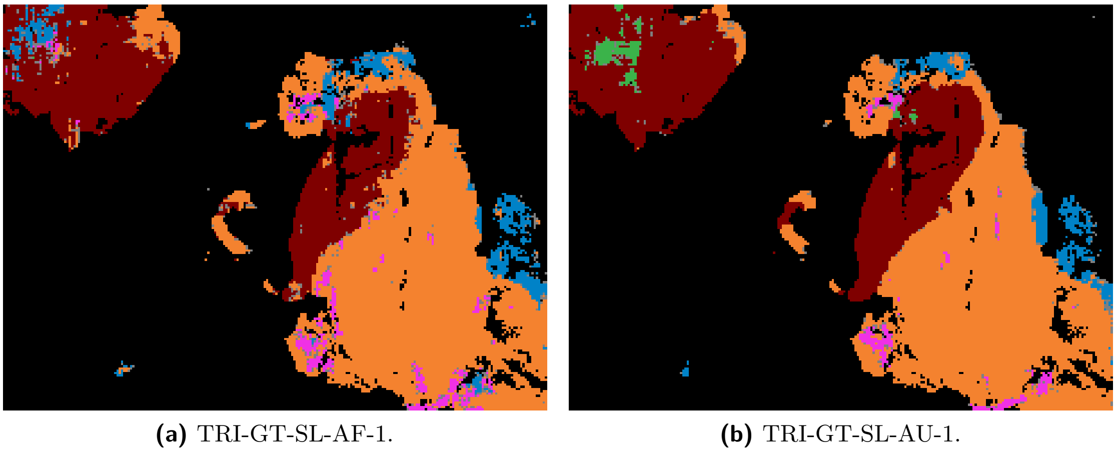


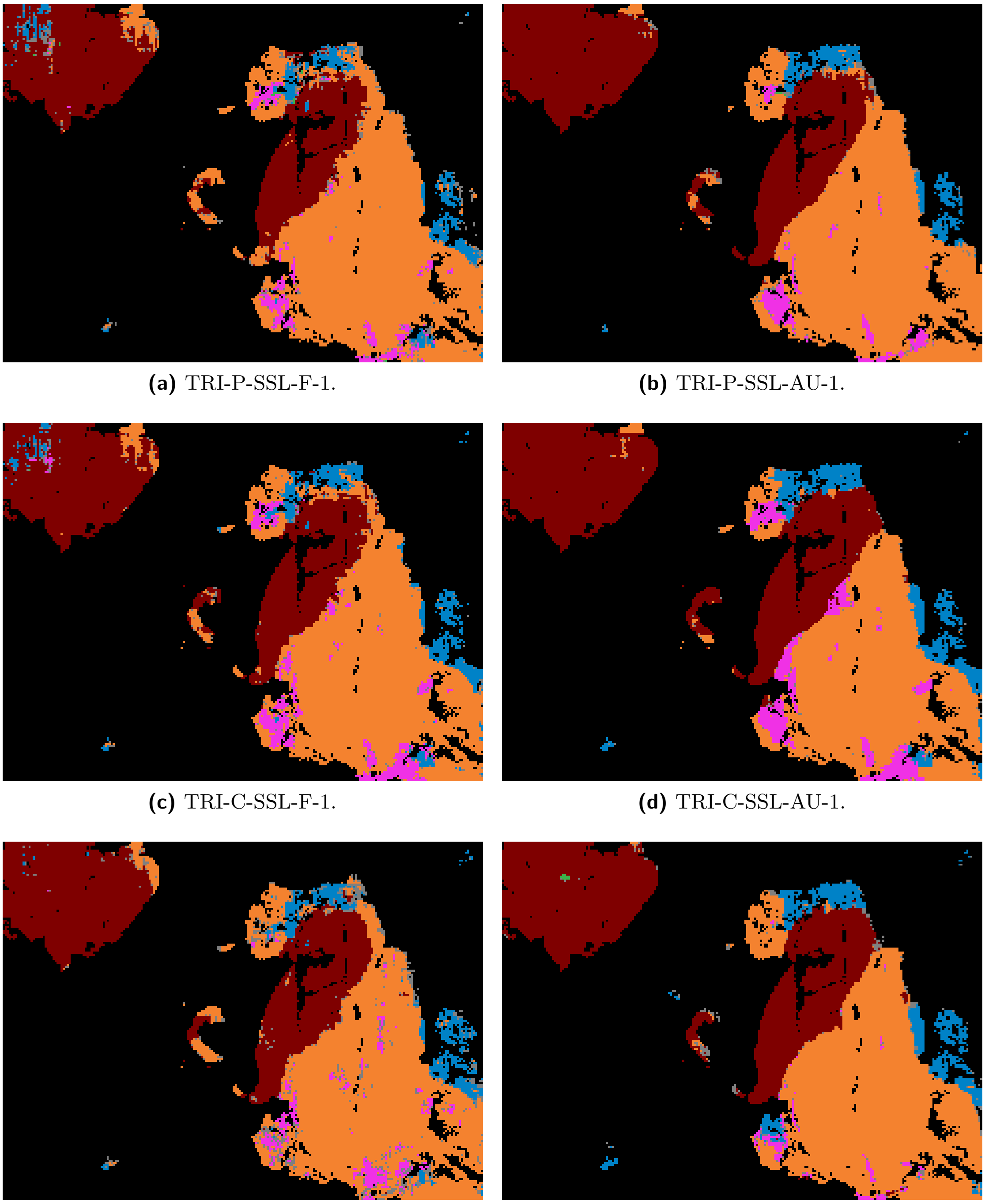
Figure 6.3: Ground truth annotations. Colours represent ground truth annotated areas: Green = Blood, Black = Urothelium, Cyan = Damaged.



(a) TRI-GT-SL-AF-1.

(b) TRI-GT-SL-AU-1.

Figure 6.4: Low magnification region in WSI-084.
Colours represent predicted labels: Red = Blood, Black = Background, Orange = Urothelium, Blue = Damaged, Pink = Stroma, Green = Muscle, Grey = Undefined.



(a) TRI-P-SSL-F-1.

(b) TRI-P-SSL-AU-1.

(c) TRI-C-SSL-F-1.

(d) TRI-C-SSL-AU-1.

(e) TRI-NE-SL-F-1.

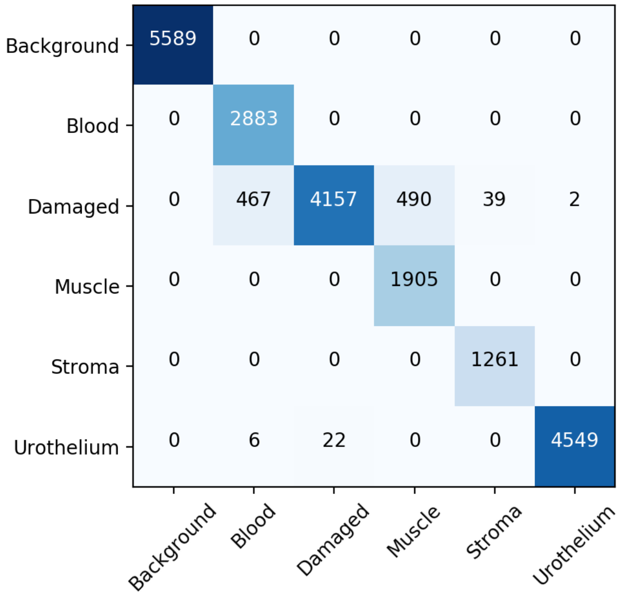
(f) TRI-NE-SL-AU-1.

Figure 6.5: Low magnification region in WSI-084.
Colours represent predicted labels: Red = Blood, Black = Background, Orange = Urothelium, Blue = Damaged, Pink = Stroma, Green = Muscle, Grey = Undefined.

Model TRI-NE-SL-AU-1 appears to be performing best, as it finds most of the smaller areas of damaged and blood tissue. It is also the model that found the least stroma, which there should be none of according to the annotations. It does, however, classify some blood cells as muscle.

6.3 Modification of Dataset

Looking back at Table 6.3, the model TRI-NE-SL-AU-1 has the best F_1-score for classes muscle, stroma and urothelium. Despite this, only achieving an accuracy about the same as the models where the VGG16 networks are frozen. The reason for this can be seen in its confusion matrix, in Figure 6.6. 998 damaged tiles out of the total 5155 damaged tiles that exist in the test dataset get misclassified, nearly 20 %. About half of those get classified as muscle, and the other half as blood. When analysing the individual tiles, almost every single one originate from the two damaged regions in WSI-015, especially the one displayed in Figure 6.7. Additionally, a tile that got misclassified as muscle by model TRI-GT-SL-AF-1 is depicted in Figures 6.9a, 6.9b and 6.9c, and a normal muscle tissue tile is depicted in Figures 6.9d, 6.9e and 6.9f.



	Background	Blood	Damaged	Muscle	Stroma	Urothelium
Background	5589	0	0	0	0	0
Blood	0	2883	0	0	0	0
Damaged	0	467	4157	490	39	2
Muscle	0	0	0	1905	0	0
Stroma	0	0	0	0	1261	0
Urothelium	0	6	22	0	0	4549

Figure 6.6: Confusion matrix for model TRI-NE-SL-AU-1 on test dataset $D_{gt}\{test_1\}$. Predicted class on the vertical horizontal axis, true class on the vertical axis.

This lead to a thorough analysis of the ground truth labeled dataset, and two things were noted: a) there exist several stroma regions in the ground truth labeled dataset that contain fairly large regions of blood, and b) there only exists such features as the ones in Figure 6.7 in one patient: WSI-015. Stroma can in-fact contain blood cells from a medical perspective, but from an engineers perspective this is not optimal with an individual class dedicated to blood. The damaged regions in WSI-015 were discussed with the pathologists, but they stayed true to their annotations. The areas were in fact damaged, and could possibly be burnt muscle or blood tissue.

A modification of the current dataset split in $D_{gt}\{train_1\}$ and $D_{gt}\{test_1\}$ was needed. Two options were considered, either completely remove the WSI from the dataset or transfer it to the training dataset. It was opted for the latter, as the dataset initially lacks tiles. WSI-015 was moved to the trainig set, replaced by WSI-002, which has a similar quantity of tiles and was moved to the training dataset. This gave birth to two new datasets, $D_{gt}\{train_2\}$ and $D_{gt}\{test_2\}$, as described in Table 4.2. With the new dataset, new models were trained with same parameters as previously, and a compelling performance increase was seen in both accuracy and F_1-score, as can be seen in Table 6.4. The tiles used in models TRI-P-SSL-U-2 and TRI-C-SSL-U-2 are from D_{pw} and D_{cw} respectively, meaning these tiles were predicted by a model trained on the original split of the dataset, $D_{gt}\{train_1\}$, and are arguably partially incomparable. All four models performs very well on the test dataset, but their ultimate purpose is segmentation of WSIs. The recently annotated WSI-084 was predicted with all four models, as shown in Figure 6.8.

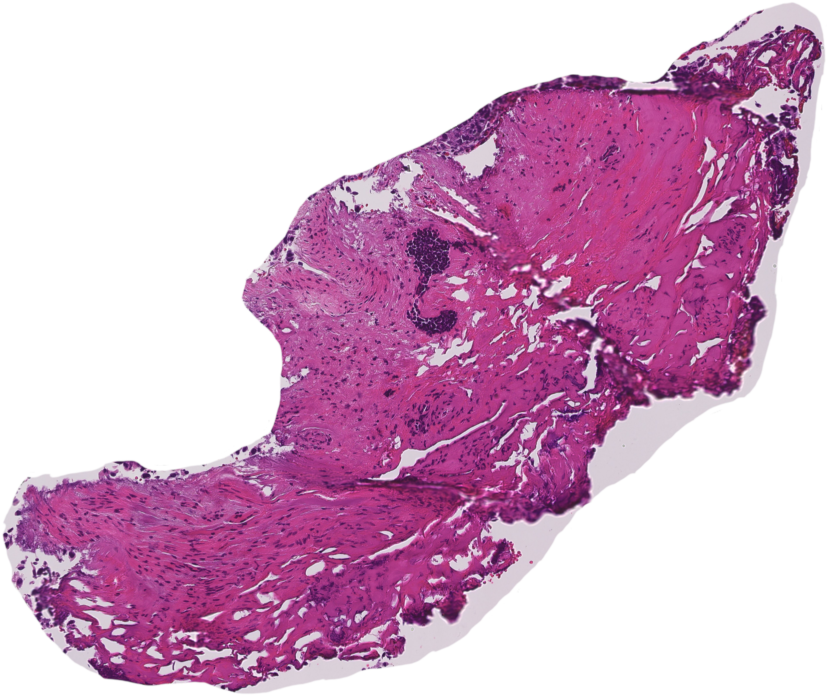
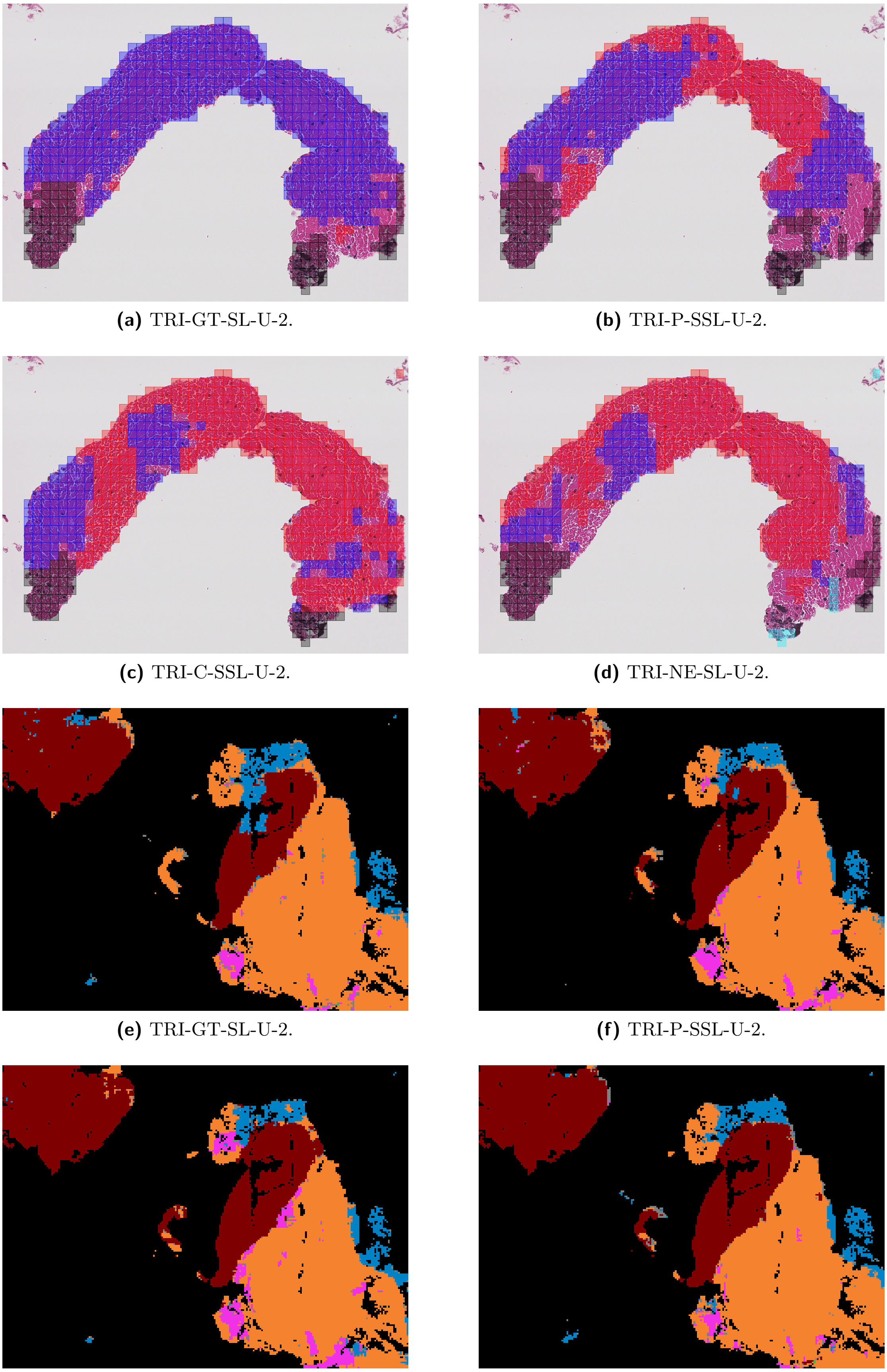


Figure 6.7: Damaged tissue in WSI-015 holding some unique features.

Table 6.4: F_1-Scores for models trained on dataset $D_{gt}\{train_2\}$, and tested on $D_{gt}\{test_2\}$. Green text indicate the best result within each category, if one unique exist. Acc = Accuracy. * trained on tiles predicted by a model trained on the original split of the dataset, $D_{gt}\{train_1\}$.

	Ba	**Bl**	**Da**	**Mu**	**St**	**Ur**	**Acc**
TRI-GT-SL-U-2	99.89%	99.70%	98.66%	98.43%	92.31%	98.49%	98.73%
TRI-C-SSL-U-2*	99.98%	99.72%	98.96%	97.44%	98.80%	99.25%	99.41%
TRI-P-SSL-U-2*	99.79%	99.91%	99.20%	97.96%	99.20%	99.63%	99.24%
TRI-NE-SL-U-2	100.00%	99.52%	99.90%	99.90%	100.00%	99.43%	99.43%



(a) TRI-GT-SL-U-2.

(b) TRI-P-SSL-U-2.

(c) TRI-C-SSL-U-2.

(d) TRI-NE-SL-U-2.

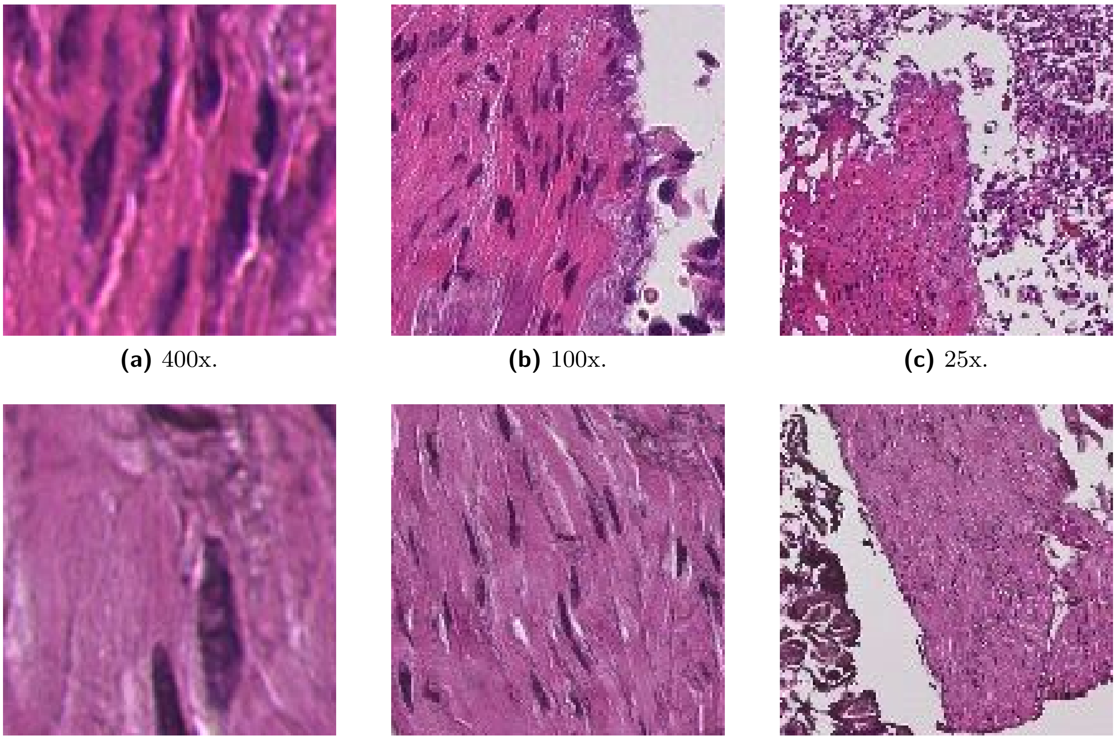
(e) TRI-GT-SL-U-2.

(f) TRI-P-SSL-U-2.

(g) TRI-C-SSL-U-2.

(h) TRI-NE-SL-U-2.

Figure 6.8: Predictions with models of the new dataset $D_{gt}\{train_2\}$ for a region in WSI-084 with ground truth label blood. Color specifies predicted tile class: Blue = Urothelium tissue, Red = Blood cells, Black = Background, Cyan = Damaged.


(a) 400x. **(b)** 100x. **(c)** 25x.

(d) 400x. **(e)** 100x. **(f)** 25x.

Figure 6.9: (a,b,c) Misclassified tile with ground truth label damaged in WSI-015. (d,e,f) Correctly classified tile with ground truth label muscle from another WSI.

6.4 Ground Truth vs. Non-expert Annotations

To measure the quality of non-expert annotations made by the author, an experiment was needed to test the two against each other. The non-expert dataset is split at an approximate ratio of 85/15 into a training set $D_{ne}\{train\}$, and a test set $D_{ne}\{test\}$. Since the dataset D_{ne} contain no background tiles, this is ignored. Thereafter, one new model is trained on only the non-expert training dataset $D_{ne}\{train\}$, and one model is trained on both $D_{ne}\{train\}$ and $D_{gt}\{train_2\}$. Finally, these two models, along with the model trained on ground truth labels only, TRI-GT-SL-U-2, are tested on both $D_{gt}\{test_2\}$ and $D_{ne}\{test\}$ individually, yielding the results presented in Table 6.5. All three models are trained with VGG16 unfrozen and no augmentation, and parameters for learning rate, optimizer, dropout rate, and early-stopping is set to the same as for the models in Table 6.1.

Table 6.5: Comparison of F_1-scores for model trained on ground truth labels versus model trained on both ground truth labels and manual labels. tr = train, te = test.

Training dataset	Test dataset	Bl	Da	Mu	St	Ur
$D_{gt}\{tr_2\}$	$D_{gt}\{te_2\}$	99.70%	98.66%	98.43%	92.31%	98.49%
$D_{gt}\{tr_2\}$	$D_{ne}\{te\}$	75.31%	90.55%	92.72%	80.77%	84.18%
$D_{ne}\{tr\}$	$D_{gt}\{te\}$	85.97%	85.77%	90.67%	99.80%	99.12%
$D_{ne}\{tr\}$	$D_{ne}\{te\}$	99.33%	98.16%	98.76%	93.23%	96.08%
$D_{gt}\{tr_2\}$,$D_{ne}\{tr\}$	$D_{gt}\{te_2\}$	99.93%	99.74%	99.63%	99.72%	99.82%
$D_{gt}\{tr_2\}$,$D_{ne}\{tr\}$	$D_{ne}\{te\}$	99.38%	98.30%	98.47%	96.17%	97.28%

6.5 Model duplication

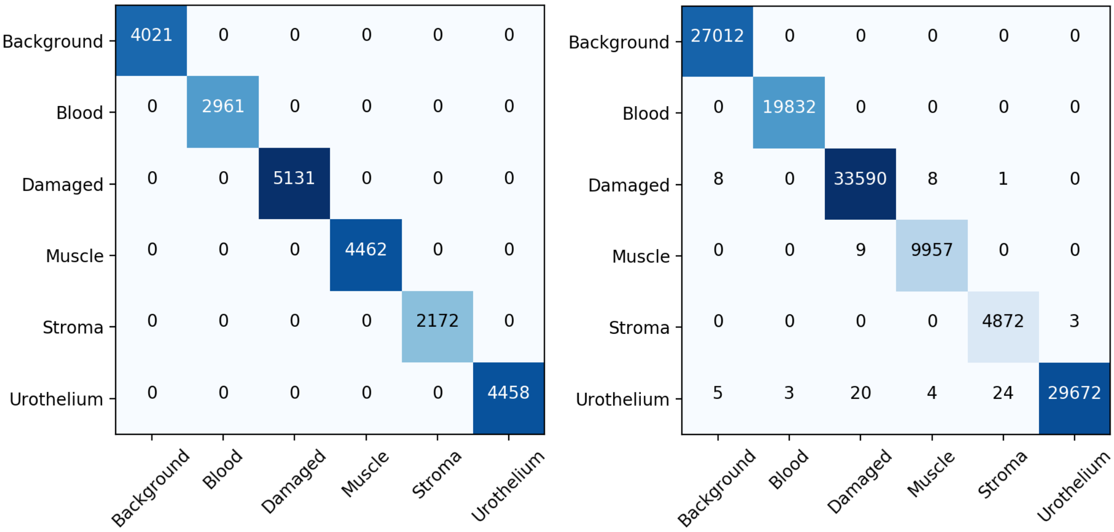
As an experiment, a new model, referred to as TRI-SL-GT-U, was trained on the entire ground truth dataset D_{gt}. The model was then used to produce predictions on 99 new unlabeled WSIs, which will produce millions of tiles for most classes. The tiles were then selected for the model duplication dataset D_{dm}, through a method very similar to PBST, with the criteria listed in Table 6.6. Finally, a new model is trained on the model duplication dataset D_{dm}, referred to as TRI-SSL-DM-U, which is further tested on the entire ground truth dataset D_{gt}.

Table 6.6: Tile criteria for model duplication dataset D_{dm}.

	Ba	Bl	Da	Mu	St	Ur
Min. tile probability	94.93%	96.67%	97.89%	96.64%	94.82%	97.99%
Min. tiles per WSI	100	100	100	100	100	100
Max. tiles tot.	100 000	100 000	100 000	100 000	100 000	100 000

First the mean probability is calculated per class from all tiles in all 99 WSIs. This is then set as the minimum tile probability criteria for the respective class. The method first runs through each WSI to count how many WSIs have enough tiles of sufficient probability. The number of viable WSI for a certain class is then used to calculate how many tiles should be extracted from each WSI, based on the maximum total number of tiles. Thereafter, the calculated number of tiles per WSI is extracted from each sufficient WSI, in such a way that the extracted tiles are spaced linearly. This is done to properly distribute tiles: imagine extracting 4 000 urothelium tiles from a WSI that holds 300 000 sufficient urothelium tiles. Finally, approximately 100 000 tiles are extracted for each class, and appended to the dataset D_{dm}.

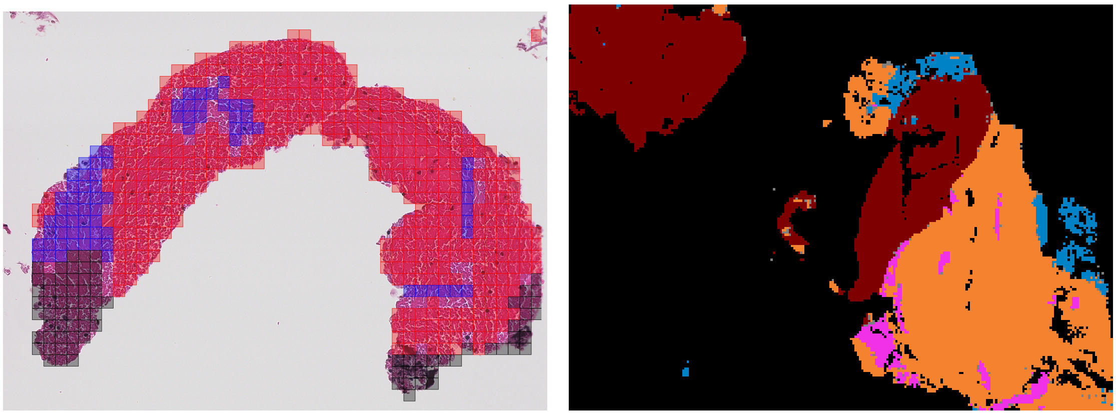
A new model, TRI-SSL-DM-U, is trained on the dataset D_{dm}, and then tested on all tiles in the dataset D_{gt}. Training was done with the VGG16 networks unfrozen, and without the use of augmentation. Parameters such as learning rate, optimizer, dropout rate, and early-stopping was set to the same as for the models in Table 6.1. The confusion matrix for the final validation test after trainig of model TRI-SL-GT-U in Figure 6.10a. The confusion matrix for the duplicated model, TRI-SSL-DM-U, when tested on the entire ground truth dataset, D_{gt}, is shown in Figure 6.10b, with an accuracy of 99.93 %. Finally, the same regions in WSI-084 as before were then predicted by the model TRI-SSL-DM-U, see Figure 6.11



(a) Validation test of original model, TRI-SL-GT-U.

(b) Test of duplicated model on all training data, TRI-SSL-DM-U.

Figure 6.10: Confusion matrices for the model duplication experiment. Predicted class on the vertical horizontal axis, true class on the vertical axis.



(a) Small region.

(b) Large region.

Figure 6.11: Predictions on WSI-087 by the model TRI-SSL-DM. Predicted class on the vertical horizontal axis, true class on the vertical axis. Color specifies predicted tile class: Blue = Urothelium tissue, Red = Blood cells, Black = Background, Cyan = Damaged

Discussions and Conclusion

This chapter offers a discussion on performance of proposed methods, things that went well and things that did not, future work and more. Finally, a conclusion on the best methods.

7.1 Discussions

Different methods and experiments have been performed, which will be discussed in this section.

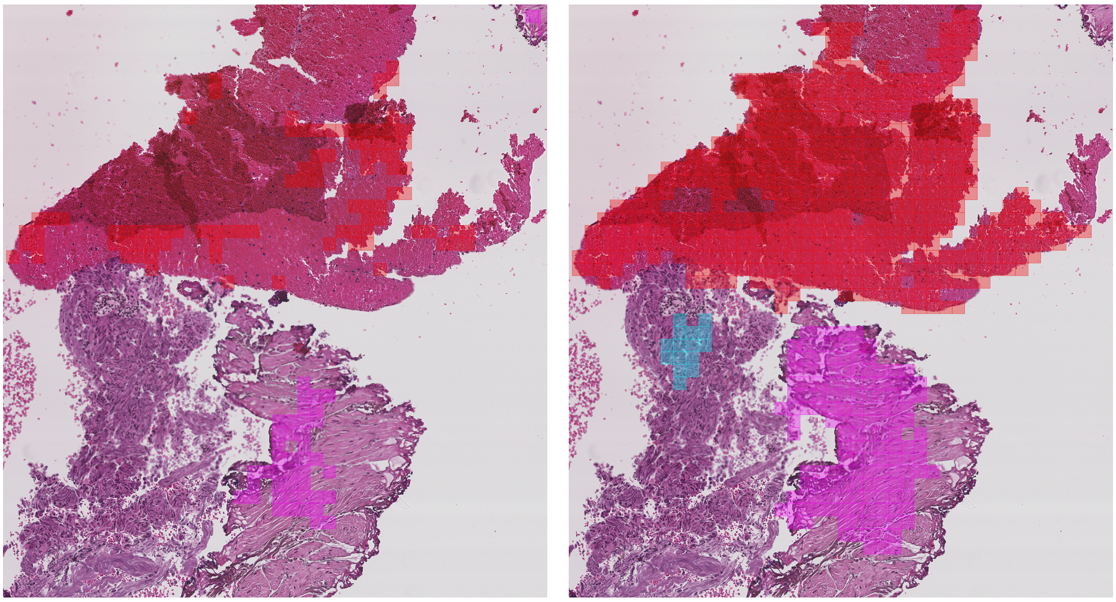
7.1.1 Tissue Selection

In this thesis, three main approaches was proposed to select tiles from unlabelled WSIs. For the two semi-supervised approaches, CBST and PBST, the distribution is typically like depicted in Figure 7.1. Non-expert labels are most similar to CBST in that typically entire regions are annotated.

Each WSI will on average produce hundreds of thousands of tiles. Often, if a model is trained with an unbalanced dataset dedicating a large portion of tiles to a specific class, that model will predict more tiles with higher probability for that class in new WSIs. This poses a challenge when selecting tiles through an automated self-training method. To counter this, different strategies were taken in the different methods.

A minimum number of tiles per patient was set to discard WSI containing few tiles for a particular class, as these tiles are more likely to be misclassified. Predictions are subjects to noise, and will typically associate a few random features to classes that are not necessarily in the WSI. Muscle and large blood regions seem to be appearing in about only half of the WSIs.

Another strategy throughout all SSL methods was to use a lower minimum threshold of 60 % with regards to classification. In theory, this threshold could have been set anywhere above 50 % for the model to classify only one class. The extra 10 % is added as a margin, and ensures that not too many tiles are excluded.



(a) PBST. **(b)** CBST.

Figure 7.1: Extracted tiles in an area of WSI-113 by using both self-training methods. Red = Blood, Cyan = Damaged, Pink = Muscle.

For PBST, this lower minimum of tiles per WSI does not prevent an over-representation of the top-left part of a WSIs, which can occur when several WSIs contain large amounts of sufficient tiles of a certain class. One may also argue that a model will not have a significant gain in associated features from tiles it already is 100 % certain about, and that the method becomes more of an alternative to augmentation. In an attempt to mitigate this issue, linear spacing of tiles within a WSI is added in the model duplication method. This ensures larger variation in the 25x and 100x magnification tiles, and hopefully introduces new features across all magnification layers. An improvement is seen with regards to segmentation, when comparing the models TRI-P-SSL-AU-1 and TRI-SSL-DM-U.

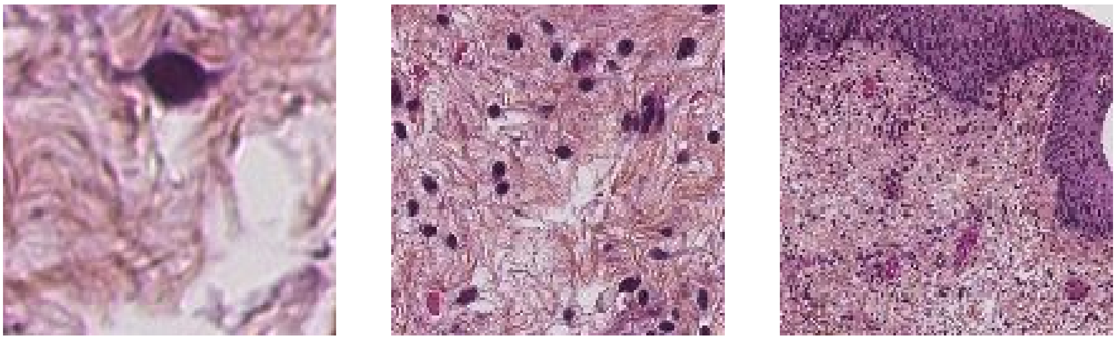
By taking a cluster-based approach, it is deemed more safe to include tiles that are of a lower probability, as it assumes that tiles closer to each other are more likely to hold the same label. Also, the method ensures that tiles are distributed more evenly across the WSIs in comparison to the probability-based self-training method. Contrarily, the segmentation images produces by the CBST based models are outperformed by the model duplication method, possibly because the probability criteria is set lower for CBST.

However, the model duplication method included more than twice as many patients as the models from CBST, and could also have an affect on the outcome.

It is difficult to compare CBST and PBST to non-expert annotations, even though the number of tiles are relatively similar in each dataset, and the same WSIs are included in all three methods. Throughout this thesis, numerous medical documents have been explored, and a significant increase is gained in tissue knowledge. The non-expert annotations took a good week to complete, and were reviewed by co-supervisor R. Wetteland afterwards. Nevertheless, pathologists have a medical education with years of practice, whos annotations are regarded as ground truth. That said, a significant increase was seen both in classification and segmentation by including non-expert annotation in the learning process.

7.1.2 Challenges with Multiscale

Most models trained during the course of this thesis are multiscale, including the three magnifications 25x, 100x and 400x. The is beneficial in that the respective model is able to capture both cell-level details, and context of nearby tissue. Nevertheless, this comes at a cost as annotations by pathologists are done at 400x magnification. In many cases, the tiles of lower magnification level (25x, 100x) will contain other types of tissue. In Figure 7.2, a tile from the ground truth test dataset, $D_{gt}\{test_1\}$, is presented, which is annotated as stroma. This tile was classified by model TRI-GT-SL-AF-1 as urothelium with a probability of approximately 99.99 %.



(a) 400x. **(b)** 100x. **(c)** 25x.

Figure 7.2: Stroma tile from ground truth dataset, classified by model TRI-GT-SL-AF-1 as urothelium with approximately 99.99 % probability.

The reason that the tile in Figure 7.2 gets classified as urothelium is probably caused by a combination of three reasons: a) there is urothelium in the outer regions of the 25x tile, b) there is a relatively high density of cell nuclei in the 100x tile, and even one in the 400x tile, and c) the training dataset $D_{gt}\{train_1\}$ consists of 3.49 % stroma and

24.26 % urothelium, i.e. the model has a much larger number of associated features for urothelium than for stroma. Another example of this is how several tiles of ground truth label blood are predicted as background in Fig. 6.2, as this area is rather isolated from nearby tissue.

7.1.3 Limitations of the Ground Truth Dataset

The ground truth dataset, D_{gt}, contains only about 8 % muscle cells, which yields an unbalanced dataset. This causes issues for the models trained on the dataset, as can be seen in that the F_1-scores is never able to go beyond 89 % for all models trained on the original dataset split, D_{gt}. This can, however, also partially be sourced in the new features presented in the damaged regions of WSI-015.

Several issues arise in a low number of stroma tissue tiles in the ground truth dataset. Since the initial model has little stroma tiles, it also has few associated features to this class. For a semi-supervised approach, this contribute to a negative effect: The original model initially lacks sufficient tiles to fully grasp the features of stroma. It will do more misclassifications in this class when predicting new weak labels. The semi-supervised model is then further trained on a number of these misclassified tiles. Intuitively, one would select more tiles from this class as it initially was lacking the most tiles. Adding in a large ratio of weak labels in comparison to ground truth labels during semi-supervised training makes the class more sensitive. On top of that, stroma contains smaller quantities of red blood cells causing confusion, and stroma had to be split between a train/test ratio of 74/26 for D_{gt} as one patient cannot exist in both training and test dataset. In total, this could very likely lead to an over-representation of false features associated with stroma in the finished trained SSL model.

7.1.4 Snowball Effect

All SSL approaches presented in this thesis appears to be facing issues with including a large portion of new tiles, in relation to how many tiles the model used for prediction had in its training dataset to begin with. Effectively, what happens from the original model trained in a supervised manner to the next generation model trained on the predicted tiles is that the weak labeled tiles probabilities are set to 100 %. In other words, tiles in the probability range 60 % to 100 % are added to the training dataset as to be tiles of probability 100 %. This causes a sort of snowball effect with regards to associated features, especially present for classes with a low number of associated features to begin with. This can be seen in Figure 7.3, where the leftmost image is trained strictly supervised, the middle image is trained with approximately 20 000 added tiles based on probability, and the rightmost image is with 100 000 predicted tiles based on probability. From left to right, there appears to be a significant increase in pink dots, i.e. stroma tiles, even though the ground truth annotations in this area hold no stroma regions whatsoever.

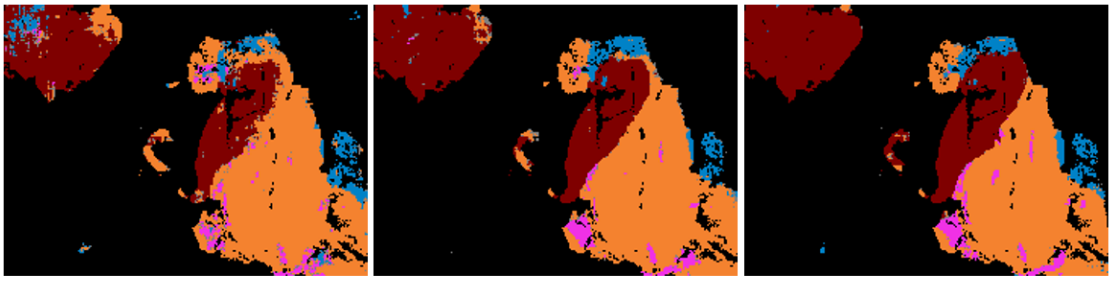


Figure 7.3: Predictions in WSI-084.
From left to right: TRI-SL-AF-1, TRI-P-SSL-AU-1, TRI-SSL-DM-U.
Color specifies predicted tile class: Blue = Damaged, Red = Blood, Black = Background, Orange = Urothelium.

Also, from left to right in Figure 7.3, the models appear to be improving at classifying the regions correctly for classes blood, damaged and urothelium. Intuitively, if a SSL process is repeated or larger and larger amounts of weak labeled tiles of a certain class are included, more and more misclassified tiles will be reintroduced as real features belonging to that class. The boundary between an area that the model is very certain about, and an area where the model is completely unsure, will shrink. This can cause feature-hungry classes, such as stroma in this case, to soak up new features quickly. On the other side, classes that are strongly represented in the initial ground truth dataset are good candidates for SSL.

7.1.5 Ground Truth vs. Non-expert Annotations

The model trained on both datasets $D_{gt}\{train_2\}$ and $D_{ne}\{train\}$ performed better than the model trained only on $D_{gt}\{train_2\}$, when both models were tested on test dataset $D_{gt}\{test_2\}$. Similarly, the model trained solely on $D_{ne}\{train\}$, and tested on $D_{gt}\{test_2\}$, shows that many of the same features exist in both datasets. This concludes that there are similar features in $D_{ne}\{train\}$ and $D_{gt}\{test_2\}$, and that the non-expert labels alone increased the performance with respect to F_1-score.

By analyzing the performance of the model trained only on ground truth dataset $D_{gt}\{train_2\}$, when tested on non-expert dataset $D_{ne}\{test\}$, it is observed that new features were introduced mainly in classes blood, i.e. where the F_1-score is lowest. However, new features were at different degrees introduced for all foreground classes. This also concludes that new features are introduced with the dataset D_{ne}, however, the degree of ground truth for those features will have to be decided by a pathologist.

7.1.6 Future work

As previously mentioned, WSI-015 contains some unique features that are not present in other damaged tissue in the ground truth dataset. It would be very beneficial to have at least one more WSI with a similar region annotated by the pathologist, so that these features could exist in both training and test dataset. The model duplication test indicate that similar features should exist among the 99 unlabeled WSIs that were included, as this model has a 99.93 % accuracy when tested on the entire ground truth dataset.

Throughout this thesis, it has gotten more and more apparent that the initial model builds the foundation for the next models trained in a semi-supervised manner. The initial dataset lacks tiles in both muscle and stroma, which consequently causes more tiles to be misclassified and introduced through SSL. For a semi-supervised approach to further benefit from these approaches, a larger variety in features for these two classes is needed.

For the probability-based self-training method, better distribution of tiles in the WSI was believed to be needed for this method to be improved. This was achieved by implementing linear spacing between tiles within a WSI for the model duplication experiment, which, for a multiscale model, will introduce new features across all magnification levels. For the cluster-based self-training method, several things can be considered for future work, like random selection of clusters within a WSI, selecting clusters more evenly spaced, or perhaps increase the criteria for classes that are poorly represented in the initial dataset.

7.2 Conclusion

The two SSL methods PBST and CBST both increased performance from an inital model trained only on ground truth labels with regards to both classification and segmentation. By far, a cluster-based approach outperforms a pure probability-based approach with regards to segmentation, which is further abbreviated in the submitted paper *Semi-supervised Tissue Segmentation of Histological Images* [83], abstract appended in Appendix B. The lack of labeled data makes both methods well suited to automatically increase the training data, and experiments conclude that augmentation is beneficial to some degree, and that unfrozen weights in the VGG16 model is preferred.

Different patients are very unique on cell-level, and become even more unique as the tissue sample is affected by different staining effect, color variation, ink, burn, etc. This sets the need for a large labeled dataset when working with medical images. Failing to do so sources issues with a unbalanced dataset with few features for some of the classes.

The gain from semi-supervised learning comes at a cost: new features will be introduced. If the initial dataset is large, then the higher the probability will be for these introduced features to be associated correctly.

Obviously, non-expert annotations cannot compare to ground truth annotations with regards to true class. That said, results presented in this thesis show that the method is able to introduce new features and simultaneously increase the performance of a model. It also shows that introducing non-expert annotations produced the most accurate segmentation maps for WSI.

List of Figures

List of Tables

Bibliography

[1] Kreftregisteret. BLÆREKREFT. URL https://www.kreftregisteret.no/Temasider/kreftformer/blarekreft/.

[2] Cancer Registry of Norway. Cancer in norway 2005. *Cancer incidence, mortality, survival and prevalence in Norway*, page 18, 2006. ISSN 0332-9631. URL https://www.kreftregisteret.no/globalassets/publikasjoner-og-rapporter/cin2005_del1_web.pdf.

[3] Cancer Registry of Norway. Cancer in norway 2010. *Cancer incidence, mortality, survival and prevalence in Norway*, page 26, 2012. ISSN 0332-9631. URL https://www.kreftregisteret.no/globalassets/cin_2010.pdf.

[4] Cancer Registry of Norway. Cancer in norway 2015. *Cancer incidence, mortality, survival and prevalence in Norway*, page 28, 2016. ISSN 0332-9631. URL https://www.kreftregisteret.no/globalassets/cancer-in-norway/2015/cin-2015.pdf.

[5] Cancer Registry of Norway. Cancer in norway 2018. *Cancer incidence, mortality, survival and prevalence in Norway*, page 20, 2019. ISSN 0806-3621. URL https://www.kreftregisteret.no/globalassets/cancer-in-norway/2018/cin2018.pdf.

[6] The Global Cancer Observatory, World Health Organization. Bladder, Source: Globocan 2018. URL https://gco.iarc.fr/today/data/factsheets/cancers/30-Bladder-fact-sheet.pdf. Last accessed 15.04.2020.

[7] R. Montironi A. Lopez-Beltran. Non-invasive urothelial neoplasms: According to the most recent who classification. *European Urology*, pages 170 – 176, Volume 46, Issue 2, 2004. doi:10.1016/j.eururo.2004.03.017.

[8] T. M. de Reijke A. Anastasiadis. Best practice in the treatment of nonmuscle invasive bladder cancer. 4Therapeutic Advances in Urology:13–32, 2012. doi:10.1177/1756287211431976.

[9] Stavanger Aftenblad. Pasienter må vente åtte uker på prøvesvar. 2020. URL https://www.aftenbladet.no/lokalt/i/Wk332/pasienter-ma-vente-atte-uker-pa-prvesvar.

[10] O.M. Mangrud, R. Waalen, E. Gudlaugsson et al. Reproducibility and Prognostic Value of WHO1973 and WHO2004 Grading Systems in TaT1 Urothelial Carcinoma of the Urinary Bladder. 12(6), 2014. doi:10.1371/journal.pone.0083192.

[11] The Economist. The world's most valuable resource is no longer oil, but data. *Leaders*, 2017. URL https://www.economist.com/leaders/2017/05/06/the-worlds-most-valuable-resource-is-no-longer-oil-but-data.

[12] A. S. Miller, B. H. Blott, T. K. Hames. Review of neural network applications in medical imaging and signal processing. *Med Biol Eng Comput.*, 30(5):449–464, 1992. doi:10.1007/BF02457822.

[13] G. Litjens, T. Kooi, B. E. Bejnordi et al. A survey on deep learning in medical image analysis. *Medical Image Analysis*, 42:60–88, 2017. doi:10.1016/j.media.2017.07.005.

[14] Z. Shi, L. He, K. Suzuki, T. Nakamura, H. Itoh. Survey on neural networks used for medical image processing. *International journal of computational science*, 3:86–100, 2009. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4699299/.

[15] C. Tao, H. Pan, Y. Li, Z. Zou. Unsupervised spectral–spatial feature learning with stacked sparse autoencoder for hyperspectral imagery classification. *IEEE Geoscience and Remote Sensing Letters*, 12(12):2438–2442, 2015. doi:10.1109/LGRS.2015.2482520.

[16] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010. doi:10.1109/TKDE.2009.191.

[17] Y. Song, C. Zhang, J. Lee et al. Semi-supervised discriminative classification with application to tumorous tissues segmentation of mr brain images. *Pattern Anal Applic*, 12:99–115, 2009. doi:10.1007/s10044-008-0104-3.

[18] V. Cheplygina, M. de Bruijne, J. P.W. Pluim. Not-so-supervised: A survey of semi-supervised, multi-instance, and transfer learning in medical image analysis. *Medical Image Analysis*, pages 280–297, 2019. ISSN 1361-8415. doi:10.1016/j.media.2019.03.009.

[19] O. J. Skrede et al. Deep learning for prediction of colorectal cancer outcome: a discovery and validation study. 395(10221):350–360, 2020. doi:10.1016/S0140-6736(19)32998-8.

[20] Stavanger Aftenblad. Ny norsk studie: På tre minutter stiller datamaskinen en mer presis kreftdiagnose enn legene. 2020. URL https://www.aftenbladet.no/innenriks/i/70w6Pv.

[21] V. Godbole et al. S. M. McKinney, M. Sieniek. International evaluation of an ai system for breast cancer screening. 577:89–94, 2020. doi:10.1038/s41586-019-1799-6.

[22] K. Dercksen, W. Bulten, G. Litjens. Dealing with label scarcity in computational pathology: A use case in prostate cancer classification. *Proceedings of Machine Learning Research – Accepted :1–4, 2019, Extended Abstract – MIDL 2019 submission*, 2019. URL https://arxiv.org/pdf/1905.06820.pdf.

[23] M. Peikari, S. Salama, S. Nofech-Mozes et al. A cluster-then-label semi-supervised learning approach for pathology image classification. *Scientific Reports*, (7193), 2018. doi:10.1038/s41598-018-24876-0.

[24] R. Wetteland. Classification of histological images of bladder cancer using deep learning. *Medical Image Analysis*, pages 1–77, 2017. URL http://hdl.handle.net/11250/2455555.

[25] R. Wetteland, K. Engan, T. Eftestøl, V. Kvikstad, E. A. M. Janssen. Multi-class tissue classification of whole-slide histological images using convolutional neural networks. *Proceedings of the 8th International Conference on Pattern Recognition Applications and Methods - Volume 1: ICPRAM*, pages 320–327, 2019. doi:10.5220/0007253603200327.

[26] R. Wetteland, K. Engan, T. Eftest, V. Kvikstad, E. A. M. Janssen. Multiscale Approach for Whole-Slide Image Segmentation of Five Tissue Classes in Urothelial Carcinoma Slides. *(in press)*. Accepted for publication in Journal of Technology in Cancer Research Treatment (TCRT) on 19 June 2020.

[27] Augusta University. Animal Tissues. URL https://www.augusta.edu/scimath/biology/docs/animaltissues.pdf. Last accessed 22.05.2020.

[28] Charlène Guillot and Thomas Lecuit. Mechanics of epithelial tissue homeostasis and morphogenesis. *Science*, 340(6137):1185–1189, 2013. ISSN 0036-8075. doi:10.1126/science.1235249.

[29] Wikimedia Commons. File:403 epithelial tissue.jpg. Last accessed 17.05.2020. URL https://commons.wikimedia.org/wiki/File:403_Epithelial_Tissue.jpg.

[30] BC Open Textbooks. 25 4.3 CONNECTIVE TISSUE SUPPORTS AND PROTECTS. URL https://opentextbc.ca/anatomyandphysiology/chapter/4-3-connective-tissue-supports-and-protects/. Last accessed 13.05.2020.

[31] Cooper GM. The cell: A molecular approach. 2nd edition. *Sunderland (MA): Sinauer Associates*, 2000. URL https://www.ncbi.nlm.nih.gov/books/NBK9961/.

[32] Knohl SJ Kaufman DP, Basit H. Physiology, Glomerular Filtration Rate (GFR). *StatPearls [Internet]*, 2020. URL https://www.ncbi.nlm.nih.gov/books/NBK500032/.

[33] Wikimedia Commons. File:Illu bladder hr.JPG. URL https://commons.wikimedia.org/wiki/File:Illu_bladder_hr.JPG. Last accessed 22.05.2020.

[34] YouTube user Osmosis. Anatomy and physiology of the kidneys, urinary bladder, ureters, urethra, and nephron. URL https://www.youtube.com/watch?v=805VoHIIQCs. Last accessed 27.05.2020.

[35] M. Burger et al. Epidemiology and risk factors of urothelial bladder cancer. *European Urology*, pages 234–241, 2013. ISSN 0302-2838. doi:10.1016/j.eururo.2012.07.033.

[36] M. Babjuk. Trends in bladder cancer incidence and mortality: Success or disappointment? *European Urology*, pages 109–110, Volume 71, Issue 1, 2017. doi:10.1016/j.eururo.2016.06.040.

[37] YouTube user Dr. Armando Hasudungan. Bladder Cancer - Overview (types, pathophysiology, diagnosis, treatment). URL https://www.youtube.com/watch?v=FtZNN5PNLlA. Last accessed 27.05.2020.

[38] P. E. Clark, J. P. Stein, S. G. Groshen et al. Radical cystectomy in the elderly. *Cancer*, 104(1):36–43, 2005. doi:10.1002/cncr.21126.

[39] Wikimedia Commons. File:Diagram showing early stage bladder cancer CRUK 442.svg. URL https://commons.wikimedia.org/wiki/File:Diagram_showing_early_stage_bladder_cancer_CRUK_442.svg. Last accessed 20.04.2020.

[40] Epstein J.I. Sesterhenn I.A. (Eds.) Eble J.N., Sauter G. World health organization classification of tumours. *Pathology and Genetics of Tumours of the Urinary System and Male Genital Organs.*, page 447-461, 2004. doi:10.1016/j.eururo.2016.05.041.

[41] Fathollah Keshvar Mostofi, Leslie H Sobin, Humberto Torloni, and World Health Organization. Histological typing of urinary bladder tumours / f. k. mostofi, in collaboration with l. h. sobin, h. torloni and pathologists in fourteen countries. *International histological classification of tumours ; no. 10*, page 36 p., 1973.

[42] Burger M et al. Babjuk M, Böhle A. Eau guidelines on non-muscle-invasive urothelial carcinoma of the bladder: Update 2016. *European Urology*, 71:447-461, 2017. doi:10.1016/j.eururo.2016.05.041.

[43] V. Kvikstad, O. M. Mangrud, E. Gudlaugsson et. al. Prognostic value and reproducibility of different microscopic characteristics in the who grading systems for pta and pt1 urinary bladder urothelial carcinomas. *Diagnostic Pathology*, 14(90), 2019. doi:10.1186/s13000-019-0868-3.

[44] Marcello Malpighi and Fundador de la Microanatomía. Marcello malpighi (1628-1694), founder of microanatomy. *Int. J. Morphol*, 29 (2):399–402, 2011. URL https://www.researchgate.net/profile/Rafael_Romero_Reveron/publication/262597000_Marcello_Malpighi_1628-1694_Fundador_de_la_Microanatomia/links/0e0995608acca2f21b374028/Marcello-Malpighi-1628-1694-Fundador-de-la-Microanatomia.pdf.

[45] A. H. Fischer, K. A. Jacobson, J. Rose, R. Zeller. Hematoxylin and eosin staining of tissue and cell sections. 2008. doi:10.1101/pdb.prot4986.

[46] L. Gröntoft E. Edston. Saffron—a connective tissue counterstain in routine pathology. *Journal of Histotechnology*, 20:123–125, 1997. doi:10.1179/his.1997.20.2.123.

[47] M. T. Shaban, C. Baur, N. Navab, and S. Albarqouni. Staingan: Stain style transfer for digital histological images. *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, pages 953–956, 2019. ISSN 1945-8452. doi:10.1109/ISBI.2019.8759152.

[48] J. Cupitt et al. pyvips. *GitHub repository*, 2017. URL https://github.com/libvips/pyvips.

[49] W. Pitts W. S. McCulloch. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115–133, 1943. doi:10.1007/BF02478259.

[50] D. Hebb. A logical calculus of the ideas immanent in nervous activity. *ICML 2012*, 1949. URL https://archive.org/details/in.ernet.dli.2015.226341/page/n1/mode/2up.

[51] Pace University, Seidenberg School of CSIS. The Man who forever changed Artificial Intelligence. URL http://csis.pace.edu/~ctappert/srd2011/rosenblatt-contributions.htm. Last accessed 28.05.2020.

[52] A. Griewank, Documenta Math, Extra Volume ISMP (2012) 389–400. Seppo Linnainmaa. URL https://www.math.uni-bielefeld.de/documenta/vol-ismp/52_griewank-andreas-b.pdf. Last accessed 28.05.2020.

[53] Youtube user Sebastian Schuchmann. The Year Artificial Intelligence changed forever. URL https://www.youtube.com/watch?v=yRUUDJfDarU. Last accessed 28.05.2020.

[54] T. S. Huang J. Weng, N. Ahuja. Cresceptron: a self-organizing neural network which grows adaptively. *Proc. International Joint Conference on Neural Networks*, 1:576–581, 1992. doi:10.1109/IJCNN.1992.287150.

[55] A. Y. Ng et al. Building high-level features using large scale unsupervised learning. 2012. URL https://icml.cc/2012/papers/73.pdf.

[56] Wikipedia. Artificial neural network, section History, . URL https://en.wikipedia.org/wiki/Artificial_neural_network#History. Last accessed 28.05.2020.

[57] Investor Place. 10 Companies Using AI to Grow. URL https://investorplace.com/2019/08/10-companies-using-ai-to-grow/. Last accessed 28.05.2020.

[58] The Verge, A. J. Hawkins. It's Elon Musk vs. everyone else in the race for fully driverless cars. URL https://www.theverge.com/2019/4/24/18512580/elon-musk-tesla-driverless-cars-lidar-simulation-waymo. Last accessed 28.05.2020.

[59] Youtube user Google. TensorFlow: Open source machine learning. URL https://www.youtube.com/watch?v=oZikw5k_2FM. Last accessed 28.05.2020.

[60] M. Beale M. T. Hagan, H. B. Demuth. Neural network design. 1995. URL https://hagan.okstate.edu/NNDesign.pdf.

[61] Y. Ito. Representation of functions by superpositions of a step or sigmoid function and their applications to neural network theory. *Neural Networks*, 4:385–394, 1991. doi:10.1016/0893-6080(91)90075-G.

[62] T. N. Sainath G. E. Dahl and G. E. Hinton. Improving deep neural networks for lvcsr using rectified linear units and dropout. *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8609–8613, 2013. ISSN 2379-190X. doi:10.1109/ICASSP.2013.6639346.

[63] Dataaspirant, S. Polamuri. DIFFERENCE BETWEEN SOFTMAX FUNCTION AND SIGMOID FUNCTION. URL https://dataaspirant.com/2017/03/07/difference-between-softmax-function-and-sigmoid-function/. Last accessed 01.06.2020.

[64] Towards Data Science. Common Loss functions in machine learning, . URL https://towardsdatascience.com/common-loss-functions-in-machine-learning-46af0ffc4d23. Last accessed 03.06.2020.

[65] Towards Data Science. Understanding the Mathematics behind Gradient Descent, . URL https://towardsdatascience.com/understanding-the-mathematics-behind-gradient-descent-dde5dc9be06e. Last accessed 03.06.2020.

[66] R. Hecht-Nielsen. III.3 - Theory of the Backpropagation Neural Network. *Proceedings of the International Joint Conference on Neural Networks 1*, 1:65–93, 1992. doi:10.1016/B978-0-12-741252-8.50010-8.

[67] L. Bottoue. Stochastic gradient descent tricks. *Proceedings of the International Joint Conference on Neural Networks 1*, pages 421–436, 2012. doi:10.1007/978-3-642-35289-8_25.

[68] Keras. Keras faq. *Last accesses: 23.06.20*. URL https://keras.io/getting_started/faq/.

[69] Towards Data Science. A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way, . URL https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53. Last accessed 03.06.2020.

[70] Jason Brownlee. A gentle introduction to pooling layers for convolutional neural networks. 2019. URL https://machinelearningmastery.com/pooling-layers-for-convolutional-neural-networks/.

[71] Jason Brownlee. A gentle introduction to dropout for regularizing deep neural networks. 2019. URL https://machinelearningmastery.com/dropout-for-regularizing-deep-neural-networks/.

[72] TensorFlow. tf.keras.layers.dropout. 2020. URL https://www.tensorflow.org/api_docs/python/tf/keras/layers/Dropout.

[73] Machine Learning Mastery. A Gentle Introduction to Transfer Learning for Deep Learning, . URL https://machinelearningmastery.com/transfer-learning-for-deep-learning/. Last accessed 03.06.2020.

[74] Machine Learning Mastery. Supervised and Unsupervised Machine Learning Algorithms, . URL https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/. Last accessed 03.06.2020.

[75] Wikipedia. Ground truth, . URL https://en.wikipedia.org/wiki/Ground_truth. Last accessed 03.06.2020.

[76] Wikipedia. Weak supervision, Section "Types of weak labels", . URL https://en.wikipedia.org/wiki/Weak_supervision#Types_of_weak_labels. Last accessed 03.06.2020.

[77] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[78] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei. Imagenet: A large-scale hierarchical image database. *CVPR09*, 2009. URL http://www.image-net.org/papers/imagenet_cvpr09.bib.

[79] M. Abadi et al. Tensorflow: Large-scale machine learning on heterogeneous systems. 2015. URL https://www.tensorflow.org/. Software available from tensorflow.org.

[80] F. Chollet et al. keras. *GitHub repository*, 2015. URL https://github.com/keras-team/keras.

[81] F. Pedregosa et al. A survey on transfer learning. *Scikit-learn: Machine Learning in Python*, 22(10):1345–1359, 2010. URL http://jmlr.org/papers/v12/pedregosa11a.html.

[82] G. van Rossum et al. Pickle. *GitHub repository*, 1990. URL https://github.com/python/cpython/blob/3.8/Lib/pickle.py.

[83] O.N. Dalheim, R. Wetteland, K. Engan, V. Kvikstad, E. A. M. Janssen. Semi-supervised Tissue Segmentation of Histological Images. *(submitted)*. Submitted for publication in Colour and Visual Computing Symposium on 18 June 2020.

Detailed overview of tiles per WSI

Manually Marked Ground Truth Dataset

Table A.1: Ground truth labels that make up the dataset D_{gt}, and corresponding tiles. Size is indicated in pixels [height x width].

WSI	Size	Ba	Bl	Da	Mu	St	Ur
WSI-001	43 024 x 33 552						90
WSI-002	87 696 x 90 896			2 696			627
WSI-003	62 480 x 32 016						688
WSI-004	93 712 x 133 520		597	3 750		215	5 076
WSI-005	79 248 x 96 144		4 052				
WSI-006	62 480 x 48 784			3 507			224
WSI-007	81 424 x 71 952			1 612		1 098	676
WSI-008	81 680 x 117 648			3 770			737
WSI-009	86 672 x 129 808	5 589					
WSI-010	69 904 x 80 912	1 261					
WSI-011	89 872 x 106 640						1 790
WSI-012	88 848 x 141 840						1 231
WSI-013	34 576 x 51 856	1 677					
WSI-014	74 128 x 109 584	4 206					
WSI-015	53 008 x 55 056			1 648			1 943
WSI-016	61 840 x 73 232			10 320			777
WSI-017	88 720 x 104 592	4 817					
WSI-018	70 032 x 117 648				1 905	1 261	675
WSI-019	85 136 x 121 744				2 289		1 207
WSI-020	87 696 x 131 216						1 189
WSI-021	83 216 x 125 712						674
WSI-022	81 424 x 82 448					516	988
WSI-023	84 240 x 119 312						1 061
WSI-024	90 384 x 127 632						546
WSI-025	80 400 x 69 776						198
WSI-026	84 624 x 89 872						425
WSI-027	43 024 x 41 872				3 138		2 695
WSI-028	61 456 x 69 776	2 556		4 461	2 634	1 785	601
WSI-029	82 448 x 121 360						1 010

Continued on next page

Table A.1 – continued from previous page

WSI	Size	Ba	Bl	Da	Mu	St	Ur
WSI-030	69 776 x 85 648	6 906		1 843			
WSI-031	64 528 x 67 728						355
WSI-032	62 480 x 76 688						564
WSI-033	79 248 x 128 784						659
WSI-034	88 720 x 101 392		12 300				1 938
WSI-035	88 720 x 135 056		2 166				
WSI-036	74 128 x 97 424		717				
WSI-037	80 400 x 101 392						1 084
Tot. tiles	all = 125 020	27 012	19 832	33 607	9 966	4 875	29 728
% of all		21.6%	15.9%	26.9%	8%	3.9%	23.8%
WSIs	39	7	5	9	4	5	28
tiles/WSI	3 205	3 859	3 966	3 734	2 491	975	1 062

Probability-weak Dataset

Table A.2: Probability-weak labels that make up the dataset, D_{pw}, and corresponding tiles. Size is indicated in pixels [height x width].

WSI	Size	Ba	Bl	Da	Mu	St	Ur
WSI-138	45456 x 32784	254		23			480
WSI-139	50960 x 67728	1049	2	710		3	7
WSI-140	91024 x 146192	43		120		144	480
WSI-141	55952 x 83472	33	1530	710	4582	1853	480
WSI-142	88720 x 117776	211	429	710	46	248	480
WSI-143	16912 x 16656	205				19	380
WSI-144	74128 x 125328	25	395	710	2	5	480
WSI-145	41360 x 28816	179		8	131	261	480
WSI-146	38928 x 24848	911		2	343	385	480
WSI-147	37392 x 33040	291		74		6	480
WSI-148	88336 x 109584	1591	11	710	416	2305	480
WSI-149	37392 x 36880	223	414	370	1	22	480
WSI-150	49552 x 49040	142		41			480
WSI-151	74128 x 93456	2	2	710	7218	4621	480
WSI-152	78992 x 102416	742	7900	710		38	480
WSI-153	67728 x 59664	1916		710	768	145	480
WSI-154	92304 x 109840	1479	3	710	521	1001	27
WSI-155	72592 x 83216	113		710	177	300	480
WSI-156	80272 x 62864	53		710	277	445	480
WSI-157	82320 x 93328	42	1	710	283	212	480
WSI-158	92048 x 137872	1393	1	710	11	61	480
WSI-159	91024 x 94992	78		710	712	215	480
WSI-160	53648 x 73232	12		33		58	480
WSI-161	65552 x 58640	321		440	1	489	480

Continued on next page

Table A.2 – continued from previous page

WSI	Size	Ba	Bl	Da	Mu	St	Ur
WSI-162	76048 x 99344	1854		710		46	480
WSI-163	87440 x 102928	556	738	710		3	480
WSI-164	65936 x 69136	4		56	16	1824	480
WSI-165	29072 x 20752	261		85		4	28
WSI-166	87952 x 89360	633		710	306	19	480
WSI-167	82320 x 101520	33	13	710	1607	899	480
WSI-168	53264 x 51088	827	9	13		66	480
WSI-169	83728 x 126608	1303	922	372	8	193	480
WSI-170	41360 x 67216	161				3	480
WSI-171	91024 x 94224	553		710	22	53	480
WSI-172	61456 x 148240	548	7590	710	1	210	480
WSI-173	33168 x 16656	55				21	149
WSI-174	51600 x 63632	515		710	678	22	480
WSI-175	70032 x 89360	32	50	710	383	1111	480
WSI-176	93200 x 121744	587		710	1877	581	480
WSI-177	61840 x 121744	33		710		204	480
WSI-178	64528 x 68112	8		107	2	67	480
WSI-179	85776 x 121744	284	26	710	4	1451	480
WSI-180	33296 x 117648	1		17			480
WSI-181	21008 x 24848	86				10	291
WSI-182	75664 x 107280	513		289	23	606	480
WSI-183	33168 x 36880	145		376			480
Tot. tiles	all = 121 239	20 300	20 036	20 176	20 416	20 229	20 082
% of all		16.7%	16.5%	16.6%	16.8%	16.7%	16.5%
WSIs	46	46	18	42	28	42	46
tiles/WSI	2 636	441	1 113	480	729	481	437

Cluster-weak Dataset

Table A.3: Cluster-weak labels that make up the dataset, D_{cw}, and corresponding tiles. Size is indicated in pixels [height x width].

WSI	Size	Ba	Bl	Da	Mu	St	Ur
WSI-138	45456 x 32784	3 47		643		236	1 193
WSI-139	50960 x 67728	1 311	152	800		643	1 236
WSI-140	91024 x 146192	798	1 440				1 200
WSI-141	55952 x 83472		4 027	795	4 747	1 430	1 225
WSI-142	88720 x 117776	202	2 986	780	787	1 435	1 242
WSI-143	16912 x 16656	63				1 440	1 078
WSI-144	74128 x 125328		1 509	800	95	1 431	1 196
WSI-145	41360 x 28816	233		237	1 420	1 348	1 220
WSI-146	38928 x 24848	629	20	201	1 592	1 327	1 190
WSI-147	37392 x 33040	65		791		299	1 227

Continued on next page

Table A.3 – continued from previous page

WSI	Size	Ba	Bl	Da	Mu	St	Ur
WSI-148	88336 x 109584	1 639	351	784	4 830	1 408	1 200
WSI-149	37392 x 36880	56	2 485	799	81	1 387	1 239
WSI-150	49552 x 49040	124		336			1 230
WSI-151	74128 x 93456		243	774	4 836	1 452	1 215
WSI-153	67728 x 59664	2 655		810	2 903	1 445	1 230
WSI-154	92304 x 109840	1 001	98	800	4 321	1 456	1 218
WSI-155	72592 x 83216	56		810	1 521	1 438	1 232
WSI-156	80272 x 62864	314		782	734	1 440	1 260
WSI-157	82320 x 93328		21	800	1 361	1 443	1 218
WSI-158	92048 x 137872	1313	190	800	195	1 367	1 200
WSI-159	91024 x 94992	599		800	2 885	1 442	1 232
WSI-160	53648 x 73232			796		1 332	1 236
WSI-161	65552 x 58640	274		806	156	1 439	1 225
WSI-162	76048 x 99344	2 476		800		1 440	1 248
WSI-163	87440 x 102928	267	3 224		752	759	1 224
WSI-164	65936 x 69136		36	795	561	1 450	
WSI-165	29072 x 20752	389		796		865	
WSI-166	87952 x 89360	328		800	1 155		517
WSI-167	82320 x 101520		142	770	4 797	1 435	
WSI-168	53264 x 51088	892	493	784		1 440	
WSI-169	83728 x 126608	1 198	2 892	816	51	1 388	
WSI-170	41360 x 67216	136		104		200	
WSI-171	91024 x 94224	249	20	774	103	1 444	
WSI-172	61456 x 148240	354	19 105	806		1 443	
WSI-173	33168 x 16656	54		238		1 440	
WSI-174	51600 x 63632	525		798	1 940	823	
WSI-175	70032 x 89360		349	800	1 229	1 440	
WSI-176	93200 x 121744		20	792	4 806	1 431	
WSI-177	61840 x 121744	792			20	1 428	
WSI-178	64528 x 68112				60	1 449	
WSI-179	85776 x 121744	153	295		150	1 430	
WSI-181	21008 x 24848	107				201	
WSI-182	75664 x 107280	1 625	47		271	1 450	
WSI-183	33168 x 36880	67				500	
Tot. tiles	all = 221 612	21 281	42 630	24 817	48 359	52 794	31 731
% of all		9.6%	19.2%	11.2%	21.8%	23.8%	14.3%
WSIs	44	34	23	35	29	41	26
tiles/WSI	5 037	626	1 854	709	1 668	1 288	1 220

Non-expert Dataset

Table A.4: Non-expert labels that make up the dataset, D_{ne}, and corresponding tiles. Size is indicated in pixels [height x width].

WSI	Size	Bl	Da	Mu	St	Ur
WSI-138	45456 x 32784	3	404		11	497
WSI-140	91024 x 146192	63	514		30	395
WSI-141	55952 x 83472	708	262	429	2 155	226
WSI-142	88720 x 117776	1 614	941		348	4 125
WSI-143	16912 x 16656				92	91
WSI-144	74128 x 125328	598			145	1 283
WSI-145	41360 x 28816			215	527	636
WSI-146	38928 x 24848	8		186	1 438	318
WSI-147	37392 x 33040	20	239			379
WSI-148	88336 x 109584	104	286	1 979	1 577	705
WSI-149	37392 x 36880	2 750	164		656	930
WSI-150	49552 x 49040		197			1 415
WSI-152	74128 x 93456	405	130	1 514	574	359
WSI-153	67728 x 59664	28	1 323	439	309	1 498
WSI-154	92304 x 109840		791		185	
WSI-155	72592 x 83216	435	566	282	496	72
WSI-156	80272 x 62864	43	1 403	200	2 678	443
WSI-157	82320 x 93328	105	483	851	175	205
WSI-158	92048 x 137872	145	3 146	187	867	1 040
WSI-159	91024 x 94992	76	676	2 725	724	965
WSI-160	53648 x 73232	45	76		1 003	1 117
WSI-161	65552 x 58640	304	607	117	484	794
WSI-162	76048 x 99344	65	2 987	11	157	573
WSI-163	87440 x 102928	1 193	323	79	655	766
WSI-164	65936 x 69136	69	234	103	1 003	656
WSI-165	29072 x 20752	15	318		584	
WSI-166	87952 x 89360	22	1 693	372	88	292
WSI-167	82320 x 101520	120	1 265	1 892	949	178
WSI-168	53264 x 51088	395			71	212
WSI-169	83728 x 126608	2 183	1 124		48	671
WSI-170	41360 x 67216	29	19			213
WSI-171	91024 x 94224	11	950			460
WSI-172	61456 x 148240	5 304	625	28	296	442
WSI-173	33168 x 16656			10	1 607	91
WSI-174	51600 x 63632	3	697	1 662	253	650
WSI-175	70032 x 89360	320	800	627	1 152	786
WSI-176	93200 x 121744	227	326	720	507	1 031
WSI-177	61840 x 121744	75	333	180	663	624
WSI-178	64528 x 68112	10	141		436	2 450
WSI-179	85776 x 121744	230	537	11	553	520
WSI-180	33296 x 117648					1 062
WSI-182	75664 x 107280	174	420	323	749	3 399
WSI-183	33168 x 36880		134			412
Tot. tiles	all = 115 401	17 899	25 134	15 142	24 245	32 981
% of all		15.5%	21.8%	13.1%	21.0%	28.6%

Continued on next page

Table A.4 – continued from previous page

WSI	**Size**	**Bl**	**Da**	**Mu**	**St**	**Ur**
Patients	43	36	36	25	37	41
tile/pat.	2 683	497	698	606	655	804

Semi-supervised Tissue Segmentation of Histological Images

Semi-supervised Tissue Segmentation of Histological Images

Ove Nicolai Dalheim[1][0000−0002−4822−7920], Rune Wetteland[1][0000−0002−9995−4204], Vebjørn Kvikstad[2,3], Emiel A.M. Janssen[2,3], and Kjersti Engan[1][0000−0002−8970−0067]

[1] Department of Electrical Engineering and Computer Science, University of Stavanger, Norway
https://www.uis.no/tn/ide/
[2] Department of Pathology, Stavanger University Hospital, Norway
[3] Department of Chemistry, Bioscience and Environmental Engineering, University of Stavanger, Norway
https://www.uis.no/faculty-of-science-and-technology/chemistry-bioscience-and-environmental-engineering/
ove.nicolai@dalheim.as
{rune.wetteland,kjersti.engan}@uis.no
{vebjorn.kvikstad, emilius.adrianus.maria.janssen}@sus.no

Abstract. Supervised learning of convolutional neural networks (CNN) used for image classification and segmentation has produced state-of-the-art results, including in many medical image applications. In the medical field, making ground truth labels would typically require an expert opinion, and a common problem is the lack of labeled data. Consequently, the models might not be general enough. Digitized histological microscopy images of tissue biopsies are very large, and detailed truth markings for tissue-type segmentation is scares or non-existing. However, in many cases, large amounts of unlabeled data that could be exploited are readily accessible. Methods for semi-supervised learning exists, but it is hardly explored in the context of computational pathology. This paper deals with semi-supervised learning on the application of tissue-type classification in histological whole-slide images of urinary bladder cancer. Two semi-supervised approaches utilizing the unlabeled data in combination with a small set of labeled data is presented. A multiscale, tile-based segmentation technique is used to classify tissue into six different classes by the use of three individual CNNs. Each CNN is presented tissue at different magnification levels in order to detect different feature types, later fused in a fully-connected neural network. The two self-training approaches are: using probabilities and using a clustering technique. The clustering method performed best and increased the allover accuracy of the tissue tile classification model from 94.6% to 96% compared to using supervised learning with labeled data. In addition, the clustering method generated visually better segmentation images.

Keywords: CNN · semi-supervised learning · bladder cancer · histological images · tissue segmentation