



Universitetet
i Stavanger

FACULTY OF SCIENCE AND TECHNOLOGY

MASTER'S THESIS

Study programme/specialisation:
Information technology -
Automation and Signal Processing

Spring semester, 2019

Open/~~Confidential~~

Author: Fredrik Orrebakken Svendsen

.....*F. Svendsen*.....
(signature of author)

Program coordinator: Professor Kjersti Engan (UiS)

Supervisor(s): Professor Kjersti Engan (UiS), Emiel Janssen (SUS)

Title of master's thesis:

Image Processing and Deep Neural Networks for Detection of Immune Cells on Histological Images of Bladder Cancer

Credits: 30

Keywords: Bladder Cancer, Classical Image Processing, Deep Neural Network, Weak Labeled Data.

Number of pages: 64
+ supplemental material/other: 3 +
embedded files

Stavanger, 15th of June 2019
date/year

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

Image Processing and Deep Neural Networks for Detection of Immune Cells on Histological Images of Bladder Cancer

Master's Thesis in Information Technology - Automation and Signal processing
by

Fredrik Orrebakken Svendsen

Internal Supervisors

Kjersti Engan

External Supervisors

Emiel Janssen

June 2019

Abstract

Bladder cancer is the tenth most common cancer type, where urothelial carcinoma is the most common type of bladder cancer. Bladder cancer has been classified as the most expensive type of cancer per patient, as the need for post-treatment monitoring often lasts the rest of the patient's life. A pathologist needs to diagnose and evaluate the risk of progression and relapse from analyzing histological images.

Recent research shows a correlation between the number of regulatory T-cells and which patients that get progression to a higher cancer grade. Today a computer randomly picks out a sub-set of cells, that is to be manually counted and classified; this will serve as an estimation for regulatory T-cells compared to other cells. This paper proposes a more automated solution to aid in analyzing histological images for the number of regulatory T-cells and other cells present.

The two proposed systems are using classical image processing to find and classify the cells based on color and using a convolutional neural network to detect and classify smaller parts of the images. Both systems will attempt to estimate the number of regulatory T-cells compared to other cells.

The classical image processing had an underestimation of 4.7% for regulatory T-cells while having a 4.5% overestimation of other cells. The convolutional neural network showed a correlation between the number of classifications and the actual amount of cells but requires further work to be usable.

Preface

This thesis marks the end of the Master of Science degree at the University of Stavanger, Department of Electrical Engineering and Computer Science. The thesis has proven to be both challenging and educational in the technical aspect, and have broadened my knowledge in the medical field.

I want to thank my supervisor professor Kjersti Engan, for her enthusiasm, excellent guidance, and encouragement throughout the thesis.

I would also like to thank Emiel Janssen and Melinda Lillesand for sharing their medical knowledge. I would also like to give an extra thanks to Melinda Lillesand and Vebjørn Kvikstad for making time to sit down with me and label some data.

Contents

Abstract	ii
Acknowledgements	iii
Abbreviations	vi
List of Figures	vii
List of Tables	ix
1 Introduction	1
1.1 Motivation	1
1.2 Previous Work	3
1.3 Thesis Objective	3
1.4 Thesis Outline	3
2 Background Theory	5
2.1 Medical Background	5
2.1.1 Urothelial Carcinoma	5
2.1.2 CD25+ Staining	7
2.2 Image Processing	10
2.2.1 Morphology	11
2.2.2 Image Histogram	14
2.2.3 Thresholding	15
2.2.4 Color Format	16
2.3 Neural Networks	17
2.3.1 Activation Function	18
2.3.2 Fully Connected	19
2.3.3 Learning	20
2.3.4 Pooling Layers	20
2.3.5 Structure	21
2.3.6 Convolutional Neural Networks	21
2.3.7 Transfer Learning	22
2.3.8 Performance Validation	22

3	Data Material	25
3.1	Data Set	25
4	Unlabeled Data Approach	27
4.1	Methods	27
4.1.1	Methods	28
4.1.2	Pre-Processing Whole Slide Images	29
4.1.3	Interesting Areas	31
4.1.4	Classical image processing	32
4.1.5	Deep Learning with Weak Labeled Data	37
4.1.6	U-net	37
4.1.7	Implementation	39
4.2	Experiments and Results	40
4.2.1	Pre-Processing	40
4.2.2	Finding Region of Interest	40
4.2.3	Cell center	41
4.2.4	Classification	43
4.2.5	U-net	43
5	Labeled Data Approach	44
5.1	Methods	44
5.1.1	Classical Image Processing	46
5.1.2	Implementation	48
5.2	Experiments and Results	49
5.2.1	VGG-16 Trained on Weak Labels	49
5.2.2	Classical Image Processing Methods	51
6	Discussion	55
6.1	Approach	55
6.1.1	Classical Image Processing	55
6.1.2	Convolutional Neural Network	56
6.2	Analyzing the Results	57
7	Conclusion and Future Work	58
7.1	Conclusion	58
7.2	Future Work	59
	Bibliography	64
	A Supplementary Material	65
	B Table of Results	66
	B.1 Image Processing	67
	B.2 VGG-16	68
	C Images Used in Tests	70

Abbreviations

IARC	I nternational A gency R easearch C ancer
WSI	W hole- S lide I mage
BC	B ladder C ancer
NIMBC	N on- M uscle I nvasive B ladder C ancer
TURB	T rans U rethral R esection of the B ladder
HE	H ematoxylin and E osin
TNM	T umour N ode M etasis
WHO	W orld H ealth O rganization
NN	N eural N etwork
DNN	D eep N eural N etwork
PUNPLM	P apillary U rothelial N eoplasm of L ow M alignant P otential
TIFF	T ag I mage F ile F ormat
HSV	H ue S aturation V alue
RGB	R ed G reen B lue
TP	T rue P ositive
TN	T rue N egative
FP	F alse P ositive
FN	F alse N egative
CPU	C entral P rocessing U nit
GPU	G raphics P rocessing U nit
TURB	T rans- U rethral R esection and B iopsy
BCG	B acillus C almette G uérin
SUS	S tavanger U niversitets S jukehus
ReLU	R ectified L inear u nit

List of Figures

2.1	Illustration of the bladder with different the tumor stages of BC	6
2.2	WSI marked with CD25 immuno stain	8
2.3	WSI marked with HE stain	8
2.4	Pyramid image visualization	9
2.5	Morphological erosion example	12
2.6	Morphological dilation example	12
2.7	Numerical visualization of dilation	13
2.8	Overlapping circles	14
2.9	Euclidean distance transform	14
2.10	Rectilinear distance transform	14
2.11	Histological image	15
2.12	Histogram of cell image	15
2.13	HSV	17
2.14	Biological neuron	18
2.15	NN with fully connected layers	19
2.16	Max pooling	21
2.17	Typical CNN structure	22
2.18	Confusion matrix	24
3.1	Labelbox labeled image	26
4.1	Process overview for unlabeled approach	29
4.2	Pre-processing overview	30
4.3	Process overview for finding a region of interest	31
4.4	Step by step process for classical image processing used	33
4.5	Original image thresholded with and without smoothing	34
4.6	Overview of unlabeled approach to segmentation network	37
4.7	U-net architecture	38
4.8	Highlighted areas	41
4.9	Marked images with classical image processing	42
5.1	Overview of the proposed system	45
5.2	Modified classical image processing	46
5.3	Overview of VGG-16 steps	47
5.4	512 by 512 image evenly split into different sized tiles	48
5.5	Performance of VGG-16 model with 32x32 tiles	50
5.6	Performance of VGG-16 model with 64x64 tiles	50
5.7	Estimation of non-positive CD25+ markings using VGG-16	51

5.8	Estimation of positive CD25+ markings using VGG-16	51
5.9	Non-positive CD25+ marked cells, classified with image processing	52
5.10	Positive CD25+ marked cells, classified with image processing	53
5.11	Amount of TP of non-positive CD25+ markings, using Classical image processing	53
5.12	Amount of TP of positive CD25+ markings, using Classical image processing	54
5.13	Confusion matrix for image processing	54

List of Tables

2.1	Overview of terms used in evaluation performance	23
4.1	Implementation for unlabeled data approach	39
4.2	Number of images saved and deleted for a subset of patients, as well as time used on each WSI.	40
4.3	Manually counted cells by the author of this thesis, compared to the cell count found with image processing	42
5.1	Implementation of used methods in the labeled data approach	49
B.1	Results from classical image processing for localization and classification .	67
B.2	Result from VGG-16 network trained on 64x64 tiles	68
B.3	Result from VGG-16 network trained on 32x32 tiles	69

Chapter 1

Introduction

This chapter consists of the motivation for this thesis, related and previous work, the objective of this thesis and lastly the outline of the entire thesis.

1.1 Motivation

According to International Agency for Research on Cancer (IARC), there were estimated 549 393 new cases of bladder cancer (BC) worldwide in 2018, placing it at the tenth most common cancer type for both sexes combined [1]. In the same year, the estimated deaths caused by BC, for both genders was just below 200 000. Statistically, males have a four times higher likelihood of getting BC compared to women. In 2018 it was the sixth most common cancer type for males with 424 082 new cases [1]. Urothelial carcinoma makes up approximately 90% of BC, making it the most common type of BC [2]. By comparing incidence data in Norway from 2008 to the incidence estimate for 2018, it can be seen that the number of incidences has not been reduced [1, 3]. The last four decades indicate that there has been an increase of 70% in BC incidences for Norwegian men and similar 40% for women [4]. A study found that statically one in 40 men and one in 130 females will get BC [4]

Since the recurrence rate of urothelial BC is relatively high, the patient needs to be regularly monitored to check for recurrence or progression. When BC is diagnosed in Norway, the average age group is 70-75 and is likely to be with them for the rest of their lives [3]. BC is considered the overall most expensive cancer type per patient, because of the extensive post-treatment monitoring [5].

The monitoring process can be particularly strenuous on the patient, because of the age. Having a cancer type that needs to be monitored for the rest of their lives is also a heavy psychological burden and can affect the quality of life [6].

Today the tissue images, called histological images, are a valuable asset for pathologists to diagnose different types of cancer and diseases[7]. These histological images are the digitization form for studying the tissue under microscopes and is called whole-slide images (WSI). The tissue is derived from a thin slice from a biopsy and is placed on a glass plate for scanning. Different biomarkers can be added to the tissue, to enhance different cells in the tissue.

The digitization of the microscopic images have opened up for several opportunities, such as WSI computer analysis. This led to the computer assisted diagnosis (CAD) tools, which can assist with making faster and more objective diagnoses [8]. Since the recurrence and the progression rate in BC are still high, and constant monitoring is needed to ensure the patients best welfare, it is clearly a better solution is desired. A study indicates that the use of different biomarkers on the WSI can help predict the progression of BC [9]. This is done with the different immune stains, which are called biomarkers. These will attach themselves to a certain type of cell and make it easier to find different cells, for example, different types of immune cells. One immune stain, CD25+ stood especially out, as it had a sufficient discriminative ability to predict progression in non-muscle-invasive bladder cancer (NMIBC) [9]. This was found from a counting estimate of cells marked with the biomarker in different patients and compared with those who had a recurrence or progression [9]. Being able to better predict progression and recurrence with an automated system based on different biomarkers can reduce the diagnose time and total monitoring needed for each patient.

Currently, the diagnosis and analyzing of WSI are mostly done manually with the assistance of a program, this process is slow and costly, as well as the accuracy can vary with the expertise of each pathologist. With the improvements in image processing and deep neural networks (DNN) within the classification aspect, a better system can be made for analyzing the tissue images. A possible fully or semi-automated system can be faster, cheaper, more accurate in diagnosis and better at predicting progression or recurrence in more frequent incidences of BC. With less time spent on common occurrences of BC, more time can be spent on more advanced or rare cases of BC.

1.2 Previous Work

Using DNN and classical image processing to classify and analyze different types of cancer is being researched in an increasing number of fields, this also applies to histological images which contain information about different types of cells [10, 11].

Currently, research is conducted with similar data material, but on different types of cancer. One of these is the CAMELYON challenge organized by Diagnostic Image Analysis Group (DIAG) and the Department of Pathology of the Radboud University Medical Center (Radboudumc), whose goal is to find and classify breast cancer in WSI of histological lymph node section [12].

Analyzing tissue in WSI are topics that have been given in previous master theses earlier, but each with a different goal and data set compared to this thesis. The previous master theses worked with: Predicting recurrence in BC, classifying different tissues, and creating a quantitative measurement for tumor-infiltrating lymphocytes [13–15]. The latter is the most interesting in regard to this thesis as it has the most similarities; finding cells center and classifying two classes with support vector machine.

1.3 Thesis Objective

In this thesis, the objective is to use classical image processing and DNN to analyze WSI from urinary BC patients. The goal is to estimate the number of cells present in an area and how many of these are marked with the biomarker CD25+. The thesis will be divided into two parts: Part one will be an unlabeled data approach, and part two will be a labeled data approach. The unlabeled approach will use classical image processing to locate, count, and classify cells that will be used as a foundation for training a DNN; this will be used to estimate the number of cells and whether the biomarker CD25+ has stained them or not. Part two will apply a more DNN oriented approach to estimation, using labeled data as a measure of quality.

The cells in the histological images used for this thesis will only be considered as positive CD25+ marked cells or not marked.

1.4 Thesis Outline

Chapter two will contain the background theory for BC and methods used in the thesis. The data material format will also be introduced here.

Chapter three will give information about the data set used in this thesis.

Chapter four will contain the methods, experiments, and results in an unlabeled approach to the data material.

Chapter five will look at a labeled approach to the data material, containing methods, experiments, and results.

In Chapter six the solution approach, methods, and results will be discussed.

Chapter seven will draw a conclusion based on the result of the thesis as a whole.

Chapter 2

Background Theory

This chapter will give background information around both the technical and the medical aspects of this thesis.

2.1 Medical Background

This section will cover the medical background information used in this thesis.

2.1.1 Urothelial Carcinoma

Carcinoma is a collective term for cancer types that are made from epithelial cells [\[16\]](#). The Urothelial carcinoma starts inside the lining of the bladder, the urothelial lining; however, it can be found anywhere inside the bladder.

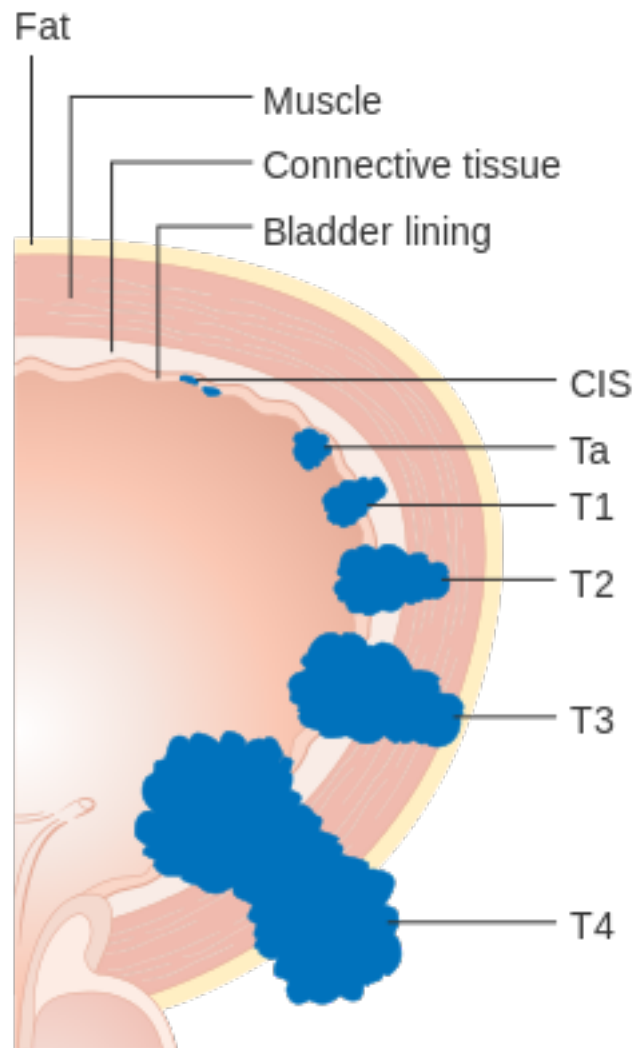


Figure 2.1: Illustration of the bladder with different the tumor stages of BC. The figure is reprinted in an unaltered form from Wikimedia commons *File: Diagram showing the T stages of BC CRUK 372.svg*, created by Cancer Research Uk, CRUK. Licenssed under [CC BY-SA 4.0](https://creativecommons.org/licenses/by-sa/4.0/).

According to the Tumor, Node, Metastases (TNM) classification system, urothelial carcinoma is divided into different stages, seen in Figure 2.1. The different letters in TNM are often accompanied by an alphanumeric code, which provides additional information. The T in TNM describes the tumor size, and if it has been spread to other tissues. N describes the lymph nodes which are included, and M gives information about cancer spread from one place to another. There are several tumor stages, but for simplicity, they can be divided into two groups, muscle-invasive and non-muscle invasive. Approximately 70% of patients at the initial presentation, are diagnosed with non-muscle invasive stages, pTa or pT1, [17].

BC is diagnosed according to the World Health Organization (WHO) grading systems, WHO04 and WHO73. Both systems are in use today to diagnose BC as the classifications system have some correlations, but are not directly interchangeable. WHO04 has three classifications, high grade, low grade and papillary urothelial neoplasm of low malignant potential (PUNPLM), while WHO73 has a grading system that ranges from one to three. For example, grade two in WHO73 can in different cases, be classified into any of the three classifications in WHO04 [4].

Treatment

To provide the optimal treatment, which is a necessity to lessen the effect cancer has on a patient's quality of life, an expert needs to make a judgment regarding the cancer grade and stage. The decision is based on the medical data, samples acquired from an invasive procedure, patients age and expected lifespan, as well as cancer stage and grade. There are several options for treatment. Some of these treatments are chemotherapy, immunotherapy, targeted therapy drugs and transurethral resection of the bladder (TURB). Cancer cells often hide or camouflage themselves so that the immune system does not recognize them as a threat. Immunotherapy is used to activate the patients own immune system to attack the cancer cells, this is done by using medicine to help the body to identify and attack the cancer cells. Bacillus Calmette Guérin (BCG) is a bacteria that can be injected in fluid form into the bladder to help trigger an immune response, that will then attack the cancers cells [18].

The most common treatment for non-muscle invasive bladder cancer (NIMBC) is TURB, which is suggested to be followed up by a single chemotherapy instillation [19]. For high-risk patients BCG immunotherapy is given along with TURB, this has shown to reduce the recurrence rate with 56% compared to TURB alone, this uses the immune system present in the bladder to attack the tumor [20].

2.1.2 CD25+ Staining

biomarkers are used in diagnosing cancer, and it is a valuable asset in creating reproducible diagnoses [9]. The CD25+ biomarker attaches itself to T-cells and give them a distinct color, giving a prognostic value. This marking last a longer on regulatory T-cells and can therefor be used together with other biormarkers. The CD25+ marked images are different from the Hematoxylin and Eosin (HE) stained images, which is used for the diagnosis of cancer. In the CD25+ images, the cells have a blue color and positive CD25+ marked cells will have a brown color around it. The difference between CD25+ stained images and HE stained images can be seen in Figure 2.2 and 2.3. CD25+ is used to

identify regulatory T-cells, which are related to suppression of immune response and cancer diagnosis [4, 9].

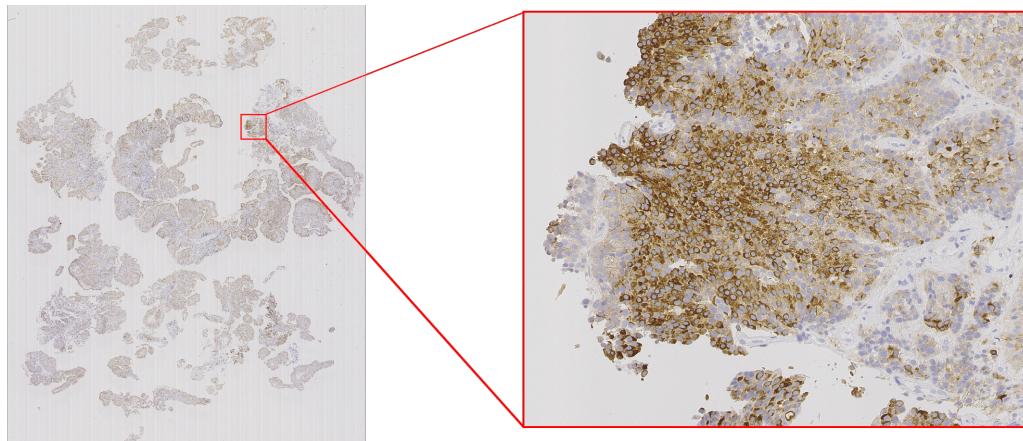


Figure 2.2: WSI marked with CD25 immuno stain. The left image is a WSI viewed at a neutral zoom, and the right images is zoomed in on a specific spot marked by the red lines on the left image.

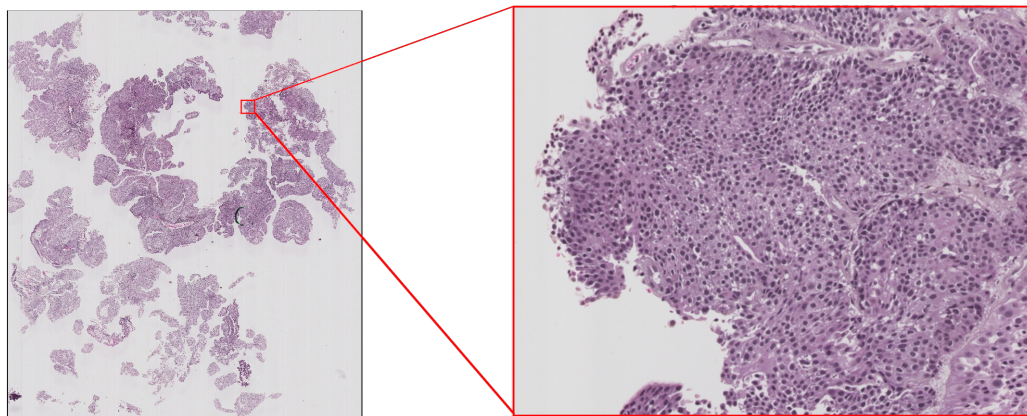


Figure 2.3: WSI marked with HE stain, often used in diagnosing cancer in histological images. The left image is a WSI viewed at a neutral zoom, and the right images are zoomed in on a specific spot marked by the red lines on the left image.

Digital Pathology

When scanning the glass slide with the tissue, called a histological slide, it will give a WSI, which has zooming capabilities similar to a microscope, but is easier to use. The WSI is structured so that it has a high-resolution histological image in the bottom and several downsampled versions over it, as seen in Figure 2.4, these downsampled versions are saved into the image format to avoid the downsampling process each time the image viewed.

This structure makes it easy to navigate at a higher level quickly and then smoothly zoom into the desired area by going through several images. In the high-resolution image, each pixel has a height and width equal to 0.25 micrometers, resulting in one image having approximately $3 \cdot 10^{10}$ pixels. The amount of pixels along with the 8-bit color bit depth gives each WSI an enormous size. Having this pyramid-like structure means it is possible to visually navigate at lower resolutions, without having to load an entire image at a max resolution into the computer memory or applying interpolations or filtering. Digitization of histological slides opened up various opportunities for pathology, such as image processing, more automated analyzes, machine learning, online sharing, and storage.

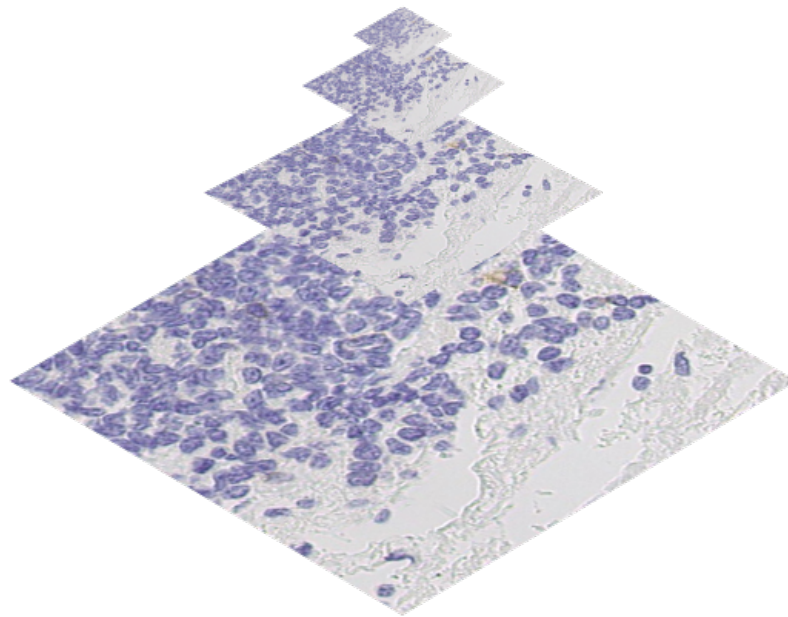


Figure 2.4: Pyramid image visualization

VIPS

To be able to handle this kind of image, special methods for viewing and processing were made. One of them was the VASARI project, which was intended to measure the color and texture changes in old master paintings. The demand for this was 20 pixels per millimeter, the resolution along with the multiple color channels produced an image too large to be handled by computers at the time. The solution was to build a c-library which uses memory-mapped files, and this library was called VIPS [21, 22]. Using this method meant that the large images did not need to be loaded as one whole image into

memory, but only the region of interest alone was required to be loaded into the memory. This file structure became known as the Tag Image File Format (TIFF).

The library VIPS has since been further developed to handle larger files. With other advancements in technology, the resolution could also be further improved to represent information at the cellular level. TIFF had the limitation of only be able to handle a file with a maximum size of 4 GB because it uses a 32-bit offset; this is not enough for modern WSI used in histopathology. This storage limitation was solved by basing a new file format, which was based on the existing TIFF, called BigTIFF. BigTIFF uses 64-bit offsets instead of 32-bit offset, making it able to handle files larger than TIFF. Using 64-bit offset resulted in the maximum storage capacity of 16-ExaBytes ($16 \cdot 10^9$ GB) for one single BigTIFF image [23].

2.2 Image Processing

Image processing is a term that collects all types of processing of an image; this can be classical image processing methods, neural networks (NN), or DNN.

When talking about classical image processing it is often, but not always, referred to less complex operations to do various tasks; to mention some of the numerous tasks: Filtering, segmentation, feature enhancing, simple object detection, and denoising. Classical image processing is used in analyzing various images looking for features based on, but not limited to, shapes, edges, texture, color, or luminance. Most images also contain additional information that is not part of the visual image; this data is called metadata. Metadata can be any information, from patient data to longitude and latitude. For hospitals, this means that one image format can have all the information regarding patients gender, age, time stamps when the image was taken, cancer grade or similar. This data is often confidential and can not be shared.

DNN is often used for more sophisticated image processing tasks such as classifications or recognition tasks, while classical image processing is often used as a way to prepare the images for the DNN so that DNN can perform better than without pre-processing. DNN outperformed classical image processing in an image processing challenge called Imagenet [24], in the year 2012. The following years most top contenders used DNN to achieve even better results. DNNs require a large amount of data to be able to achieve a good result when classifying objects; the Imagenet challenge has a few categories, where the category image classification and localization had over one million labeled images to use in the challenge 2012 [25]. DNN is therefore not always easily implemented in cases where it is hard to acquire labeled data, such as medical images. Hence image processing

is often used for simple detection when the labeled data is not sufficient for training, or used in conjunction with DNNs as a form of pre-processing and post-processing. This chapter will cover the theory of pre-processing techniques used.

Pre-processing can be divided into three steps, load the image, analyze/enhance the image and output/results. Loading the images can be accomplished by utilizing one of the many libraries that already exist, and depending on which parameters used in the method for loading the image, it can be loaded differently. Similarly to the loading of the image, the output can be saved or used in other applications with existing libraries. By analyzing or enhancing different features in the image, it is possible to make an object detector, remove noise, or make color variation different images less significant. By analyzing the features, it is possible to decide whether the image contains the right information or if it should be ignored. When processing an image and the processed image or output is to be used in further steps, a possible consequence is loss of information, if done incorrectly. Some prior knowledge about the purpose of the process is often necessary for good results.

2.2.1 Morphology

Mathematical morphology is used for analyzing and processing geometrical structures. In image processing, it provides several methods for handling shapes and features; some of the usages include removing unwanted objects, enhancing or diminish edges or image segmentation. These morphological operations are based on shapes within the image, which is binary or a luminance image.

Dilation and Erosion

Two of the most basic morphological methods are dilation and erosion. The usage of dilation and erosion on a binary image, the white pixels indicate foreground, while black pixels are considered as background. The morphological operation erode works as the name implies, it erodes the edges on the objects, making them smaller, in some cases removes them entirely. The opposite operation to erosion is dilation, this operation also as the name implies, dilates the objects making them grow. The effect of the dilate and erode operations can be seen in Figures [2.5](#) and [2.6](#).

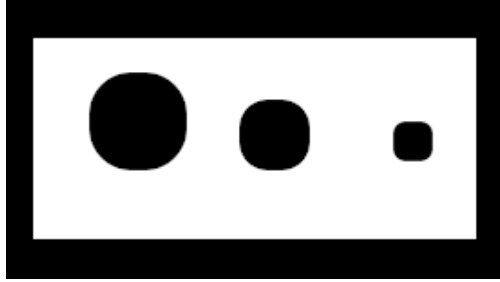
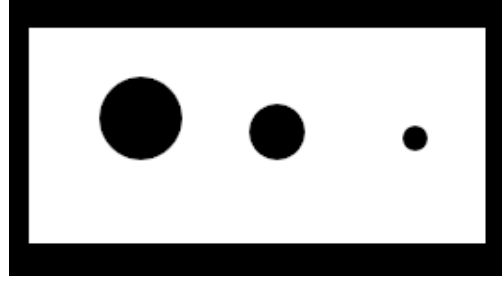


Figure 2.5: Morphological erosion performed on the figure above. It can be seen that the black area (which is considered background) is thicker than before as the foreground has been eroded away.

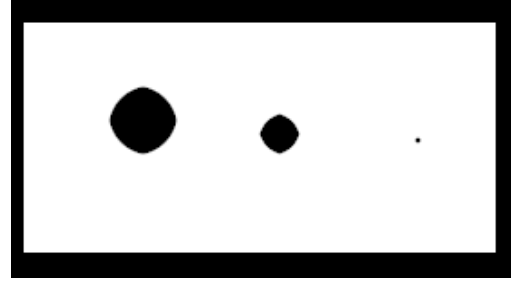


Figure 2.6: Here the morphological operation dilation has been performed on the figure above. Here it can be seen that the white area has replaced some of the black areas. The results from dilation can be seen on the smallest black dot as it has almost vanished.

Dilation and erosion are both used together with a structuring element, often called kernel, which slides across the image. Depending on the pixels inside the kernel and implementation, the center cell of the kernel is set to either black or white, seen in Figure 2.7. The two concepts for setting a pixel to either a one or a zero, are hit and fit. Fit is when all values match the one values (similar to an AND-operator) in the structure element, and a hit is when one or more of the elements match the ones in the structure element (similar to OR-operator). In Figure 2.7, the hit concepts have been used in a dilation operation. The mathematical equation for dilation and erosion can respectively be seen in Equation 2.1 and 2.2.

$$g(x, y) = (f \oplus s)(x) = \begin{cases} 1 & \text{if } s \text{ hits } f, \\ 0 & \text{otherwise.} \end{cases} \quad (2.1)$$

$$g(x, y) = (f \ominus s) = \begin{cases} 1 & \text{if } s \text{ fits } f, \\ 0 & \text{otherwise.} \end{cases} \quad (2.2)$$

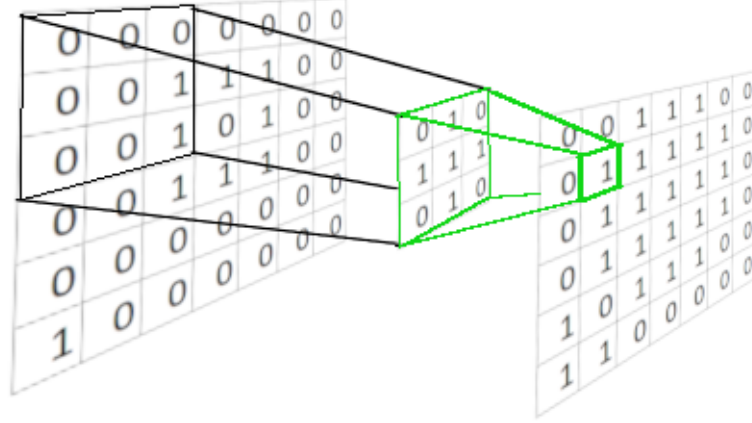


Figure 2.7: Numerical visualization of dilation. Here the structure element marked green in the middle is a three by three matrix and covers an area equal to its size on the input image. The position inside the structure element on the input image is the output position. Here it can be seen that the structure elements match one value and therefore is a hit, resulting in a one on the output image.

The combination of these two operations is in image processing called morphological opening and closing. Morphological closing is when a dilation operation is used and followed directly by an erosion operation, seen in 2.3, filling small holes inside objects while not making the object grow, resulting in less false negatives inside objects. With the morphological opening, the order is switched, and a dilation operation follows the erosion operation, as seen in 2.4. The erosion will remove small objects which are often considered noise while retaining the larger objects, if used correctly, this can remove false positives in an image.

$$f \bullet s = (f \oplus s) \ominus s \quad (2.3)$$

$$f \circ s = (f \ominus s) \oplus s \quad (2.4)$$

Distance Transform

A distance transform is used on a binary image to create a distance map. A distance map is most commonly focused on the white foreground objects and gives a distance value to each pixel depending on the length to the closest background pixel. Depending on which metrics used with this transformation, the results will vary. Two metrics that are often used are euclidean distance and rectilinear distance. For a two dimensional case, euclidean and rectilinear are calculated respectively 2.5 and 2.6. Euclidean distance will measure the length of a straight line towards the background, while rectilinear will measure the distance along one axis followed by the next axis, the result of this can be seen in Figure 2.9 and Figure 2.10.

$$Distance_{Euclidean} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (2.5)$$

$$Distance_{Rectilinear} = |x_1 - x_2| + |y_1 - y_2| \quad (2.6)$$

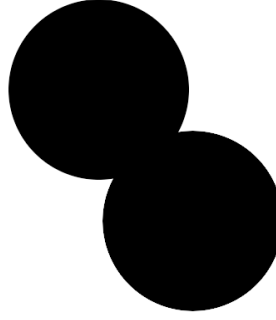


Figure 2.8: Two circles with overlap.

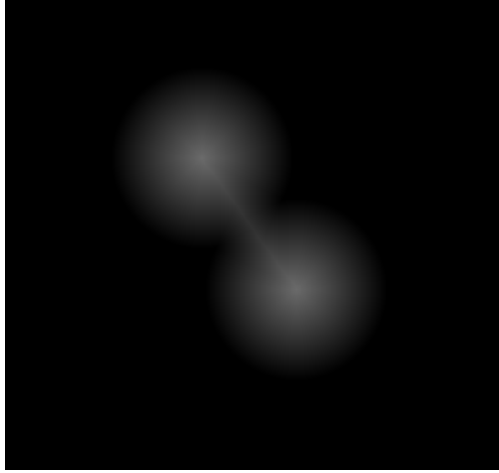


Figure 2.9: Euclidean distance transform performed on Figure 2.8. It can be seen here that the distance from the center of one circle smoothly fades out towards the background.

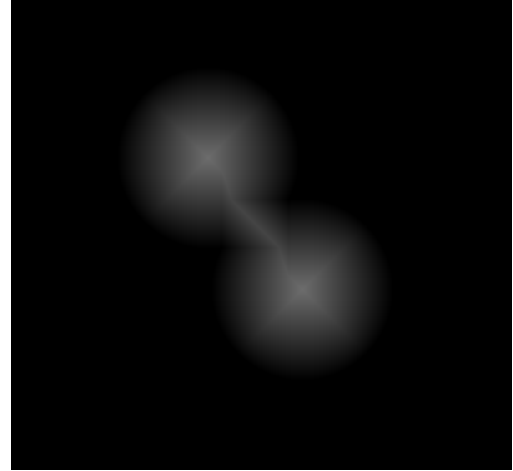


Figure 2.10: Distance transform with rectilinear, performed on Figure 2.8. Here it can be seen that there are some lines from the center of the circle going outwards, this is the result of measuring distance along one axis at a time.

2.2.2 Image Histogram

An image histogram is a graphical representation of the color values in images and is used in analyzing the content of the image; this can be interpreted as a probability density function. Histograms plots the number of pixels that shares the same value range

into bins, shown in 2.7. These bins can represent a collection of pixels within a given value range and are usually of equal size. The horizontal axis represents the number of bins, while the vertical axis represents the number of pixels in the different bins. These histograms enable a user to determine the distribution of the pixel values in the picture with one glance, as seen in 2.12

$$[t1_i, t2_i] = [min_i, max_i] \quad i = [1, \dots, \text{number of bins}]$$

$$Bin_i = \sum_x \sum_y (t1_i < image(x, y) < t2_i) \quad i = [1, \dots, \text{number of bins}] \quad (2.7)$$

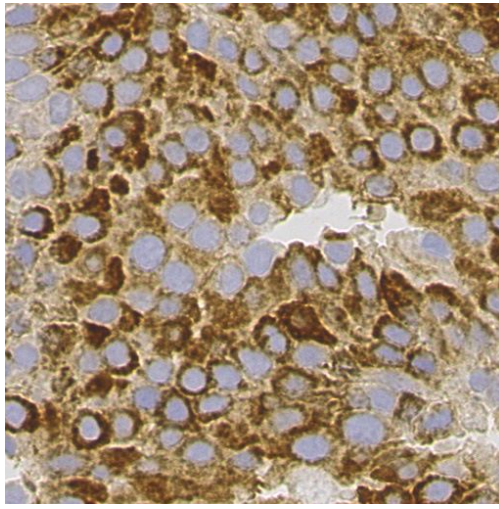


Figure 2.11: Histological image

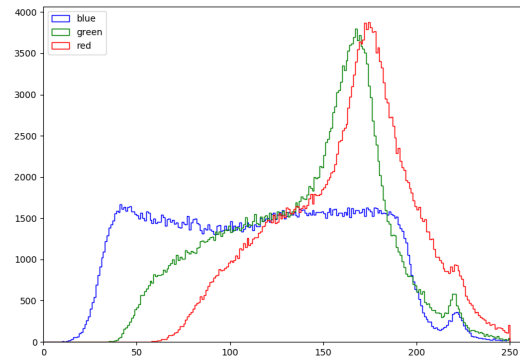


Figure 2.12: Corresponding histogram to the image on the left, 2.11. The color of each line represents the probability density function for that given color.

2.2.3 Thresholding

Thresholds are often used to make a decision and usually follows the general approach:

Algorithm 2.1 General thresholding

initialization: initialize threshold, T **forall** x **do**

if $x \leq t$ **then**
 | set $x = \text{value}$

else
 | set $x = 0$

end

end

Result: Image or object with two values, True or False

When thresholding an image with a single threshold, the output will be a binary image. Thresholds are often useful when detecting certain objects in an image, but are susceptible to being too inflexible. One common usage for thresholding is its separate foreground from background for easier processing or object detection. Since it is not desirable to manually set a threshold each time the need for differentiating between foreground and background, adaptive algorithms have been made. One adaptive thresholding technique for separating background and foreground is Otsu's method. The results of this method are achieved by iterating through thresholds and searching for the best threshold that minimizes the intraclass variance; this was found to be the same as maximizing the inter-class variance [26]. Otsu's algorithm is described in algorithm 2.2:

Algorithm 2.2 Otsu's algorithm [26]

Initialization:

```

1: Compute the histogram, and the probabilities for each bin denoted
2: Initial class probabilities denoted  $w_i(t)$  and class mean denoted  $\mu_i(t)$  for  $t = 0$ 
while  $t \geq \text{max threshold}$  do
    | 1: update  $w_i(t)$  and  $\mu_i(t)$ 
    | 2: compute  $\sigma_b^2(t) = w_0(t) \cdot w_1(t) \cdot (\mu_0(t) - \mu_1(t))^2$ 
    | 3: increase  $t$ 
end

```

Result: Best threshold is the maximum value of $\sigma_b^2(t)$

2.2.4 Color Format

Hue, Saturation, and Value (HSV) is a color format made to better mimic how the human vision perceives colors. As seen in Figure 2.13, the hue is represented as rotation in degrees around a circle for defining the color mix. This circle has primary red, green, blue at respectively at 0, 120, and 240 degrees. Saturation is the length of a vector from the center to the edge, giving the amount of brightness in the color mix, with white at the center. Lastly, the value gives a mix of black to the color, starting with black at zero and decreases as the value increases. HSV is an alternative color representation to red, green, and blue (RGB) representation which represents a color with three numerical values, one for each color channel red, green, and blue.

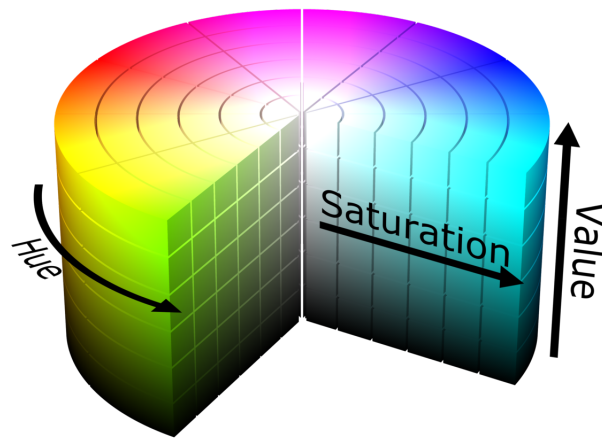


Figure 2.13: HSV representation in a cylinder. Figure reprinted unaltered form from Wikimedia commons, *file:HSV color solid cylinder.png*, created by [Datumizer](#), licenced under [CC BY-SA 3.0](#)

2.3 Neural Networks

Artificial neural networks originated from the idea to mimic a biological neural network; recreating the learning process of a biological neural network. The first step to accomplish this was to understand how a neuron works and afterward create a whole network of them, giving the NN. The first analogy between neuroscience and computer were made by W. McCulloch and W. Pitts, in a paper 1943 [27]. Figure 2.14 visualize the inputs on dendrites to the left, which travels through a tunnel-like structure called myelinated axon, towards the outputs called axon terminal. Depending on if the requirements are met or not, the neuron will fire on an output. The process from left to right in Figure 2.14 is the basis for the artificial neuron. The inputs (dendrites) on the left are connected to an activation function (axon terminal), explained in section 2.3.1, that will decide the output value.

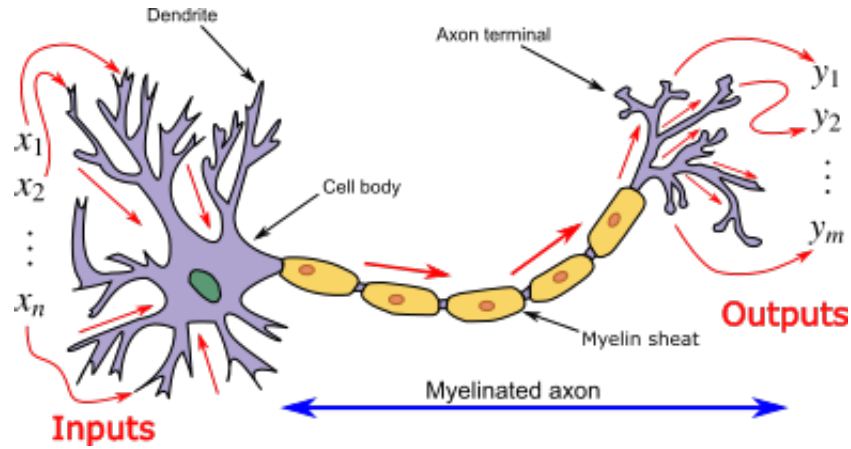


Figure 2.14: A biological neuron. Here the general idea for recreating an artificial neuron, can be seen with input(s) on the the left and outputs on the right. Figure is reprinted in an unaltered form from Wikimedia commons, *File:Neuron3.png* by Prof. Loc Vu-Quoc, licensed under [CC BY-SA 3.0](https://creativecommons.org/licenses/by-sa/3.0/)

Despite NN being a hot topic several times in history, it never got solid traction because of the lack of hardware to support the computational need. With the invention of the Graphics Processing Unit (GPU), NNs gained a considerable increase in computation speed. Steinkrau et al. in 2005 implemented a two-layer dense network on a GPU and resulted in a three-time speed up compared to the central processing unit (CPU) baseline [28]. Chellapilla et al. got similar results with a convolutional neural network (CNN) the following year [29]. In the early years of GPUs, they were mostly used for playing games that required heavy computations caused by both graphics and background tasks, to be done fast. Since the video game market has been popular, there are large corporations that are competing to make the best and cheapest GPU [30]. This has driven the development of GPUs, as well as it has reduced the prices of GPUs, making decent GPUs more available at an affordable price.

A DNN is a NN with more depth than a two-layer NN, meaning it can have several layers between the input layer and output layer; these layers are often called hidden layers. DNNs are more complex and can find similarities in data that regular NNs or even researchers could not find, leading to better results. Therefore, DNN has become more and more popular in several research fields.

2.3.1 Activation Function

An activation function sum the inputs, which have been multiplied with their respective weights, to one node outputs a value for that node. The output can be used for either classification or input in the next layer. Activation functions in NNs create non-linear properties enabling them to learn more complex patterns than a linear model.

Activation functions decides how and when the neuron fires. There are several activation functions most known ones are Rectified Linear Unit (ReLU), Tanh, and sigmoid. ReLU is the most used, since it does not have the same vanishing gradient problem as the other two have, it has also been shown that ReLU converge six times faster than Tanh on a similar networks structure [31]. ReLU is described by the following equation 2.8:

$$f(x) = \begin{cases} x & \text{for } x \geq 0 \\ 0 & \text{for } x < 0 \end{cases} \quad (2.8)$$

ReLU is often used as the default activation function, with a softmax activation function at the end of the network that outputs values between zero and one, making it a valid probability distribution. The softmax probabilities are given with the following Equation 2.9

$$\text{Softmax}(Z)_i = \sigma(Z)_i = \frac{e^{z_i}}{\sum_{j=1}^N e^{z_j}} \text{ for } i = 1, \dots, N \text{ and } Z \in \mathbb{R}^N \quad (2.9)$$

2.3.2 Fully Connected

The last layers in a classification network are often called dense layers or fully connected. In these layers, all the previous outputs are taken as input in the current layer, as seen in 2.15. For images, this means that the dense layer can gather information from the entire image instead of smaller parts of it. These layers often have more parameters than the rest and require more computation.

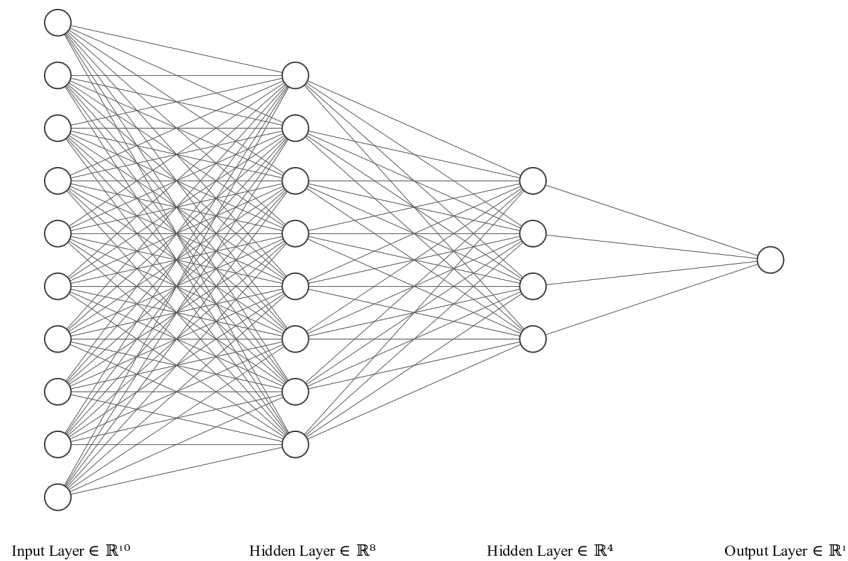


Figure 2.15: NN with fully connected layers

2.3.3 Learning

A network can learn from its mistakes by inspecting how far from its prediction it was. The learning is done by minimizing the loss function or cost function, and these represent the error in the output of the network. Reducing the loss functions is done by an optimizer which updates the weights, to find better features to match the desired output. The optimizers are used to adjust how much the weights need to be changed for each iteration. A network will typically have two measurements of how well it has performed, accuracy and loss. The accuracy of the model tells how many of correctly classified examples there were, divided by the total amount given, while loss provides information about distance from the correct class, how sure it is when classifying. These two are both important, but as accuracy goes up to 100%, loss often goes down to zero when the model is entirely sure of its prediction and will stop learning anything new.

2.3.4 Pooling Layers

Pooling functions are a way of downsampling the number of activation values in a given region. Convolutions networks have major benefits of using pooling operations; some of them are the reduction in the number of parameters that greatly reduce the computations needed for the network, enabling it to learn faster. It can also be used to ensure that varying image sizes can be used in the same network. Pooling layers in convolutional networks makes the model invariant to translation, and if used over separate convolutions parameters, the model can become transformation invariant, meaning it can learn to ignore some transformation filters [29].

There are different ways of combining neighboring values; some of the more common ones are max pooling and average-pooling. Max pooling selects the maximum value in the given area to represents that section, while average pooling takes the average value as the representative for that region, it is possible to use a weighted average based on the center pixel as well [29].

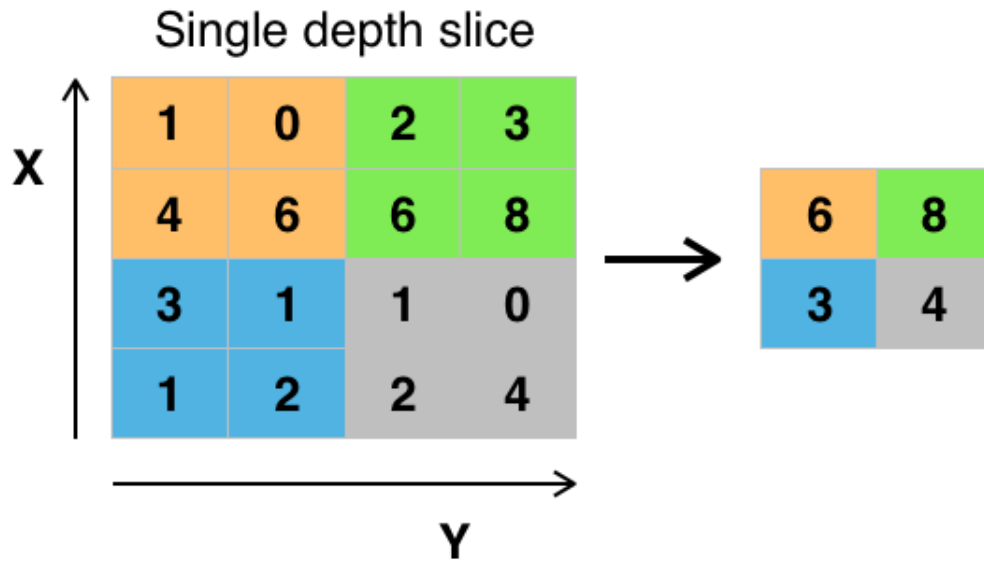


Figure 2.16: Visualization of the max pooling operation. The highest value in each color to the right is the given output to the left. Figure is reprinted in an unaltered form from Wikimedia commons, *File:Max pooling.png*, by [Aphex34](#), licensed under [CC BY-SA 4.0](#)

2.3.5 Structure

DNNs can be generalized to have at least three layers of nodes, an input layer, at least one hidden layer, and an output layer. Each layer consists of neurons which are also called nodes, and these will send a signal if the right conditions are met. The inputs to each layer are regarded as features and will be processed and sent as input to the next layer. The process repeats itself until it reaches the last layer, where a classification is made. Layers can also have a bias that will make some neurons more prone to fire than others. Between each connection in the network layers, a weight is multiplied to each feature, giving them more or less significance. The purpose of this weight is to make the network learn the best combination of features for achieving the best result. Some layers can be pooling layers, described in 2.3.4. The classification layer has the number of nodes equal to the classes decided on before training.

2.3.6 Convolutional Neural Networks

CNNs are a more specific form of DNNs mostly used for data with a matrix-like structure; this can be time-series data which can be thought of one-dimensional data or images, which is two dimensional. Just as the name implies, the usage of the mathematical convolution operator is used in this network. The operator needs two parameters, for a

two-dimensional case, it needs the input data and a kernel to slide over the image. The formula for this operation is shown in 2.10 where the asterisk represents the convolution operator. The convolutional layer consists of several learnable filters or kernels from the input; these combined covers the entire input. Each convolution layer has in general three steps, convolution operator, activation function and pooling function[29]. The structure of a typical convolution network can be seen in 2.17

$$S(i, j) = (Im * K)(i, j) = \sum_m \sum_n (Im(i, j) \cdot K(i - m, j - n)) \quad (2.10)$$

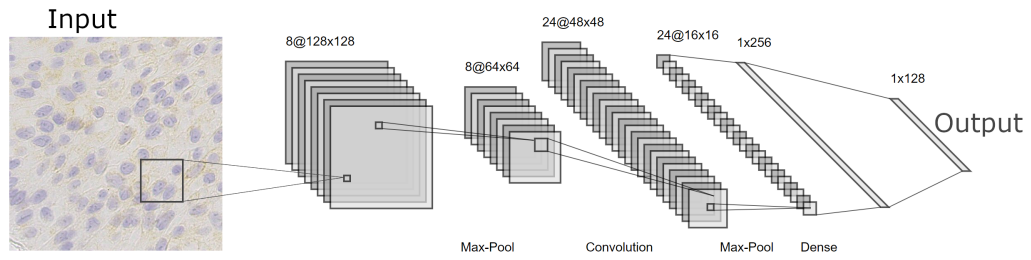


Figure 2.17: Typical CNN structure

2.3.7 Transfer Learning

When dealing with data sets with a low amount of data available, it is common to use other networks that have been trained on a large data set. The networks that are often used are the ones that have been trained on well-known challenges, such as ImageNet. One of these networks is the VGG-16 network, which has a total of 134 million parameters [32]. When using a pre-trained network as a base for a new network, it is common to remove the last fully connected layer, freeze all parameters and add a new fully-connected layer which will be trained with the given data set. The layers which will not be trained can be considered a good feature extraction to give to the new layer. This way, it is only needed to update the weight to the last layer while keeping the other layers as is.

2.3.8 Performance Validation

Evaluating the performance of different models is essential to describe how well the model did overall and in specific areas. In this section, different tools and approaches for measuring performance will be explained.

Confusion Matrix

A confusion matrix is used to visualize the overall performance of a classification method easily, this can be seen in Figure 2.18. On the diagonal of the confusion matrix, it can be seen the correctly classified classes, the rest of the confusion matrix consists of falsely classified classes. For each class one have true positives (TP), true negative (TN), false negative (FN) and false positive (FP), these are used to describe how the model work for each given class, the terms are explained in 2.1. TP for a non-binary classification can be interpreted as: is the given point inside an area or not; this is also called hit. From the values in a confusion matrix, it is possible to calculate precision, sensitivity, and accuracy.

Precision for a class is the percentage of all predictions of that class correctly predicted. The calculation can be seen in equation 2.11.

Sensitivity for a class is the percentage of TP for that class divided by all whose original label belongs to that class. The calculation can be seen in equation 2.12.

Accuracy is the overall performance of the model, the calculation can be seen in equation 2.13.

True Positive for class A	Class A Correctly predicted as class A
False Positive for class A	Wrong class predicted as class A
True Negative for class A	Class B predicted as class B
False Negative for class A	Class A predicted as class B

Table 2.1: Overview of terms used in evaluation performance

$$Precision_{ClassA} = PC_A = \frac{TP_A}{TP_A + \sum(FN)} \quad (2.11)$$

$$Sensitivity_{ClassA} = SC_A = \frac{TP_A}{TP_A + \sum(FP)} \quad (2.12)$$

$$Accuracy = \frac{\sum all TP}{Total population} \quad (2.13)$$

		Predicted class		
		Class A	Class B	Total
Ground truth	Class A	80 TP_{AA}	10 F_{AB}	88.9% $Sensitivity_A$
	Class B	3 F_{BA}	40 TP_{BB}	93.0% $Sensitivity_B$
Total		96.3% $Precision_A$	80% $Precision_B$	90.2% $Accuracy$

TP = True Positive

F = False Class

Figure 2.18: Confusion matrix with two classes. Specificity and precision are calculated from values in the confusion matrix

Chapter 3

Data Material

This chapter will give an overview of the data material available for this thesis.

3.1 Data Set

The data set consists of 95 patients with the earliest tissue sample from 2011. Each patient has several WSI available with different biomarkers; in this thesis, the CD25+ marked histological images are used.

The Department of Pathology at Stavanger University Hospital has done the digitization of the tissue samples. The scanner used to create the digitized histological images was SCN400. This scanner according to Leica uses a custom-designed optics and focusing methods to be able to scan with different magnifications 5X, 10X, 20X and 40X, where the last one has a resolution of $0.25\mu m/pixel$ and takes only 4 minutes to scan [33].

The labeled data set made available mid-may contains 183 512x512 images labeled by an expert pathologist, with only positive markings for CD25+; the rest of the cells have been manually marked as non-positive CD25+ by the author of this thesis. Out of the 183 marked by the expert pathologist, only 98 images were marked with both positive and non-positive CD25+ markings. The total amount of markings consisted of 114 positive CD25+ markings and 5542 non-positive markings; an example of this can be seen in [3.1](#).

The labeling process for the thesis was done with Labelbox. Labelbox was used with a local server on a private laptop to avoid uploading images to an external server, the labels were outputted in a .csv file.

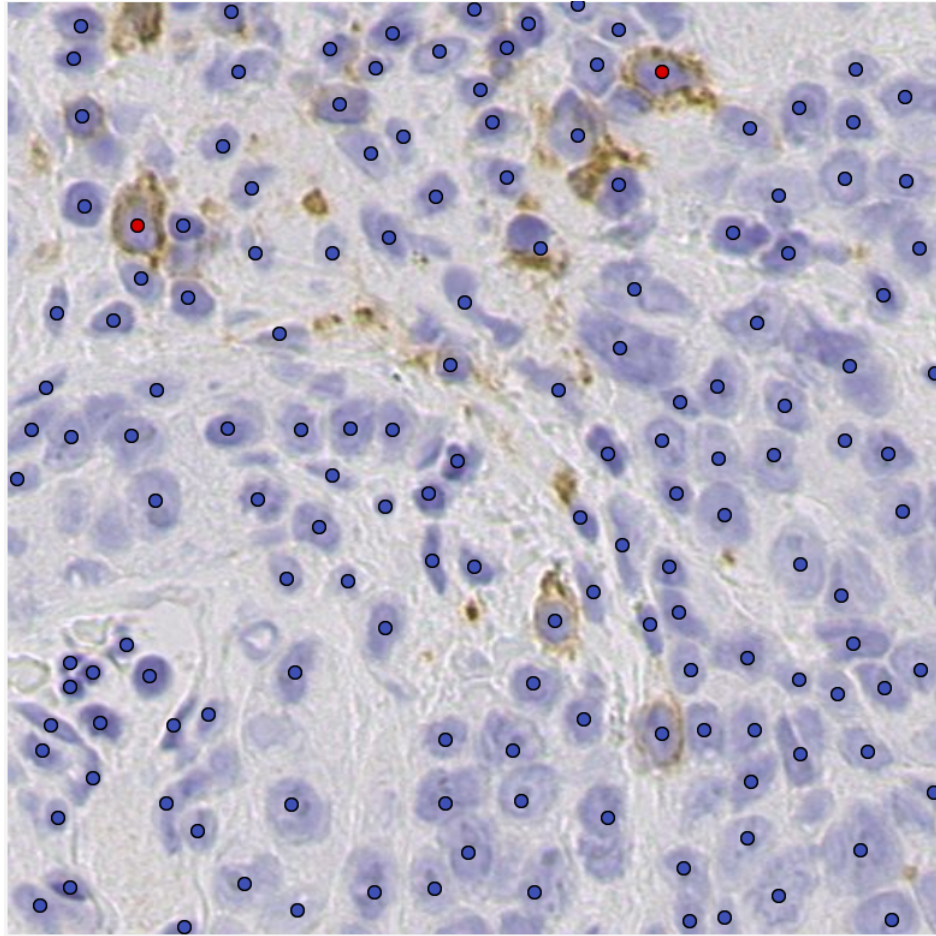


Figure 3.1: Labeled data, the red colored dots signifies positive CD25+ markings and have been confirmed by a pathologist, the blue colored ones have been placed as non-positive marking. The coordinates for each dots have been saved in a comma separated value (.CSV) file format.

Chapter 4

Unlabeled Data Approach

This chapter will cover an approach to use the provided unlabeled data to analyze and enhance features in the WSI to get an estimate of the content in the WSI.

4.1 Methods

This section will give an overview of the tools used, the proposed system, and the implementations of different methods. The first part will be about pre-processing data material, followed by the estimation and creation of labeled data using classical image processing; this labeled data will be the baseline for DNNs.

Hardware Setup

Since the WSIs are classified as confidential information because of its metadata regarding time stamps and some patient data, therefore all data is kept on the UiS server, Gorina4. Gorina4 is a GPU-server which consists of six high-end GPUs; four of them being Tesla P100-PCIE-12GB and the last two being Tesla P100-PCIE-16GB. As the data material was located on this server, all computation was also done on the server, even the ones that used the CPU, such as classical image processing. Since this is a GPU server, the CPU is not as high end as the GPUs. Running large CPU operations on the GPU server is not ideal, but was needed for processing the WSIs.

Software

For viewing the WSI images, the open-source software QuPath was used together with the remote desktop software NoMachine [34]. Making it possible to examine the whole

WSI, without breaching the confidentiality contract; having the full WSIs on a personal computer would be a breach of contract.

4.1.1 Methods

An overview of the unlabeled approach can be seen in Figure 4.1. This system has three different outputs: first, a region of interest that will be future work, the second output is cell localization and classification with the use of classical image processing and lastly an output from the DNN model.

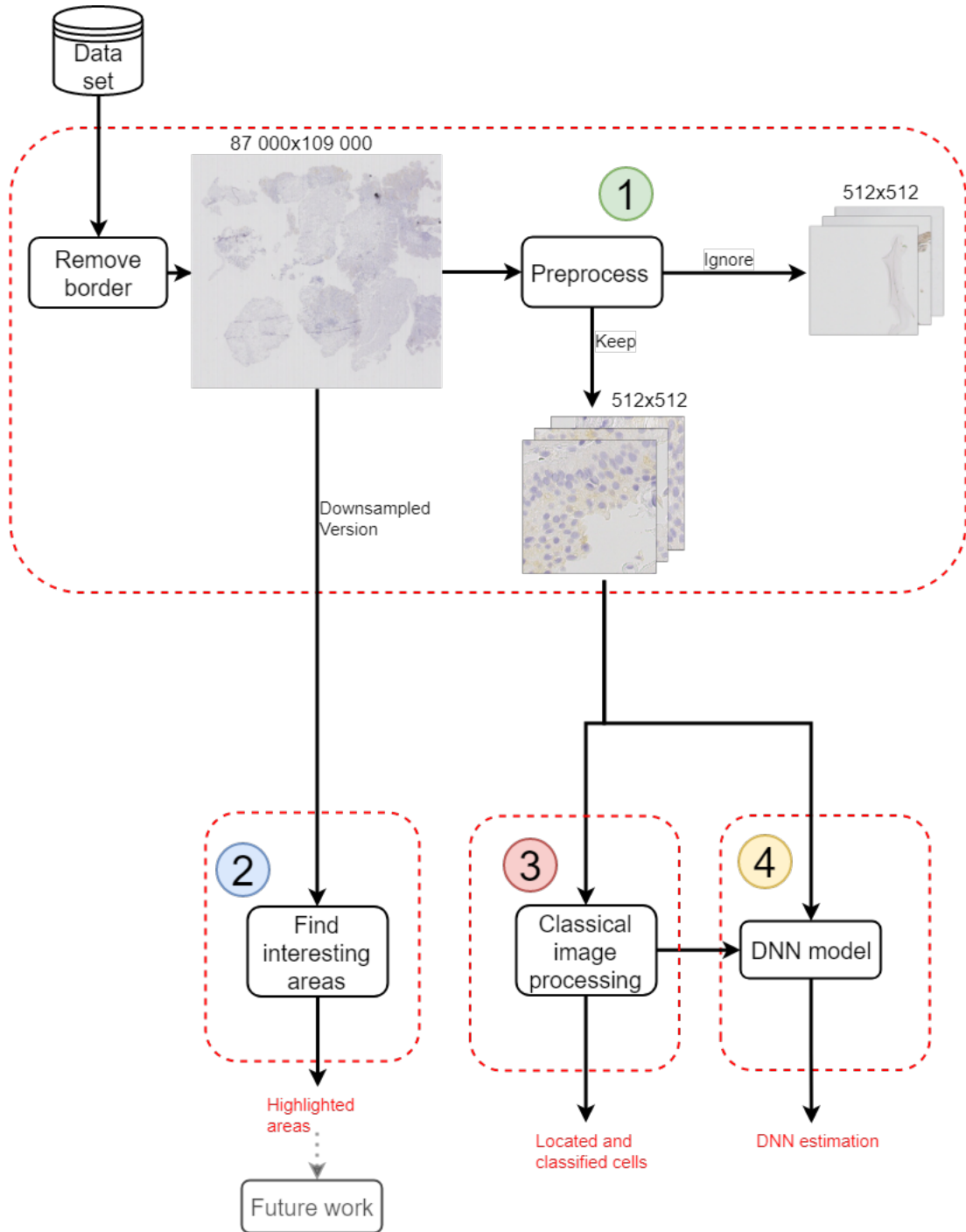


Figure 4.1: Overview of the process for unlabeled approach. Each number represents a more detailed system that will be explained in the following sections.

4.1.2 Pre-Processing Whole Slide Images

The WSI is used as described in 2.1.2; to prevent system crashes only a small part of the image will be processed at a time. The WSI size can be reduced by removing the border area around the tissue scan part. When the WSI has loaded, this area is pure

white and does not contain any vital information regarding the cells. This white area can be filtered out by looking for the RGB color value [255, 255, 255] from each side of the image. Looking at each pixel is too time-consuming and gives an excessive amount of computations. Consequently, when removing the border, the step size will change accordingly to distance towards the middle. The program starts at the edge of the border and has a significant initial step size which will become smaller when getting closer to the middle point of the WSI until it converges at the edge of the histological image. The pyvips library has in the later versions added a method for removing this quicker than looking from each side towards the middle.

This section will go over steps used in the process marked with the number one in Figure 4.1, a more in-depth overview can be seen in Figure 4.2

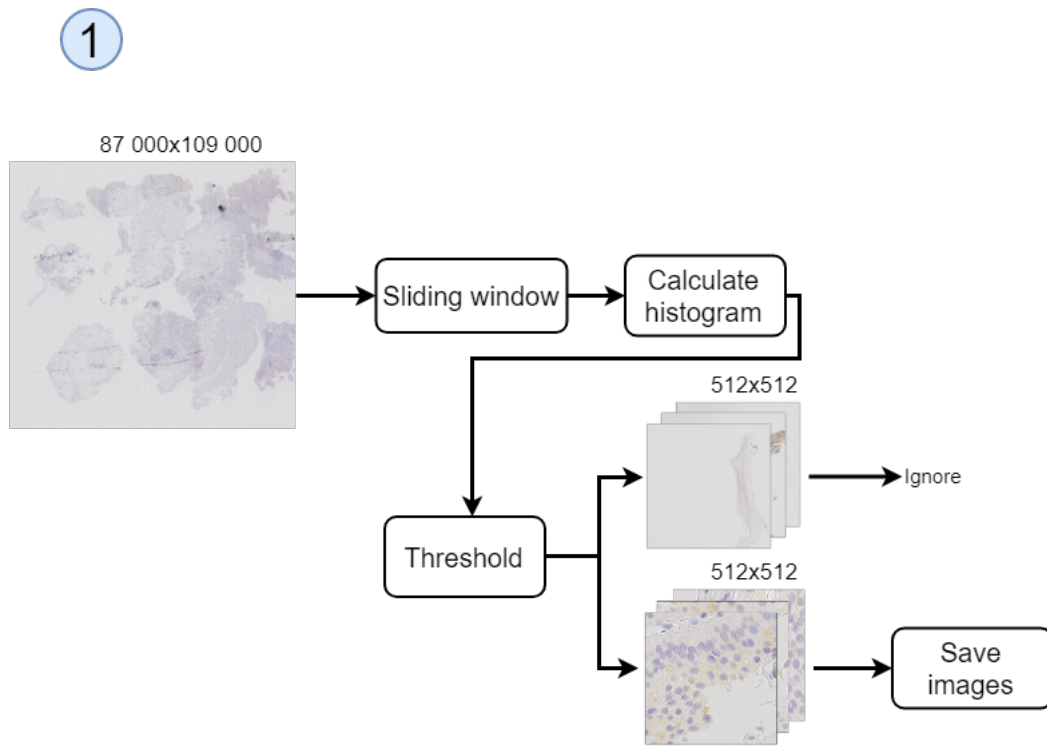


Figure 4.2: Step by step process for pre-processing the image. Images with no informational value will be ignored, while the ones with informational value will be saved.

This process receives a relatively large image of roughly the size 100 000 by 90 000 pixels (varying with each WSI) in the .scn format; this is the maximum resolution available. This resolution was selected as it contains the most information available in the WSI, giving the model the best possible starting point. Using this resolution gives an image that is too large to be loaded into memory, and only looking at smaller parts of the image is still necessary. The goal of this process is to split the large WSI into smaller images of size 512x512 pixels while ignoring the images which contain mostly background.

Separating regions into either containing useful information or not, is decided by the amount of non-tissue area captured in the region. By using a histogram to determine if an image majorly contain non-tissue area or tissue area, it was possible to separate the two by setting a threshold.

The 512x512 size was chosen at an early stage as it seemed reasonable large and could be split into smaller if needed; also, it was easy to evaluate the performance of different methods used. The smaller image of size 512x512 pixels is used in the rest of the thesis to avoid loading pre-processing multiple times.

4.1.3 Interesting Areas

This section will cover the number two marked in Figure 4.1, this is shown in more detail in Figure 4.3. Interesting areas will be regarded as the first output of the proposed system and the methods used be explained here.

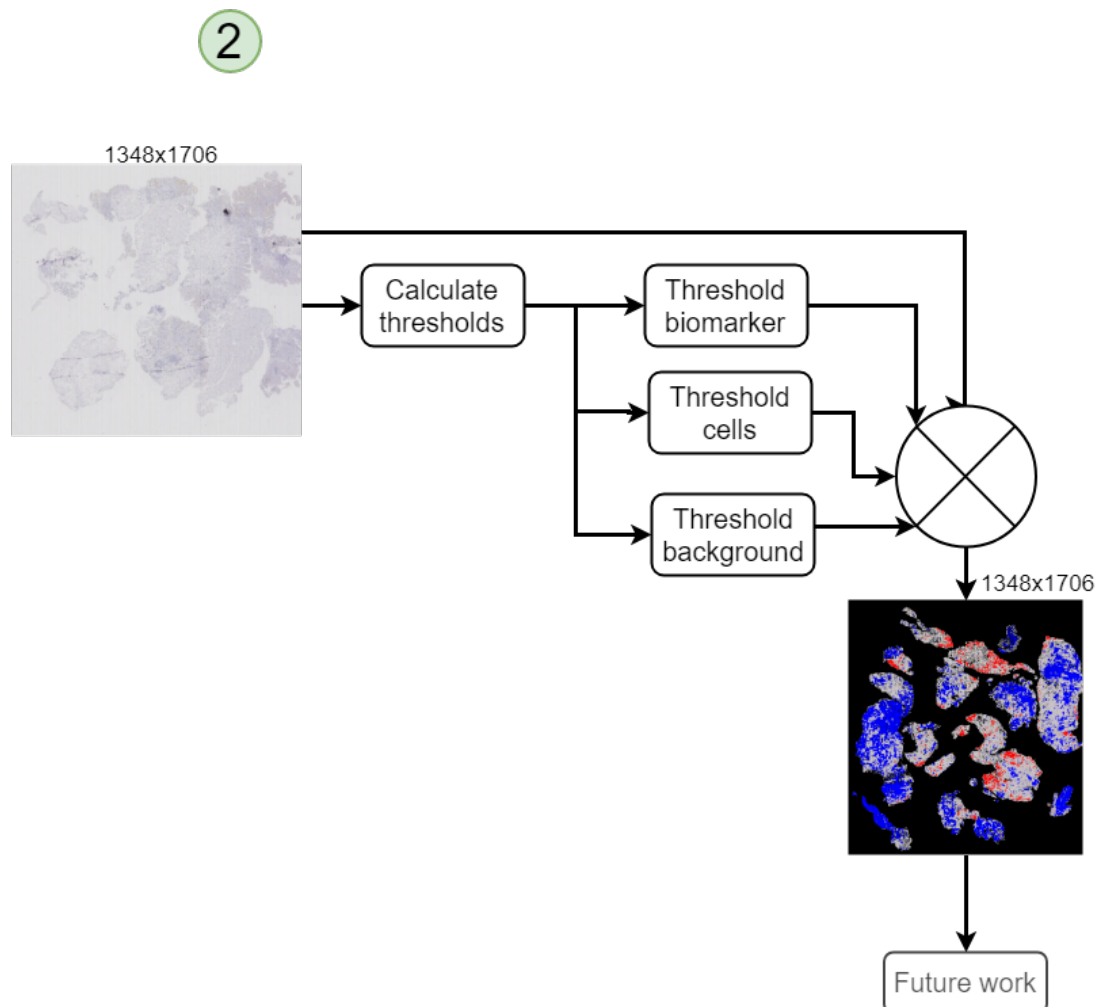


Figure 4.3: Process overview for finding a region of interest

This section will look at the WSI at a lower resolution which will fit into the memory, without having to split the image, but the border was removed as it contains no valuable information. Using a lower resolution has several advantages, such as fewer computations, giving a faster process. Processing the WSI is done by utilizing a series of thresholds to separate the color left by the CD25+ biomarker, clusters of cells (hot spots) and tissue area without cells. For separating the different colors, the different RGB color values have been analyzed, and the color red was observed to be higher in the brown area compared to the blue value; the threshold for brown is based on this observation. The non-tissue area was observed to have a higher value in all three color channels and was thresholded out based on that observation. Lastly, the blue cells are intuitively thresholded on the blue value being larger than the other two colors. The process for finding a region of interest is also shown in Algorithm 4.1.

Algorithm 4.1 Creating a map of highlighted areas

Initialization:

```

1: Image = downsampled WSI
2: Set threshold t1 and t2
forall Pixel in Image do
     $value1 = pixel_{red} - pixel_{blue}$ 
    if  $pixel < t1$  then
        | 1: Keep original pixel
    else if  $value1 > t2$  then
        | Set pixel to red
    else
        | Set pixel to blue
    end
end

```

end

Result: Highlighted image

4.1.4 Classical image processing

This section will explain how classical image processing techniques were used to label the different cells and estimate the number of cells in a given image. These labels will be used as both training and validation data in a DNN.

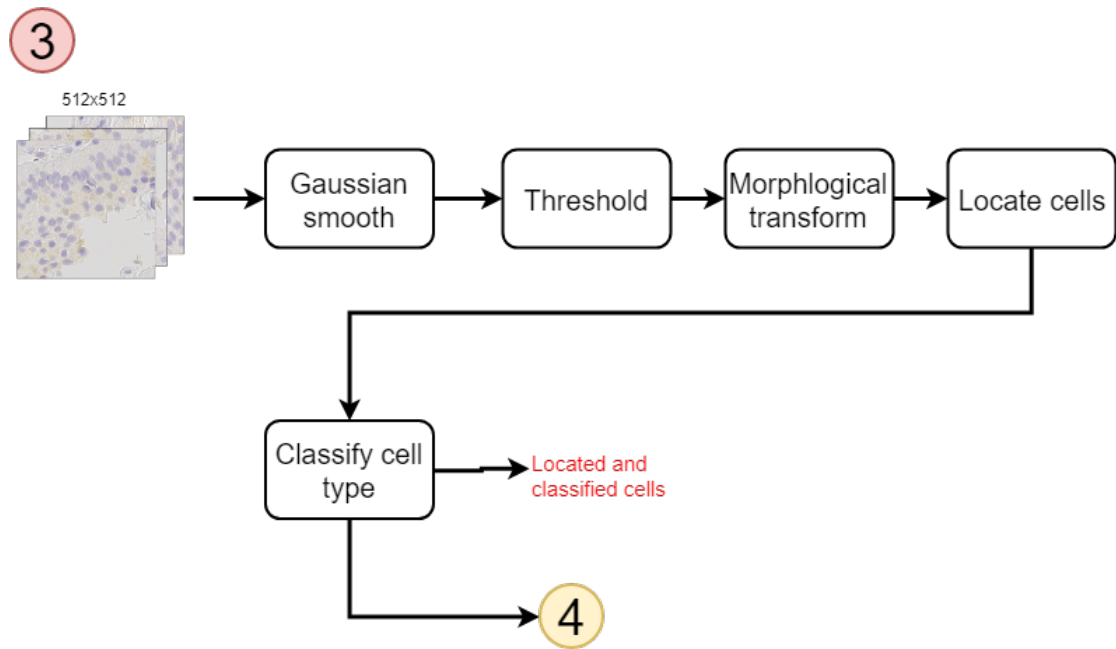


Figure 4.4: Step by step process for classical image processing used

The images were smoothed using a Gaussian smoothing, with a kernel of size 3x3 and sigma of 0.5; this smoothing leads to less FP in the image, after the thresholding step has been applied, as seen in 4.5. Since the color contains information regarding a positive CD25+ marking or not, converting to grayscale can lead to a loss of information. Therefore ideally, the threshold should cover two different color values, blue and brown; creating two different masks. Using a threshold for the color blue will give the position of the cells, as most cells are blue in the center even if the CD25+ biomarker has positively marked them. The blue color depends on where the slice cut the cell and the cell type. The brown mask will be used to find the positive CD25+ markings. Finding a threshold that covers a range of intensity for both blue and brown, is less intuitive using the RGB color format, especially with a mixed color such as brown. Using the HSV color format 2.2.4 it is more intuitive and has therefore been used instead of RGB. The thresholding process with HSV consists of selecting an upper and lower value for hue, saturation, and value and removing everything that is outside these two values. One focus here was testing several threshold ranges and choose the one that performed best. For quick testing, a python graphical user interface was used to test different parameters.

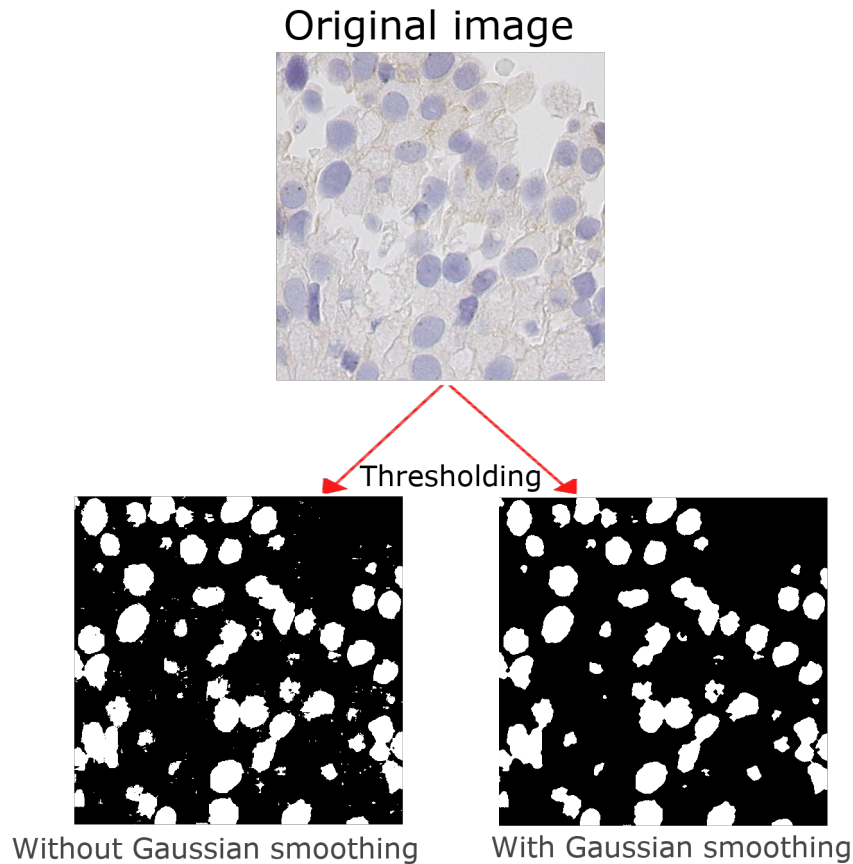


Figure 4.5: Original image that is to be thresholded is shown at the top. The bottom left is the thresholded image without smoothing. The bottom right image is with Gaussian smoothing, with a kernel size of three by three and sigma of 0,5. The bottom right image has less false positives compared to the left image.

Since some images differed from the rest in terms of color range, the thresholds did not get equal results every time, in some cases, the amount of FP in the blue mask and FN inside the cells was a problem when locating cells. Using the morphological operation opening it was possible to remove the false positives in the general image and morphological closing filled the holes inside the cells, making the mask better to use further processing. To find the center of the cells distance transform was used on the blue mask to create local maxima in the center of the cells. By using distance transform some overlapping cells created two maxima indicating two or more cells depending on how many are overlapping and how much they are overlapping.

Algorithm 4.2 Locating cells with classical image processing

Initialization:

Get all 512x512 images

forall *Images* **do**

- 1: Change color format to HSV
- 2: Gaussian smooth image
- 3: Set thresholds upper and lower limit for blue and brown
- 4: $Mask_{Blue} = Threshold_{Blue}(image_{HSV})$
- 5: $Mask_{Brown} = Threshold_{Brown}(image_{HSV})$
- 6: $Mask_{Blue} = Morph_{Close}(Morph_{Open}(Mask_{Blue}))$
- 7: $Dist = DistanceTransform(Mask_{Blue})$
- 8: cells = Find all localmax(Dist)

end**Result:** List of cells found

Locating different cells and cell center was done by finding the local maxima in matrix output from the distance transform. A method for locating the local maxima was to compare each element to the neighbors and tell if its value was larger or not. This way, the matrix would be left with only the center of each found cell. A better way is to use the library OpenCV to locate maxima with the help of two functions, one for creating labels based on connectivity with a structuring element and the other method for locating objects in those labels. This method returns some additional cells centers inside the same cell and was removed by looking at the distance to the already located cells, as seen in Algorithm 4.3

Algorithm 4.3 Removing duplicates inside same cell**Initialization:**

```

1: Get all cell centers from 4.2
2: Set threshold x
forall new_cell in cells do
     $x, y = \text{position}(\text{new\_cell})$ 
    forall already_checked_cell in checked_list do
         $x_{old}, y_{old} = \text{position}(\text{already\_checked\_cell})$ 
        if  $\text{abs}(x - x_{old}) < x$  and  $\text{abs}(y - y_{old}) < x$  then
            1: Remove cell
            2: Break
        else
            1: Add cell to checked_list
        end
    end
end

```

Result: List of cells

After all cell centers have been located, the labeling process will begin. This labeling process consists of looking at each cell center and the amount of the brown mask pixels that are inside a given area around the cell, see algorithm 4.4. Mirror padding was applied if the cell were too close to the edge of the image; this padding had the same size as the area used to classify the cells. These weak labels are used as a basis for deep learning.

Algorithm 4.4 Labeling cells.**Initialization:**

```

1: Get cell centers from checked_cells 4.3
2: Get mask for brown colour
3: set threshold T foreach cell in cells do
    1: Create a box around cell
    if  $\sum_{\text{pixels}} (\text{brown mask inside box}) > T$  then
        | give CD25+ label
    else
        | Set all cell as not CD25 marked
    end
end

```

Result: labeled list of cells

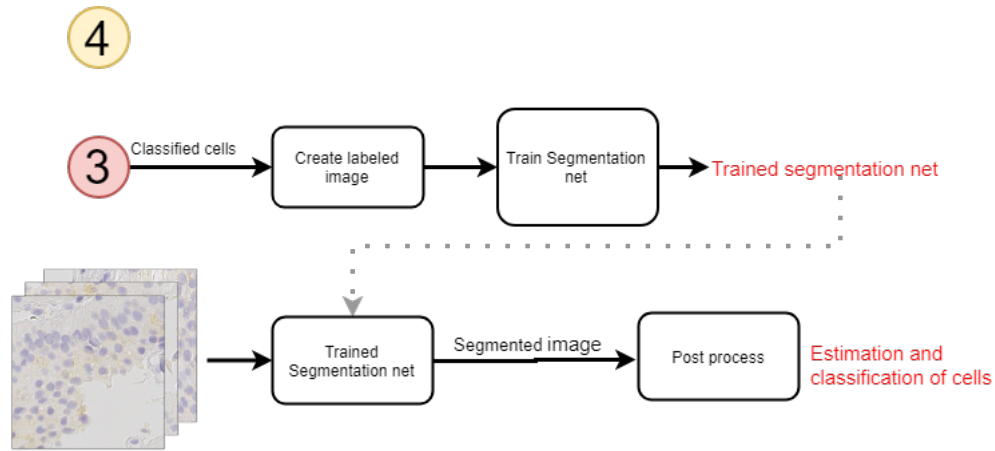


Figure 4.6: Overview of unlabeled approach to segmentation network

4.1.5 Deep Learning with Weak Labeled Data

Given the unlabeled data, classical image processing could be used to create weakly labeled data. A segmentation network seemed like a good choice since the classical image processing had already created some masks that could be used towards weak labeling the data. To create the weakly labeled data used in training and validation of the segmentation network, three different masks (brown, blue, and background) are combined into one weak labeled image. The segmentation network will create a segmentation map with three classes, and this map would need to be post-processed to get an estimation of the content in the original image. The general overview is shown in 4.6.

4.1.6 U-net

The segmentation network, U-Net was explored because it has gotten much attention for its capabilities within different medical applications [35, 36]. U-net is given a regular image and tries to create a segmentation map of the different classes given. The architecture of U-net can be seen in Figure 4.7, it can be seen here that the U-net has layers that skip the max pooling operations and go straight to the adjacent layer, this is seen in Figure 4.7. This way the segmentation network can have more information when creating the segmented image. The image can have a smaller size than the input image depending on the implementation of the U-net; if this is the case, then the information in the skip layers will be cropped to fit the adjacent layers. The general approach is shown in the algorithm 4.6.

Algorithm 4.5 U-net approach with unlabeled data**Initialization:**

Get all images

forall *images* **do**

- 1: Get cell $Mask_{Blue}$ and $Mask_{Brown}$ from algorithm 4.2
- 2: Find $Mask_{BG}$ by setting all zero-values in $Mask_{Blue}$ and $Mask_{Brown}$ to one.
- 3: Combine $Mask_{BG}$, $Mask_{Blue}$ and $Mask_{Brown}$ to create weakly labeled image
- 4: Save labeled image with same name as the original image in folder "labeled_Image"

end

- 1: Distribute the labeled images into Training data and validation data
- 2: Train U-net with three classes: Cells, CD25+ marker and background
- 3: Get segmentation image for raw data using the trained U-net

Result: U-net model trained on weakly labeled data

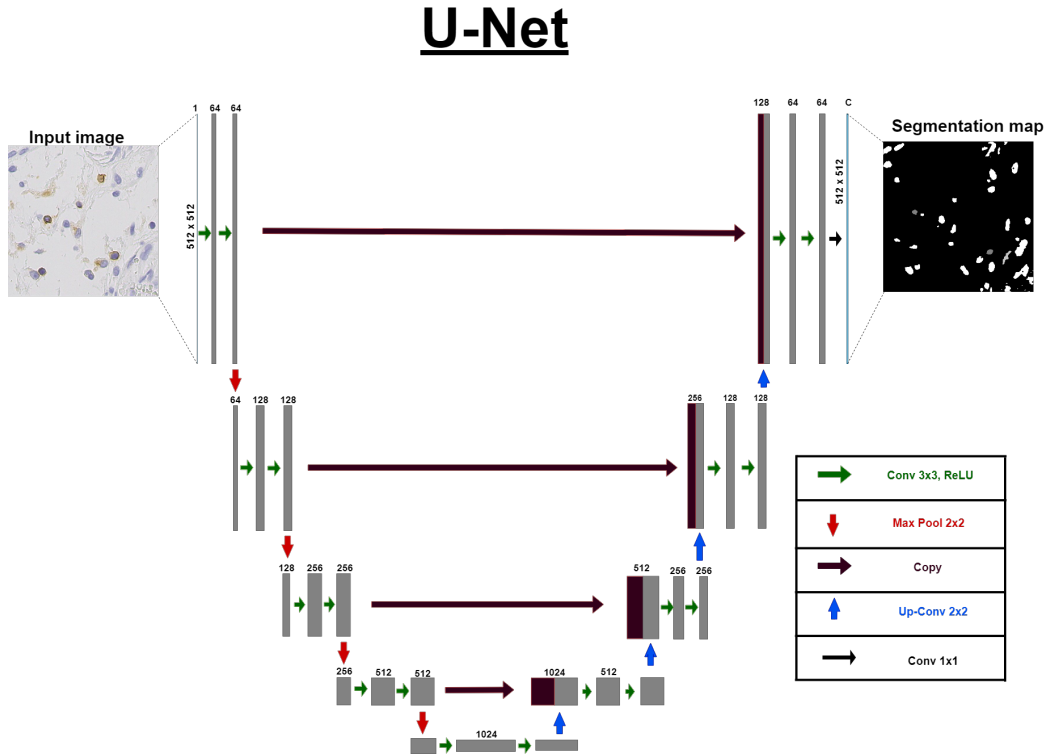


Figure 4.7: The architecture of U-net. Here it can be seen that the U-net outputs a segmentation map of the input image.

The idea was to use the U-net to create a segmentation map of cells and the CD25+ marking. If a U-net trained on weakly labeled data can manage to segment out cells and CD25+ markings, then post-processing will be used to find, count and classify cells similar to the way it was done in classical image processing. The reason for using a

segmentation network to do the segmentation is to find better features, apart from just colors, to find and estimate the distribution of cells.

Algorithm 4.6 Post processing for U-net

Initialization:

- 1: Get all trained U-net
- 2: Send in raw data to U-net and get segmentation map **forall** *Segmentation maps* **do**
 - 1: Apply distance transform to class cells
 - 2: Find local maxima in distance transform
 - 3: classify cells with the CD25+ class
 - 4: Count all cells with classification either positive CD25+ or non-positive

end**Result:** Estimate of cell distribution

4.1.7 Implementation

When looking through thousands of images, it is preferred to use existing libraries such as Numpy, OpenCV, ndimage and similar, to do the computations; these are often optimized and runs faster than most self-written code that has not had optimal optimization as the goal. Therefore using embedded or external code has been implemented where it was possible. Some methods marked as self-made in 4.1 consists of several external methods combined to create the desired result. The U-net implemented was based on a U-net used for grayscale images, but changes were made to take in colored images instead of grayscale images [36].

Method	Embedded	External	Self made
Gaussian smoothing		x	
Threshold HSV			x
Histogram		x	
Morphological transforms		x	
Local maxima		x	x
Labeling cells			x
U-net		x	x

Table 4.1: Implementation for unlabeled data approach

4.2 Experiments and Results

4.2.1 Pre-Processing

The process of splitting and removing non-tissue was run on the CPU at Gorina4; time usage, amount of image saved and ignored is listed in [4.2](#).

Patients	deleted	saved	time used
ID1	12891	4139	0:38
ID2	24900	4108	0:47
ID3	28910	10502	0:47
ID4	17873	5722	1:06
ID5	25114	16481	0:31
ID6	21271	7737	1:46

Table 4.2: Number of images saved and deleted for a subset of patients, as well as time used on each WSI.

4.2.2 Finding Region of Interest

The result for of the process for finding areas marked by the CD25+ biomarker and other hot spots of cells is shown in the image [4.8](#). The method managed to find areas with high amounts of CD25+ positive markings and areas with a high density of non-marked cells (hot spots), an expert pathologist has not evaluated this for prognostic value. This output can have several usages, and these will be discussed in the chapter discussion [6](#) and future work [7.2](#).

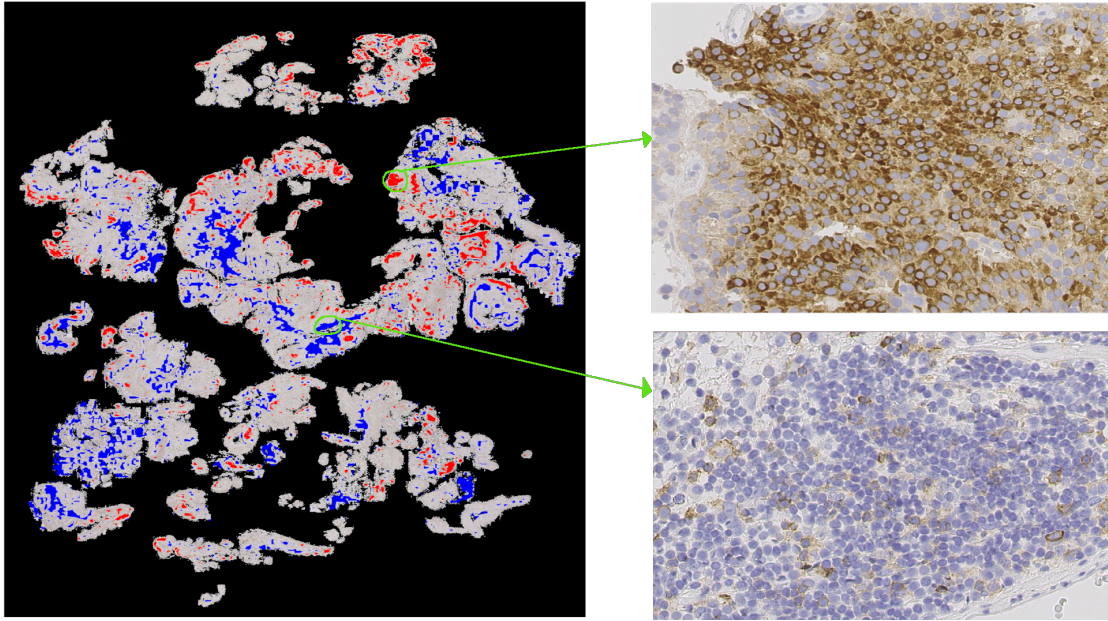


Figure 4.8: The green arrows from left to right indicates where the right images originates from. The red color is areas that have large amount of CD25+ markings while the blue areas are hot spots with other cells

4.2.3 Cell center

The localization of cells have been visually inspected at random samples; some of the results can be seen in the figures 4.9. The method places a dot in the center of each cell and counted these, to get an estimation. From visual inspection it can be seen that some of the markings are in between two cells that are close to each other; this is a by-product of the smoothing and thresholding, but in general, most cells were marked in the center. Using the detection of cell center as a way to count the number of cells found gave promising results, this can be seen in the images in Figure 4.9, and is summed in 4.3.

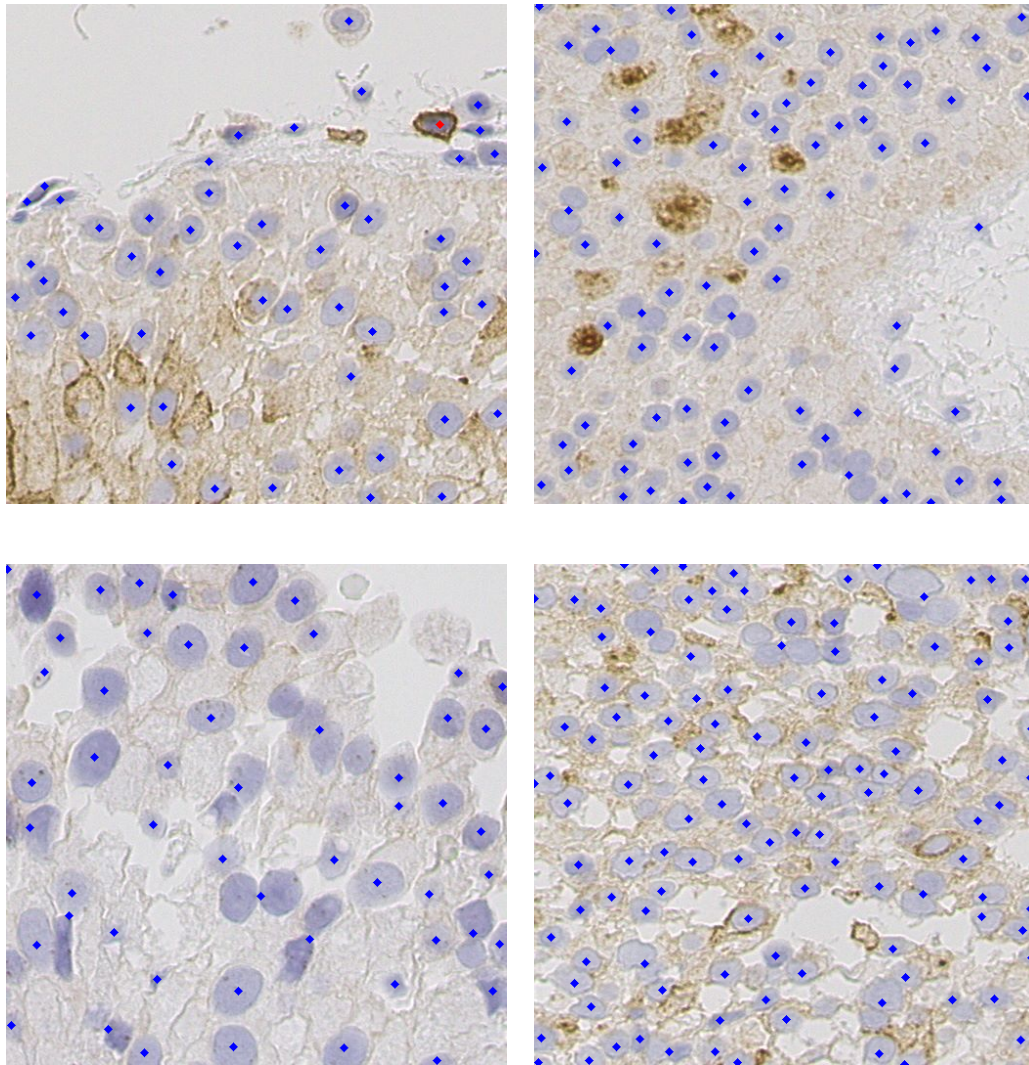


Figure 4.9: Classical image processing used to mark cells in four different images. The cells found have been marked with blue or red dots, depending on class they were classified to. These dots are only used for visualization of the localization and classification method

Image	Image processing found	Visually counting	Difference
Top left	52	54	2
Top right	90	99	9
Lower left	54	56	2
Lower right	118	133	15

Table 4.3: Manually counted cells by the author of this thesis, compared to the cell count found with image processing

4.2.4 Classification

By visual inspection, the classification managed to classify which cells were surrounded by the CD25+ marker and cells that were not. In Figure 4.9, the red dots signify CD25+ positive markings while the rest of the cells are marked with blue dots. In the upper left image in Figure 4.9, a clear positive CD25+ marking can be seen in the upper right section. While the other images have the CD25+ marking color, it is considered as leftovers of the biomarker and considered as a background color instead of a positive marking. In some images, this causes misclassifications as it surrounds a cell and the entirety of the area around.

4.2.5 U-net

With the given weak labeled masks from the classical image processing, the U-net implemented in this thesis was unable to learn the features needed to segment the three different classes, positive CD25+, other cells, and background, in the images. The U-net ran for 240 epochs and showed no improvements, that were visually checked on the validation images every five epochs. The lack of results was believed to be caused by the lack of proper training data for positive CD25+ markings.

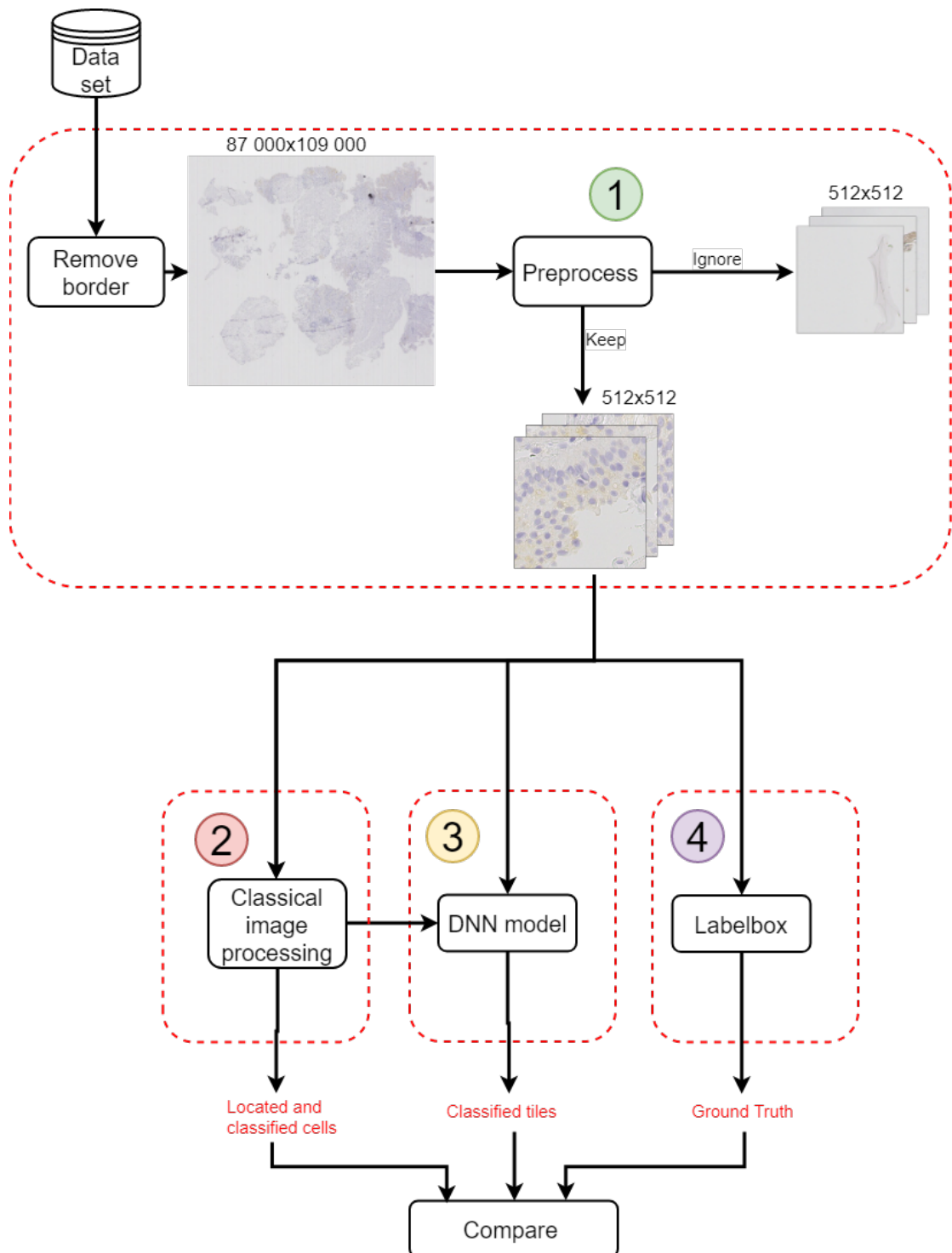
Chapter 5

Labeled Data Approach

With a set of labeled data being made available mid-May, it was possible to explore some different approaches in evaluating and testing different methods. The previous chapter explained how the images were pre-processed and how weakly labeled data was created; this will still be useful as the size of the labeled data is not large enough to support training, validation, and testing, the pre-processing steps will still be the same.

5.1 Methods

This section will explain which network that was used, how it was trained, and tested. The proposed system, as a whole, is similar to the previous chapter with slight changes seen in [5.1](#). Since the new labeled data received was marked with dots, the U-net will not be used as it requires masks. This chapter will propose a classification network to see if it is possible to get an estimation of the different marked cells. Since the labeled data contains only positive and non-positive CD25+ markings, another classification is needed for classifying none of the other classes. The last class is a non-cell class which has been made with image processing.

**Figure 5.1:** Overview of the proposed system

5.1.1 Classical Image Processing

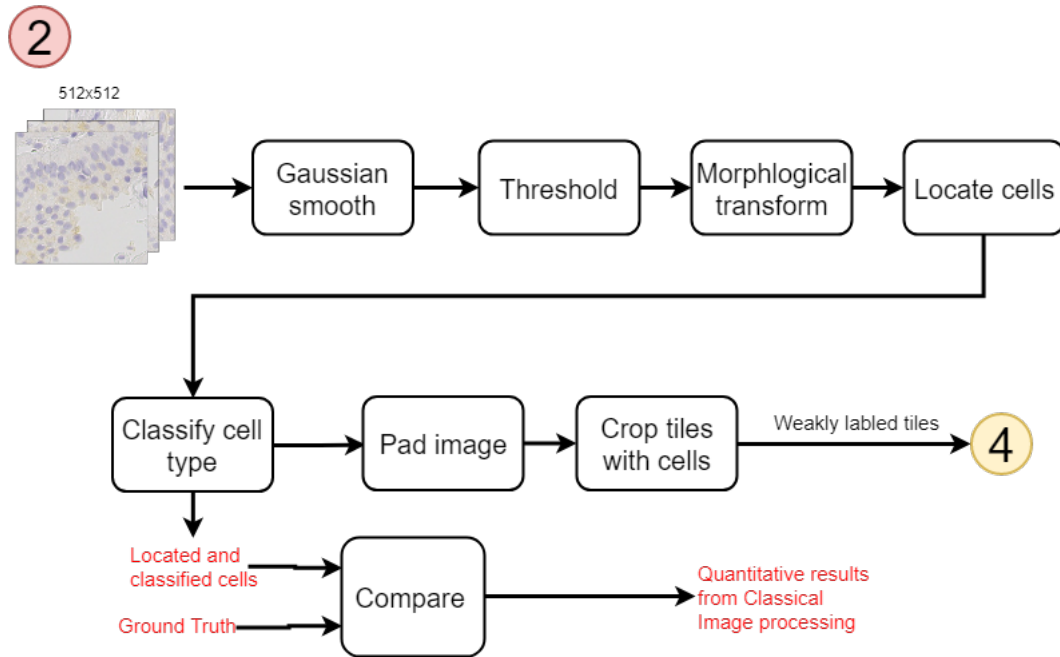


Figure 5.2: Modified classical image processing

The pre-processing shown with the number one in 5.1 remains unchanged, but the classical image processing has been changed to provide labeled tiles for a CNN network, this can be seen in 5.2. Therefore the code for extracting classes in the classical image processing is changed to finding areas with no cells; this is used as third class. Finding the empty area is done by using the blue mask from the previous chapter to perform the opposite task it was made for, finding areas that have no cells. The results from the labeling and location process in the previous chapter will be used to crop an area around each cell, attempting to make tiles with a cell in the center each time. For the third, non-cell class, an area with the complete absence of the blue mask in the center area is cropped.

Deep Neural Network

The usage of a classification network to estimate the amount of cells, a CNN called VGG-16 is used. The process for training and using the VGG-16 is seen in Figure 5.3 will.

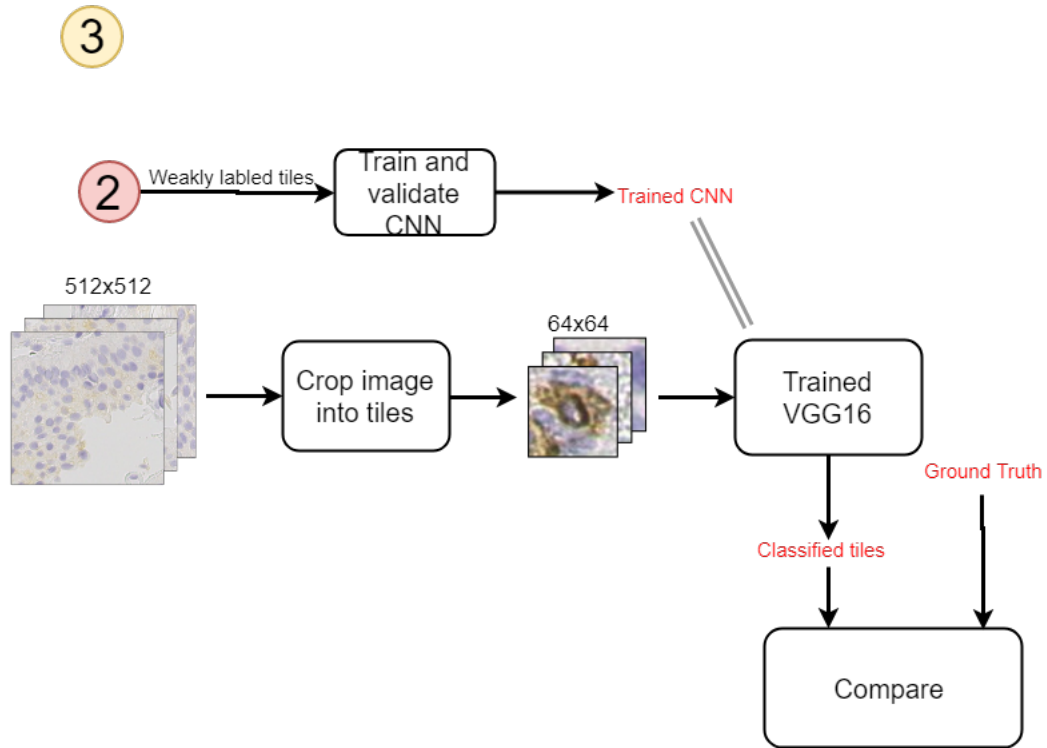


Figure 5.3: The VGG-16 network is trained with weakly labeled tiles from classical image processing. The trained model receives images that are cropped into tiles that will be classified into the different classes (positive CD25+, other-cells, and background). The output will be compared with ground truth images from the labeled data set.

VGG-16 was chosen because it is a network that consistently performs well on the Imagenet challenges [25] and a study on histology in breast cancer found it was one of the two networks that received the best results in patch-based classification [37, 38]; it has also been well documented. VGG-16 architecture requires heavy computations to be trained to achieve the good results it has in the Imagenet challenge, but this is not needed as transfer learning will be used as described in 2.3.7. Utilizing transfer learning allows removing the last layer and insert a new one, with three outputs instead of the original 1000 outputs. Since the original network architecture had 1000 outputs in the last fully connected layer, it had 409 7000 parameters connected to the last layer alone, with only three outputs the model will have 3003 parameters to update.

The VGG-16 network receives approximately 8400 weakly labeled tiles to be trained and validated on. These images are from the classical image processing step and have been cropped with a cell in the center for class CD25+ marked and other cells, the non-cell tiles are placed in the third class. The test data have been cropped similar, but the cells have been picked out by an expert pathologist. The background class has been picked out by the classical image processing, based on the lack of cells. The background class was visually confirmed to not contain a cell in the middle, some of them contain part of cells at the edge, but only a minor part of a cell.

Algorithm 5.1 VGG-16 transfer learning**Initialization:**

- 1: Get tiles for positive CD25+ marking, other cells and non-cell tiles from image processing
- 2: Divide into training and validation 3: Load pre-trained VGG-16 model
- 4: Freeze all parameters in the VGG-16 model 4: Replace the last layer with a new dense layer with three outputs.
- 5: Train VGG-16 model (update weights for the new layer) with weakly labeled tiles.

Result: VGG-16 model trained on weakly labeled tiles

After training the classifier has been given a set off test images that have been marked by an expert pathologist. These images are initially 512 by 512 images but are split into smaller tiles. Splitting the image is done with a sliding window starting top left of the image and using a step size equal to the tile size; this is done without any prior knowledge about the positions of the cells, as seen in Figure 5.4.

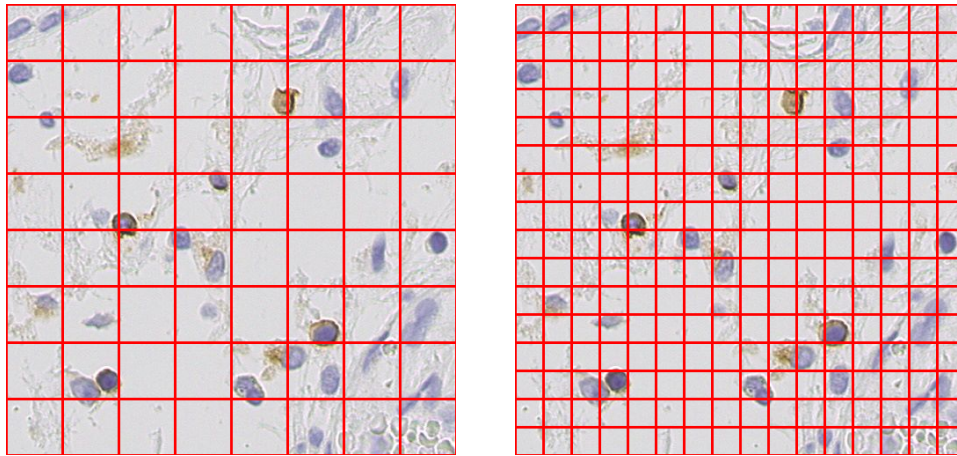


Figure 5.4: 512 by 512 image evenly split into different sized tiles

These are also cropped to create training and labeled data for a CNN. The amount of positive marked cells and other cells, as well as their location, is compared to the ground truth.

5.1.2 Implementation

In Table 5.1, the implementations of the different methods used in the labeled approach can be seen. The pre-processing or classical image processing used in Chapter 4 is not taken into account here, but can be seen in 4.1. The methods that are marked with more

than one "x" is a combination of all "x" marked. Some of the PyTorch embedded methods are a collection of several PyTorch methods to create a whole method. An example of this is the updating weights; this method contains several sub-methods connected to update the weights.

Method	Embedded pytorch	External	Self made
Padding		x	
Cropping tiles			x
Labling tiles			x
Distribute data			x
Data transforms	x		
Load training, Validation and Test data	x		
Load pretrained	x		
Updating weights	x		x
Hit and miss evaluation			x
Storing results		x	x

Table 5.1: Implementation of used methods in the labeled data approach

5.2 Experiments and Results

This section will explain how each model is tested and how well it performed compared to the ground truth.

5.2.1 VGG-16 Trained on Weak Labels

A network model trained on weakly labeled data, and tested against professional labeled data, will indicate if the trained model managed to learn the needed features to be able to discern the different classes correctly. The best model achieved an accuracy of 99 percent on the validation set and an accuracy of 95 percent on the test data. In the Figure 5.5 and Figure 5.6 trained models' performance can be seen.

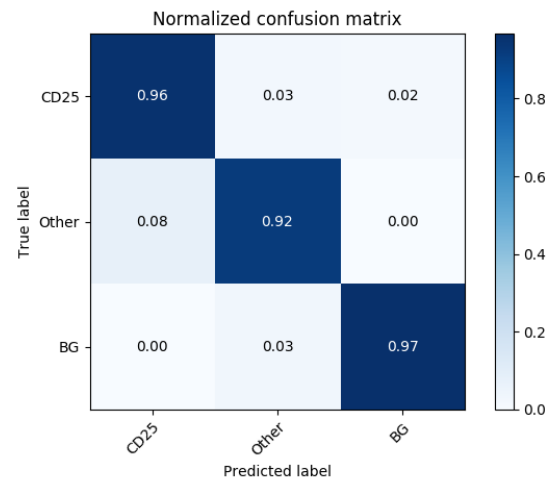


Figure 5.5: Performance of the VGG-16 model used trained on weak labeled data (using 32x32 tiles), tested on test data labeled by an expert pathologist

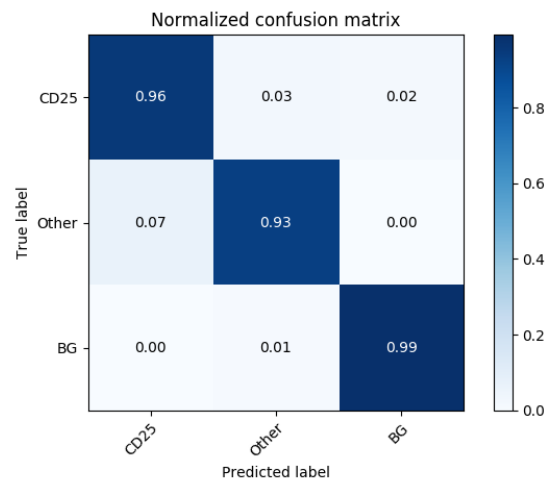


Figure 5.6: Performance of the VGG-16 model used trained on weak labeled data (using 64x64 tiles), tested on test data labeled by an expert pathologist

For a more realistic approach, the images were randomly cropped without any prior knowledge and fed to the model, the results from this can be seen in Figure.

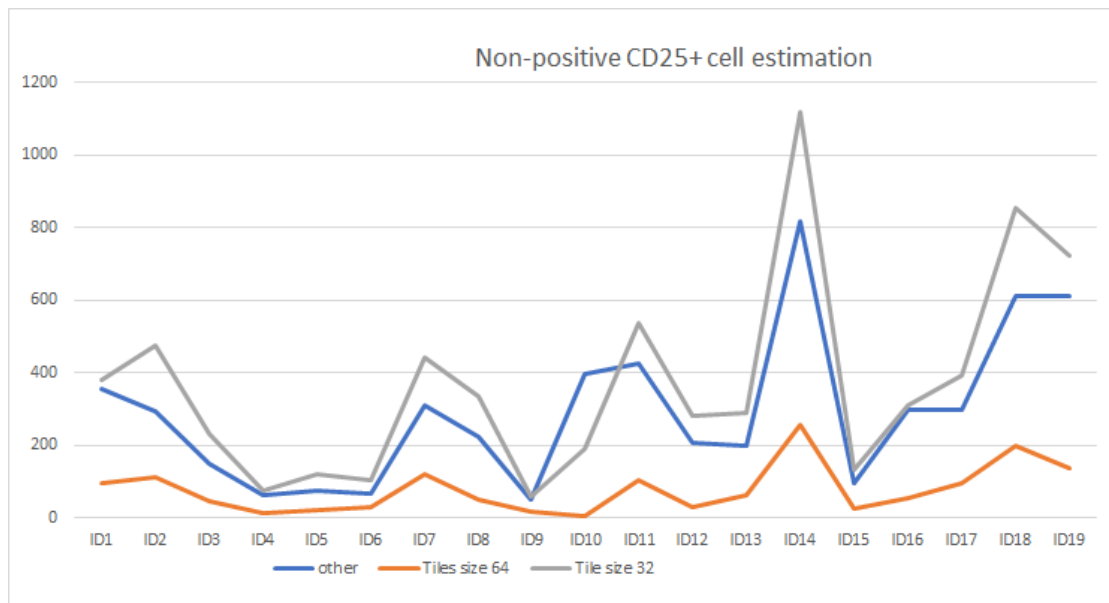


Figure 5.7: This graph shows the estimation of non-positive CD25+ marked cells with the use of VGG-16 model, trained on weak labeled data. The blue line indicate the correct number of cells.

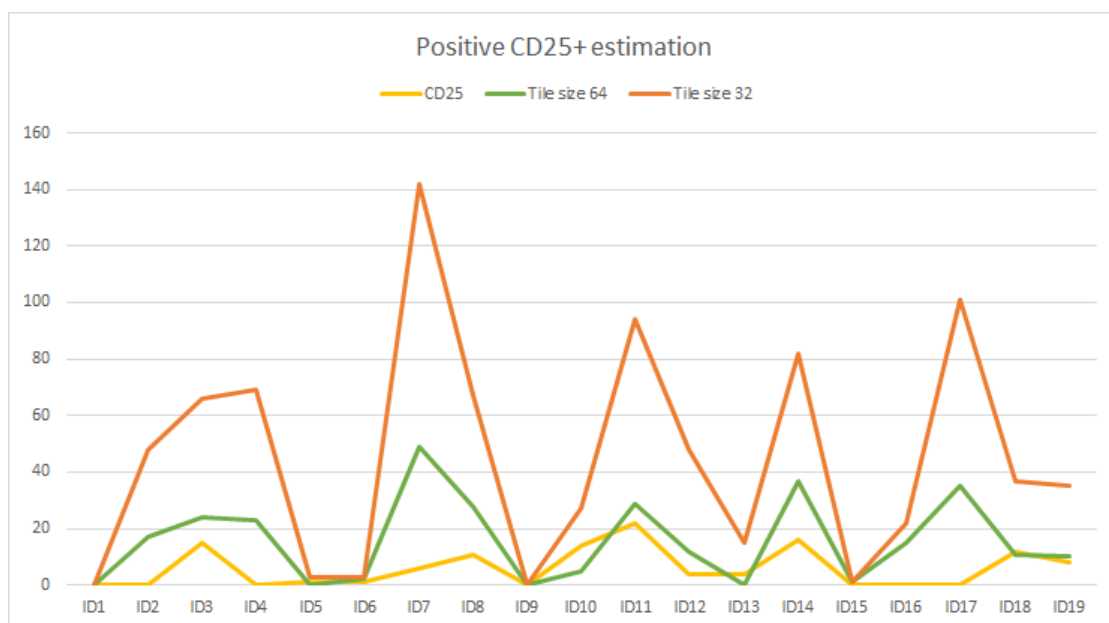


Figure 5.8: This graph showed the estimation of positive CD25+ marked cells with the use of VGG-16 model, trained on weak labeled data. The yellow line indicate the correct number of cells.

5.2.2 Classical Image Processing Methods

With labeled data made available, the methods from the previous chapter can also be tested in new ways for more qualitative results. With an expert pathologist classifying

the cells, the classical image processing will be split into two parts for testing: Locate and classification.

Cell Detection

The classical image processing chapter stated that even if the total estimate of cells was similar, some of the cells did not match the cell centers. A hit and miss approach was used to validate the localization results from classical image processing. Since the image localization method removes duplicates of cells within a given range of each other, a radius of equal size minus one pixel will be used to check if the mark is inside the cell or not.

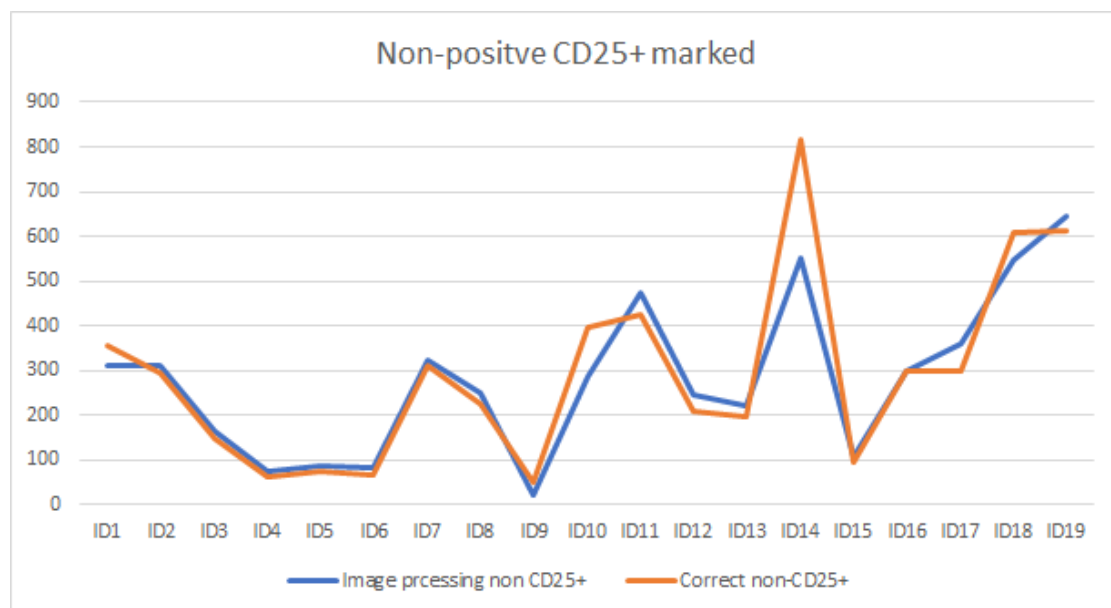


Figure 5.9: Correct amount of non-positive CD25+ marked cells (orange line) compared image processing results

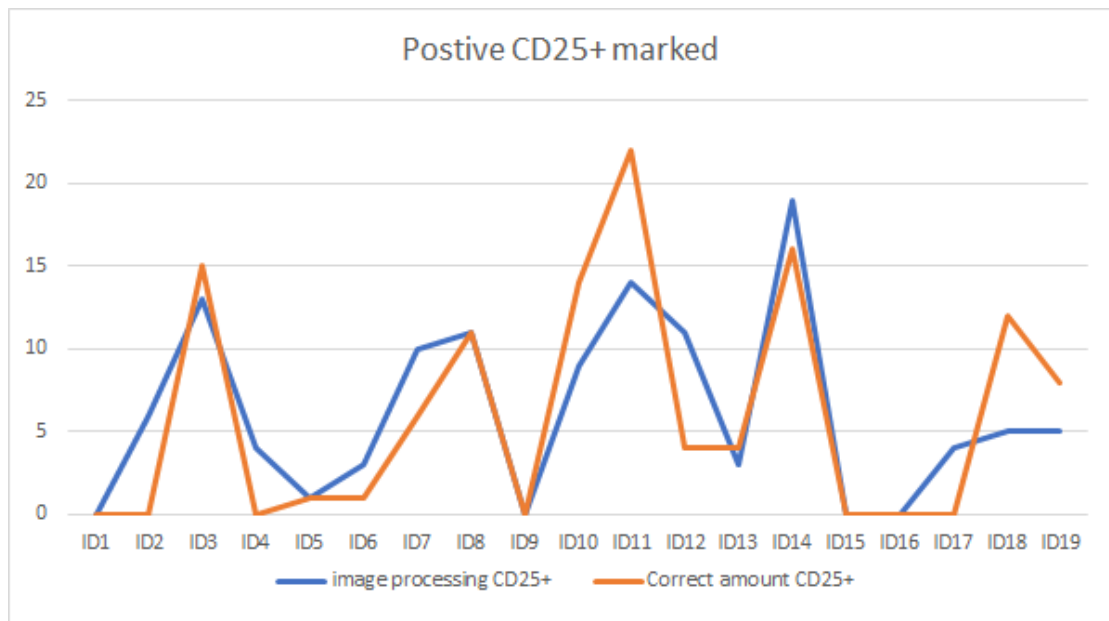


Figure 5.10: Correct amount of positive CD25+ marked cells (orange line) compared image processing results

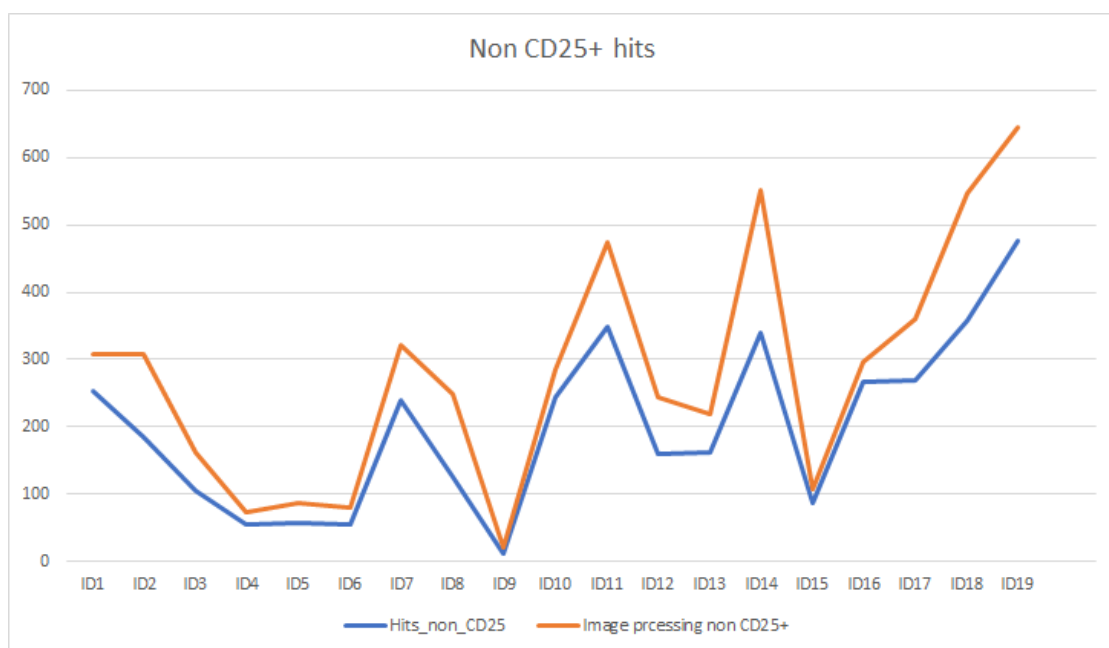


Figure 5.11: Amount of non-positive CD25+ that were close enough to ground truth to be considered a TP. The cells are from the image processing detection and the distance to the actual cell is calculated from the ground truth.

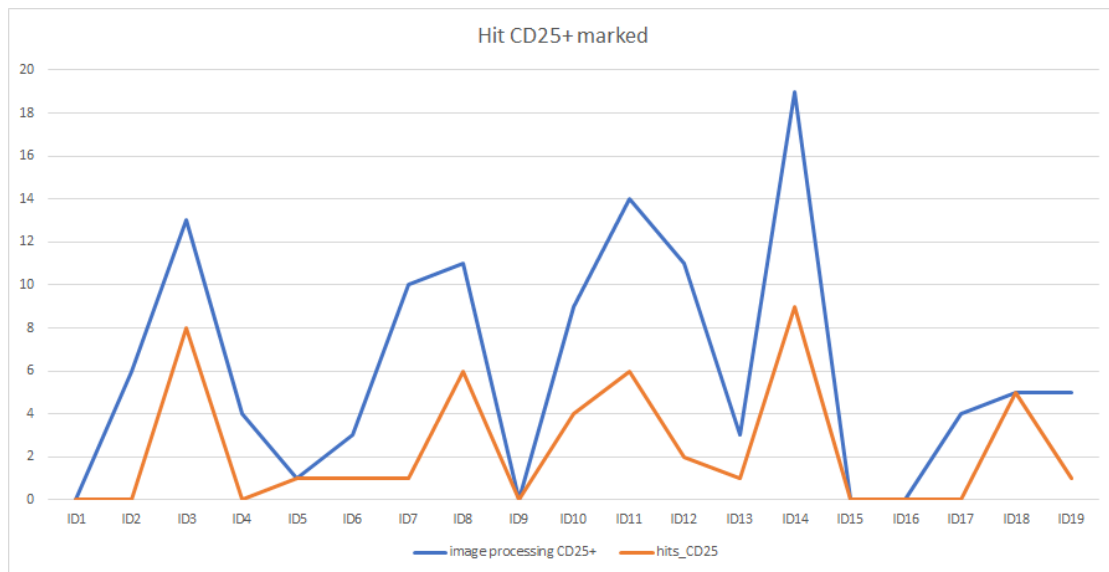


Figure 5.12: Amount of non-positive CD25+ that were close enough to ground truth to be considered a TP. The cells are from the image processing and the distance to the actual cell is calculated from the annotated data.

Cell Classification with Classical Image Processing

The classification part in the classical image processing step has been given a ground truth data to be classified and compared to the actual value. The classical image processing classification results are shown in Figure 5.13.

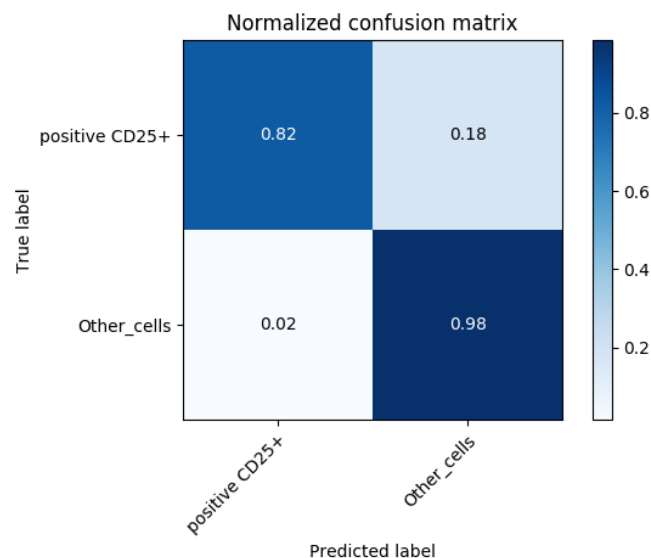


Figure 5.13: Confusion matrix for classical image processing classification.

Chapter 6

Discussion

This chapter will give short review and discuss the results from Chapter 4 and Chapter 5. Possible improvements and weaknesses will be the focus of the discussion.

6.1 Approach

The approach to get an estimation of cells using DNN without ground truth, was broken into steps. These steps were pre-processing the image, finding cells, classifying cells, and create a weakly labeled data set for training a DNN. Each step is discussed below.

6.1.1 Classical Image Processing

Pre-Processing

In the pre-processing step, a choice of size was made for convenience reasons; this size might not be the optimal size as it contains a limited field of view and results in a large number of images needed to be saved for later use. The size could be increased to 1024x1024, giving a larger field of view and fewer files needed to be stored. Using larger images would increase the time spent for each image in the following steps, but would reduce the total amount of time, as the number of images needed to load would be one quarter, and the total number of pixels to be processed would be the same. Different resolutions could also yield different results; the highest resolution available might have similar results as the second-highest resolution. If it is possible to lower the resolution and achieve similar results, fewer computations would be needed.

Locating Cells

In the process of locating cells, random WSI was tested for finding a threshold range that fits most WSI. However, some WSI deviates from the rest in terms of color intensity. Creating an adaptive threshold or applying histogram equalization to the images could give different and possibly better results. Using the blue mask seemed to give consistent results and is a good way to find the center of the cells, but in some cases where the biomarker CD25+ was on top of the cell, giving it a mix of brown and blue, and not found by the thresholds.

Using the morphological opening and closing could end up removing some cells or make adjacent cells merge in the blue mask, used for detection. Removing noise in the blue mask was necessary for the distance transform to locate the correct cells. The distance transform gave good overall results but struggled with overlapping cells to some extent

Classifying Cell

Using color to classify if the CD25+ marker positively marks the cell or not, seemed to yield good results, but can also misclassifying cells if leftovers of the biomarker surround them. As cells have different shapes and sizes, looking at a fixed area around them could make the classification pick up the brown CD25+ from adjacent cells. Different color ranges for brown, make the method work in most cases, but can struggle with images that deviate from the usual color intensities.

6.1.2 Convolutional Neural Network

Utilizing transfer learning with VGG-16 has some limitations. With an image of size, 512x512 being is split into tiles with size 64x64 the maximum amount of classifications available is 64, one for each tile. With the smaller tiles 32x32, the amount of classified tiles increases but is still not high enough to cover all scenarios. The smaller the tiles, the less information can be used when classifying cells, therefor going lower did not seem reasonable.

Since the VGG-16 network was trained on weak labels, it might not learn the optimal features needed for a better result. As the labeled data did contain a significant amount of class non-positive CD25+ marked, some of these were used for training.

6.2 Analyzing the Results

The first output from Chapter 4, provides a quick overview of the tissue, making it easier to know the location or if an area has a large amount of positive CD25+ markings. This quick overview could help experts in quickly finding an area of interest in the WSI image. As no expert has evaluated this, the prognostic value of this output has not been confirmed. Regardless of this output alone having a prognostic value, it could be used in other applications to find regions of interest to be inspected, the results from such applications could also possibly have a prognostic value. A slightly higher resolution could have more information to provide better-highlighted areas but would need more computations.

The results from classical image processing are a bit conflicting as the total estimate for cells is 97%, while the total hit percentage is 70%. These results can be interpreted as the estimation of cells were just a coincidence, but achieving similar results on 98 different images, acquired from 19 different patients is unlikely. The logical explanation for these results is that the filtering method for avoiding duplicates have picked an area that is outside the radius for the hit measurement.

Looking at the results from the VGG-16 network it can be seen that even if the results are overestimated and underestimated respectively for tiles sizes 32 and 64, it follows the original cell amount to some degree, indicating a correlation. This correlation was better for non-positive CD25+ cells than CD25+ marked cells. As the test images were randomly cropped, some cells were split between two to four tiles, making it affect the overall estimation; overlapping tiles and post-processing could solve this.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

This thesis proposed two approaches in estimating the amount of positive and non-positive CD25+ marked cells, one unlabeled approach utilizing classical image processing and second a labeled approach using a CNN network. The CD25+ marked histological images contain positive marked cells and other cells, and the positive marked ones are surrounded with a brown color which is used for classifying the cells. The use of Classical image processing to locate and classify cells based on color had a total estimation accuracy for positive and non-positive CD25+ marked cells, respectively underestimation of 4.7% and an overestimation of 4.5%, compared to the ground truth. Given the objective of this thesis to estimating cells, this is an acceptable result. Filtering out duplicate cells given by the distance transform local maxima resulted in less accurate localization of the cells, with a total hit rate of 70%.

Using transfer learning with a VGG-16 trained on the weakly labeled tiles from the classical image processing, it managed to learn sufficient features to get a 95% accuracy on the test data. The test data was created by an expert pathologist and had a cell in the center of the tile, for the classes with cells. The result after sending an image in, and randomly splitting it into tiles, gave promising results. These results showed a connection between the number of tiles classified as non-positive CD25+ cells and the correct number of cells, this was true for positive CD25+ markings as well, but it was not as clear as non-positive CD25+ cells. More extensive testing on a broader data set is needed to confirm the results.

7.2 Future Work

Finding highlighted areas at a lower resolution could be explored for several applications. For weak labeling, it could be used to find areas with a high amount of positive CD25+ markings to better provide labeled tiles for a given class. Having an automated system that highlights areas could help with quicker diagnosis and lead to a more automated system in total. The highlighted area could also give important areas that the previous system could focus on.

The methods in this thesis had some flaws in locating the exact position of a cell and separating overlapping cells. Improving this section could yield better results and possibly better prognostic value for pathologists.

Using overlapping 64x64 tiles followed by post-processing for estimation could remove the possibility of under-estimation caused by counting tiles. Using overlapping tiles could help with cells being split at the edge of the crop giving a half cell in neighboring tile, at the same time provide more information for the area that is to be classified. Introducing other classes that contain one or more cells could also help with the underestimation. creating new classes would require a new labeling process for existing classes as well as new ones. Exploring different CNNs could yield different and possibly better results,

Bibliography

- [1] Ferlay J, Ervik M, Lam F, Colombet M, Mery L, Piñeros M, Znaor A, Soerjomataram I, and Bray F. Cancer Incidence in Five Continents, CI5plus: IARC CancerBase No. 9, 2018. URL <https://gco.iarc.fr/today>.
- [2] Jun Miyazaki and Hiroyuki Nishiyama. Epidemiology of urothelial carcinoma. *International Journal of Urology*, 24(10):730–734, 2017. doi: 10.1111/iju.13376. URL <https://onlinelibrary.wiley.com/doi/abs/10.1111/iju.13376>.
- [3] Cancer Registry of Norway. Cancer in Norway 2017 - Cancer incidence, mortality, survival and prevalence in Norway. URL <https://www.kreftregisteret.no/globalassets/cancer-in-norway/2017/cin-2017.pdf>.
- [4] Ok Målfrid Mangrud. *Identification of patients with high and low risk of progression of urothelial carcinoma of the urinary bladder stage Ta and T1*. PhD, University of Bergen, 2014.
- [5] K. D. Sievert, B. Amend, U. Nagele, D. Schilling, J. Bedke, M. Horstmann, J. Hennenlotter, S. Kruck, and A. Stenzl. Economic aspects of bladder cancer: what are the benefits and costs? *World Journal of Urology*, 27(3):295–300, June 2009. ISSN 0724-4983. doi: 10.1007/s00345-009-0395-z. URL <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2694315/>.
- [6] Matthew Mossanen and John L. Gore. The burden of bladder cancer care: direct and indirect costs, September 2014. URL <https://www.ncbi.nlm.nih.gov/pubmed/24887047>.
- [7] 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. URL <https://apps.who.int/iris/handle/10665/41533>.
- [8] Lei He, L Rodney Long, Sameer Antani, and G Thoma. Computer assisted diagnosis in histopathology. *Sequence and genome analysis: methods and applications*, 15: 271–87, 2010.

- [9] Melinda Lillesand MSc, Vebjørn Kvikstad MD, Ok Målfrid Mangrud MD, PhD, Einar Gudlaugsson MD, PhD, Bianca van Diermen-Hidle BSc, Ivar Skaland PhD, Jan P. A. Baak MD, PhD, and Emiel A.M. Janssen PhD. Can immunological biomarkers CD8, CD4, CD25 and CD138 predict tumor recurrence and stage progression in non-muscle invasive urothelial carcinoma of the bladder? *Submitted for publication in Modern Pathology*, 2019.
- [10] F. A. Spanhol, L. S. Oliveira, C. Petitjean, and L. Heutte. Breast cancer histopathological image classification using Convolutional Neural Networks. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 2560–2567, July 2016. doi: 10.1109/IJCNN.2016.7727519.
- [11] Shawn Patrick Casey and Xinpeng Guan. Cervical Nuclei Segmentation in Whole Slide Histopathology Images Using Convolution Neural Network. In *Soft Computing in Data Science: 4th International Conference, SCDS 2018, Bangkok, Thailand, August 15-16, 2018, Proceedings*, volume 937, page 99. Springer, 2018.
- [12] Peter Bandi, Oscar Geessink, Quirine Manson, Marcory Van Dijk, Maschenka Balkenhol, Meyke Hermsen, Babak Ehteshami Bejnordi, Byungjae Lee, Kyunghyun Paeng, Aoxiao Zhong, Quanzheng Li, Farhad Ghazvinian Zanjani, Svitlana Zinger, Keisuke Fukuta, Daisuke Komura, Vlado Ovtcharov, Shenghua Cheng, Shaoqun Zeng, Jeppe Thagaard, Anders B. Dahl, Huangjing Lin, Hao Chen, Ludwig Jacobsson, Martin Hedlund, Melih cetin, Eren Halici, Hunter Jackson, Richard Chen, Fabian Both, Jorg Franke, Heidi Kusters-Vandeveld, Willem Vreuls, Peter Bult, Bram van Ginneken, Jeroen van der Laak, and Geert Litjens. From Detection of Individual Metastases to Classification of Lymph Node Status at the Patient Level: The CAMELYON17 Challenge. *IEEE Transactions on Medical Imaging*, 38(2):550–560, February 2019. ISSN 0278-0062, 1558-254X. doi: 10.1109/TMI.2018.2867350. URL <https://ieeexplore.ieee.org/document/8447230/>.
- [13] Jarle Urdal. *Image processing and classification of urothelial carcinoma using tissue sample images*. PhD thesis, University of Stavanger, June 2016. URL <https://uis.brage.unit.no/uis-xmlui/handle/11250/2414014>.
- [14] Rune Wetteland. *Classification of histological images of bladder cancer using deep learning*. PhD thesis, June 2017. URL <https://uis.brage.unit.no/uis-xmlui/handle/11250/2455555>.
- [15] Ørjan Malkenes. *Image processing on histopathological images of urothelial carcinoma – assessment of immune cells*. PhD thesis, June 2018. URL <https://uis.brage.unit.no/uis-xmlui/handle/11250/2565007>.

- [16] Nigel Kirkham and Nicholas R. Lemoine. *Progress in Pathology*. Cambridge University Press, January 2001. ISBN 978-1-84110-059-3.
- [17] Lester Borden, Peter Clark, and M. Hall. Bladder cancer. *Current Opinion in Oncology*, 17(3):275–280, May 2005. ISSN 1040-8746. doi: 10.1097/01.cco.0000156985.47984.9e. URL insights.ovid.com.
- [18] American Cancer Society. Immunotherapy for Bladder Cancer. URL <https://www.cancer.org/cancer/bladder-cancer/treating/immunotherapy-for-bladder-cancer.html>.
- [19] Richard J. Sylvester, Willem Oosterlinck, Sten Holmang, Matthew R. Sydes, Alison Birtle, Sigurdur Gudjonsson, Cosimo De Nunzio, Kikuo Okamura, Eero Kaasinen, Eduardo Solsona, Bedeir Ali-El-Dein, Can Ali Tatar, Brant A. Inman, James N'Dow, Jorg R. Oddens, and Marek Babjuk. Systematic Review and Individual Patient Data Meta-analysis of Randomized Trials Comparing a Single Immediate Instillation of Chemotherapy After Transurethral Resection with Transurethral Resection Alone in Patients with Stage pTa–pT1 Urothelial Carcinoma of the Bladder: Which Patients Benefit from the Instillation? *European Urology*, 69(2):231–244, 2016. ISSN 0302-2838. doi: <https://doi.org/10.1016/j.eururo.2015.05.050>. URL <http://www.sciencedirect.com/science/article/pii/S030228381500456X>.
- [20] M.D. Shelley, H. Kynaston, J. Court, T.J. Wilt, B. Coles, K. Burgon, and M.D. Mason. A systematic review of intravesical bacillus Calmette-Guérin plus transurethral resection vs transurethral resection alone in Ta and T1 bladder cancer. *BJU International*, 88(3):209–216, 2001. doi: 10.1046/j.1464-410x.2001.02306.x. URL <https://onlinelibrary.wiley.com/doi/abs/10.1046/j.1464-410x.2001.02306.x>.
- [21] John Cupitt and Kirk Martinez. VIPS: An image processing system for large images. *Proc. SPIE Vol. 2663*:p. 19 – 28, 1996. URL <https://eprints.soton.ac.uk/252227/1/vipsspie96a.pdf>.
- [22] Martinez, K and Cupitt, J. VIPS ? a highly tuned image processing software architecture. *IEEE International Conference on Image Processing*, pages pp. 574–577, 2005. URL <https://eprints.soton.ac.uk/262371/>.
- [23] The BigTIFF File Format, TIFF breaking the 4 gigabyte boundary, . URL <https://www.awaresystems.be/imaging/tiff/bigtiff.html>.
- [24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105.

- Curran Associates, Inc., 2012. URL <http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>.
- [25] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252, December 2015. ISSN 1573-1405. doi: 10.1007/s11263-015-0816-y. URL <https://doi.org/10.1007/s11263-015-0816-y>.
- [26] N. Otsu. A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, January 1979. ISSN 0018-9472. doi: 10.1109/TSMC.1979.4310076.
- [27] Warren S. McCulloch and Walter H. Pitts. A Logical Calculus of the Ideas Immanent in Nervous Activity. *Bulletin of mathematical biophysics*, vol. 5:115–133, 1943.
- [28] D Steinkraus, I Buck, and Patrice Simard. Using GPUs for machine learning algorithms. pages 1115 – 1120 Vol. 2, 2005. doi: 10.1109/ICDAR.2005.251.
- [29] Goodfellow Ian, Bengio Yoshua, and Courville Aaron. Deep Learning (Adaptive Computation and Machine Learning series). In *Deep Learning (Adaptive Computation and Machine Learning series)*. MIT Press, 2016. URL <http://www.deeplearningbook.org>.
- [30] The History of the Modern Graphics Processor, . URL <https://www.techspot.com/article/650-history-of-the-gpu/>.
- [31] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, May 2017. ISSN 00010782. doi: 10.1145/3065386. URL <http://dl.acm.org/citation.cfm?doid=3098997.3065386>.
- [32] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv 1409.1556*, 2014.
- [33] Leica SCN400 Slide Scanner | Meyer Instruments, . URL <https://meyerinst.com/slide-scanner/index.htm>.
- [34] Peter Bankhead, Maurice B. Loughrey, José A. Fernández, Yvonne Dombrowski, Darragh G. McArt, Philip D. Dunne, Stephen McQuaid, Ronan T. Gray, Liam J. Murray, Helen G. Coleman, Jacqueline A. James, Manuel Salto-Tellez, and Peter W. Hamilton.

- QuPath: Open source software for digital pathology image analysis. *Scientific Reports*, 7(1):16878, December 2017. ISSN 2045-2322. doi: 10.1038/s41598-017-17204-5. URL <https://www.nature.com/articles/s41598-017-17204-5>.
- [35] Xiaomeng Li, Hao Chen, Xiaojuan Qi, Qi Dou, Chi-Wing Fu, and Pheng-Ann Heng. H-DenseUNet: hybrid densely connected UNet for liver and tumor segmentation from CT volumes. *IEEE transactions on medical imaging*, 37(12):2663–2674, 2018.
- [36] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. *arXiv:1505.04597 [cs]*, May 2015. URL <http://arxiv.org/abs/1505.04597>. arXiv: 1505.04597.
- [37] Dayong Wang, Aditya Khosla, Rishab Gargeya, Humayun Irshad, and Andrew H Beck. Deep learning for identifying metastatic breast cancer. *arXiv preprint arXiv:1606.05718*, 2016.
- [38] Azam Hamidinekoo, Erika Denton, Andrik Rampun, Kate Honnor, and Reyer Zwiggelaar. Deep learning in mammography and breast histology, an overview and future trends. *Medical Image Analysis*, 47:45 – 67, 2018. ISSN 1361-8415. doi: <https://doi.org/10.1016/j.media.2018.03.006>. URL <http://www.sciencedirect.com/science/article/pii/S1361841518300902>.

Appendix A

Supplementary Material

The attached zip files contains the code used in this thesis.

The zipped file contains two folders, one for VGG-16 and one for classical image processing. Each folder has a readme.txt that explains the purpose of each file.

The zipped file also contains a Readme.txt file for setting up a local server on a laptop, for usage of on premiss data.

Appendix B

Table of Results

B.1 Image Processing

ID	Image processing CD25+	Correct amount CD25+	Image processing Other cells	Correct Other cells	Image process total	Annotated total
ID1	0	0	309	356	309	356
ID2	6	0	309	293	315	293
ID3	13	15	162	149	175	164
ID4	4	0	73	62	77	62
ID5	1	1	88	75	89	76
ID6	3	1	81	66	84	67
ID7	10	6	321	310	331	316
ID8	11	11	249	224	260	235
ID9	0	0	21	50	21	50
ID10	9	14	285	397	294	411
ID11	14	22	473	426	487	448
ID12	11	4	244	208	255	212
ID13	3	4	220	198	223	202
ID14	19	16	552	817	571	833
ID15	0	0	108	96	108	96
ID16	0	0	297	299	297	299
ID17	4	0	360	298	364	298
ID18	5	12	546	610	551	622
ID19	5	8	646	611	651	619
total:	118	114	5344	5545	5462	5659

Table B.1: Results from classical image processing for localization and classification

B.2 VGG-16

Name	CD25	Other	VGG16 CD25	VGG16 Other	VGG16 background	Number of Images
ID1	0	356	0	95	97	3
ID2	0	293	17	112	319	7
ID3	15	149	24	47	121	3
ID4	0	62	23	13	92	2
ID5	1	75	0	23	41	1
ID6	1	66	2	28	34	1
ID7	6	310	49	119	344	8
ID8	11	224	28	49	115	3
ID9	0	50	0	19	45	1
ID10	14	397	5	6	117	2
ID11	22	426	29	104	379	8
ID12	4	208	12	31	277	5
ID13	4	198	0	64	256	5
ID14	16	817	37	258	153	7
ID15	0	96	1	27	356	6
ID16	0	299	15	55	378	7
ID17	0	298	35	95	382	8
ID18	12	610	11	198	431	10
ID19	8	611	10	139	555	11
Total	114	5545	298	1482	4492	98

Table B.2: Result from VGG-16 network trained on 64x64 tiles

Name	CD25	Other	VGG16 CD25	VGG16 Other	VGG16 Background	Number of images
ID1	0	356	0	382	386	3
ID2	0	293	48	475	1269	7
ID3	15	149	66	230	472	3
ID4	0	62	69	76	367	2
ID5	1	75	3	119	134	1
ID6	1	66	3	105	148	1
ID7	6	310	142	441	1465	8
ID8	11	224	67	334	367	3
ID9	0	50	0	60	196	1
ID10	14	397	27	189	296	2
ID11	22	426	94	539	1415	8
ID12	4	208	48	280	952	5
ID13	4	198	15	289	976	5
ID14	16	817	82	1119	591	7
ID15	0	96	1	132	1403	6
ID16	0	299	22	309	1461	7
ID17	0	298	101	394	1553	8
ID18	12	610	37	855	1668	10
ID19	8	611	35	725	2056	11
Total	114	5545	860	7053	17175	98

Table B.3: Result from VGG-16 network trained on 32x32 tiles

Appendix C

Images Used in Tests

